BSc in Computer Engineering
CMP4103
Computer Systems and Network Security

Lecture 8

Network Security and an introduction to Penetration Testing

Eng Diarmuid O'Briain, CEng, CISSP

Department of Electrical and Computer Engineering,
College of Engineering, Design, Art and Technology,
Makerere University

Table of Contents

1. THE KALI LINUX VIRTUAL MACHINE....5

2. THE KALI LINUX USB DRIVE....8

2.1 Download Kali Linux....8
2.2 Discover USB block device....8
2.3 Copy .iso to the drive using dd....8

3. NETWORK SECURITY AND PENETRATION TESTING....9

3.1 Penetration testing steps....9

4. KALI LINUX....11

4.1 Root user....11
4.2 System update....11

5. INFORMATION GATHERING AND ANALYSIS....12

5.1 Fierce....12
5.2 nmap....13
5.3 Use nmap anonymously....13
5.4 zenmap....17

6. VULNERABILITY DETECTION AND EXPLOITATION....18

6.1 OpenVAS....18
6.2 Metasploit....22
6.3 Armitage....24
6.4 Testing Web Servers and Web Applications....28
6.5 Nikto....28
6.6 Open Web Application Security Project (OWASP)....29
6.7 OWASP Zed Attack Proxy (ZAP)....29
6.8 Reporting....31

7. DETECTION SYSTEMS....32

7.1 p0f....32
7.2 Port Scan Attack Detector (psad)....33
7.3 Passive Asset Detection System (pads)....36

8. SUMMARY....36

9. LAB EXERCISE....37

10. BIBLIOGRAPHY....37

Illustration Index

Illustration 1: Kali Linux Desktop....5
Illustration 2: VirtualBox network configuration....6
Illustration 3: Kali Linux network test....7
Illustration 4: Using the TOR network....13
Illustration 5: Source for connections through TOR....16
Illustration 6: Zenmap....17
Illustration 7: Greenbone login....19
Illustration 8: Initial dashboard....20
Illustration 9: OpenVAS Task Wizard....20
Illustration 10: OpenVAS post scan findings....21
Illustration 11: OpenVAS detail....21
Illustration 12: Armitage connect to database....24
Illustration 13: Metaspoilt via Armitage....25
Illustration 14: Armitage, scanning....25
Illustration 15: Armitage, attack....26
Illustration 16: Armitage, making the attack....26
Illustration 17: Armitage, Hail Mary attack....27
Illustration 18: Armitage, .csv report....27
Illustration 19: Zed Attack Proxy (zap)....29
Illustration 20: Zap post scan alerts....30
Illustration 21: Zap reporting....31

1. The Kali Linux Virtual Machine

Using the Kali Linux image provided on the website below, install ***VirtualBox***, build the ***.ova*** image, install and run.

https://www.offensive-security.com/kali-linux-vmware-virtualbox-image-download/

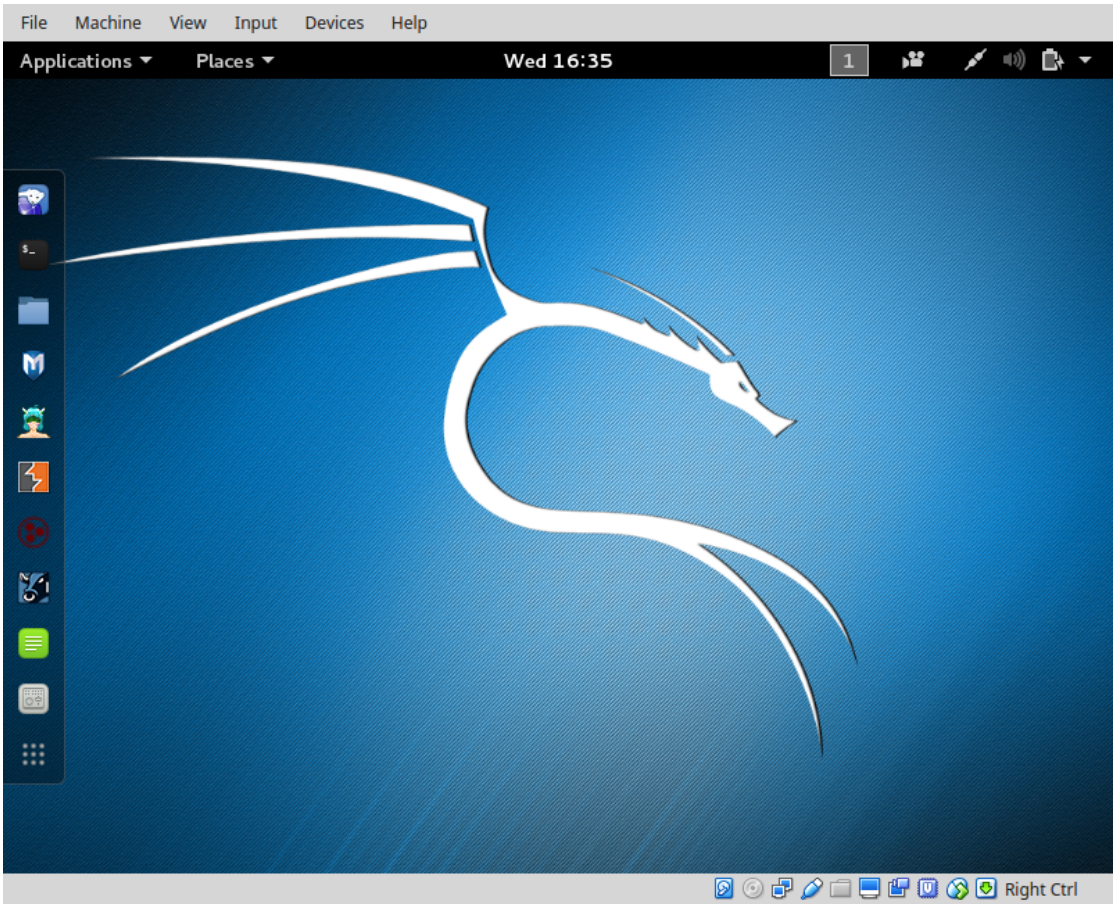


Illustration 1: Kali Linux Desktop

Login to the image with the default root username (***root***) and password (***toor***).

Run up a shell and confirm connectivity with the Internet.

```
root@kali:~# ip addr list dev eth0 | grep 'inet ' | awk '{print $2}'
10.0.2.15/24
```

The IP Address is assigned by Network Address Translation (NAT) to the VM. It is possible to bridge the VM Ethernet interface (eth0) with the active interface on the host to get an IP address from the real world Dynamic Host Configuration Protocol (DHCP) Server.

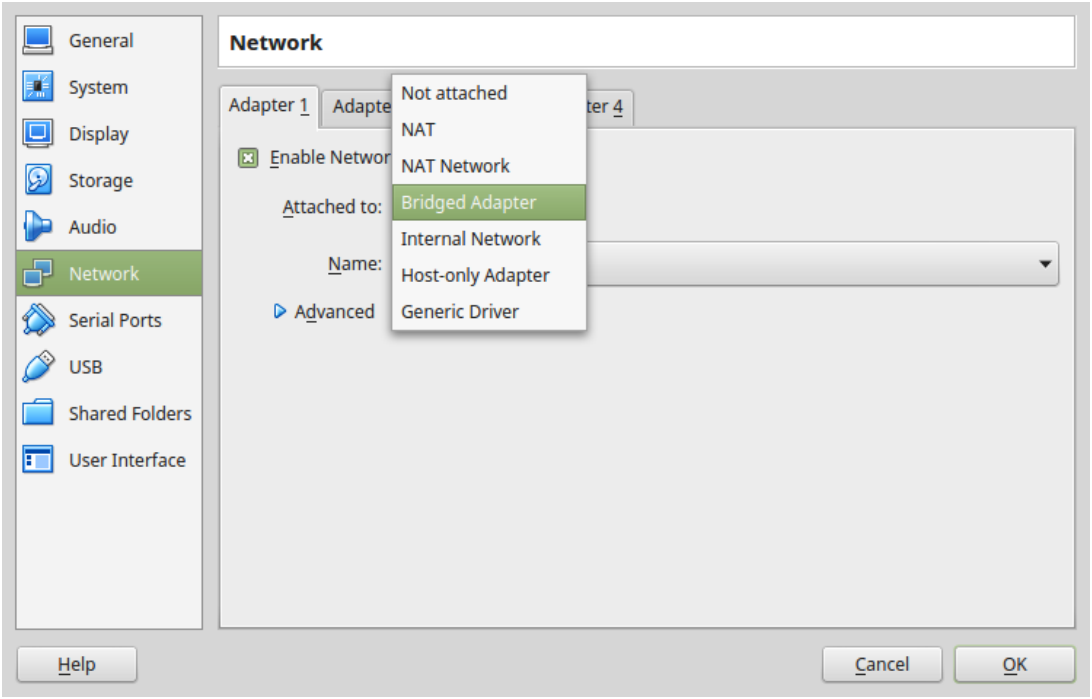


Illustration 2: VirtualBox network configuration

Whichever system is used the Internet Protocol (IP) Packet InterNet Groper (PING) test to the main google nameserver at 8.8.8.8 should elicit a response.

```
root@kali:~# ping -c3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=63 time=307 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=63 time=462 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=63 time=298 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 298.745/356.467/462.706/75.217 ms
root@kali:~#
```

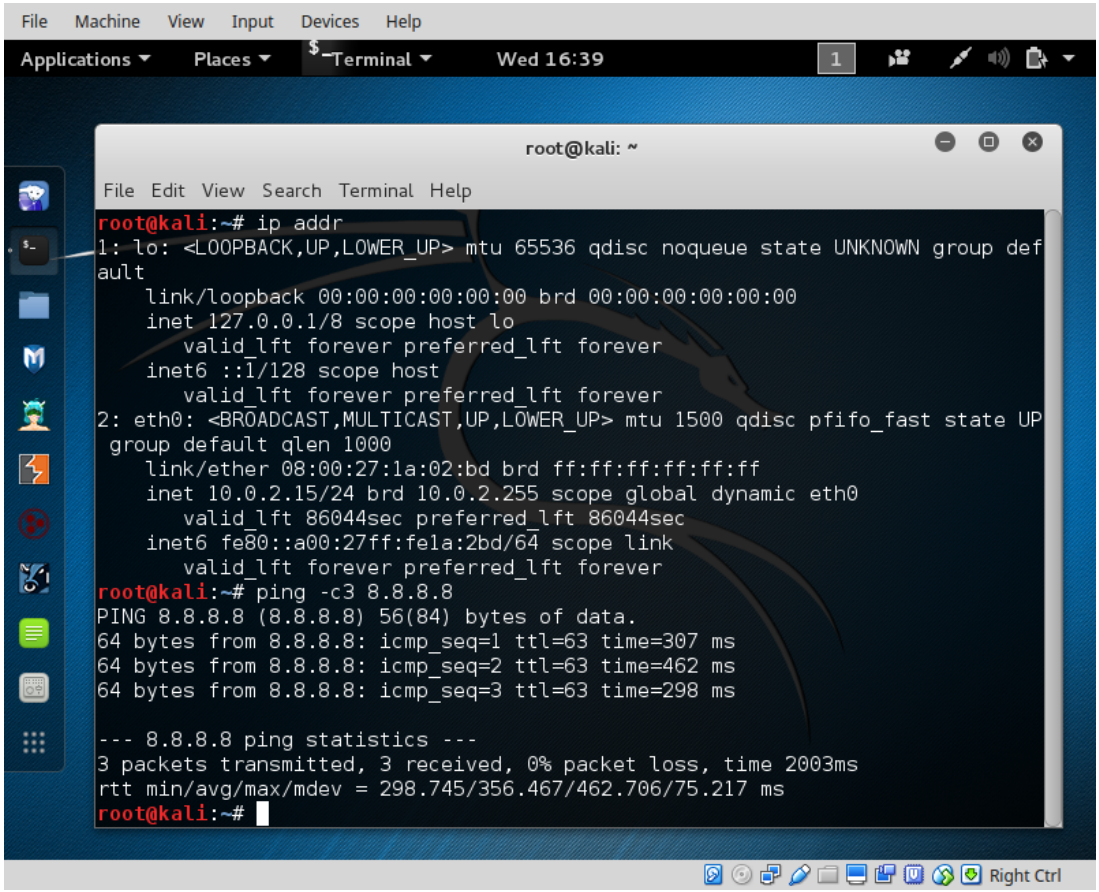


Illustration 3: Kali Linux network test

2. The Kali Linux USB Drive

Kali Linux is a very useful tool and having a copy on a USB Drive that can boot live on any computer is very handy indeed. Follow these steps to create a Kali Linux USB Drive of your own.

2.1 Download Kali Linux

Download the latest Kali Linux, in this case 2017.2 and verify the download using the procedure on the webpage - https://www.kali.org/downloads.

2.2 Discover USB block device

Insert the USB and verify block device name

```
ada:~$ lsblk
 NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
 sr0     11:0    1  1024M  0 rom
 sda      8:0    0 931.5G  0 disk
 ├─sda2   8:2    0     1K  0 part
 ├─sda5   8:5    0  15.8G  0 part [SWAP]
 └─sda1   8:1    0 915.8G  0 part /
```

Plug in the USB Drive.

```
ada:~$ lsblk
 NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
 sdb      8:16   1   7.3G  0 disk
 ├─sdb2   8:18   1   2.3M  0 part
 └─sdb1   8:17   1   685M  0 part /media/ada/Ubuntu-Server 17.04 amd64
 sr0     11:0    1  1024M  0 rom
 sda      8:0    0 931.5G  0 disk
 ├─sda2   8:2    0     1K  0 part
 ├─sda5   8:5    0  15.8G  0 part [SWAP]
 └─sda1   8:1    0 915.8G  0 part /
```

Therefore the USB Drive is block device */dev/sdb*.

2.3 Copy .iso to the drive using dd

Copy a file *kali-linux-2017.2-amd64.iso* and write to the USB Drive at */dev/sdb*. The *pv* command between the pipes monitors the progress of data through the pipe.

```
ada:~$ dd if=kali-linux-2017.2-amd64.iso | pv | sudo dd of=/dev/sdb bs=512k
[sudo] password for alovelace: babbage
 5899648+0 records in0MiB/s] [  <=>                                        ]
 5899648+0 records out
 3020619776 bytes (3.0 GB, 2.8 GiB) copied, 599.213 s, 5.0 MB/s
  2.81GiB 0:09:59 [4.81MiB/s] [     <=>                                    ]
  0+40085 records in
  0+40085 records out
  3020619776 bytes (3.0 GB, 2.8 GiB) copied, 594.7 s, 5.1 MB/s
```

That is it. Put the USB Disk in a computer and boot.

3. Network Security and Penetration testing

Penetration testing (also called ***pen-testing***) is the practice of testing a computer system, network or Web application to find vulnerabilities that an attacker could exploit.

It is a proactive and authorised attempt to evaluate the security of an IT infrastructure by safely attempting to exploit system vulnerabilities, including OS, service and application flaws, improper configurations, and even risky end-user behaviour.

3.1 Penetration testing steps

3.1.1 Planning and Preparation

A kick-off meeting with the client to discuss in detail the scope and the overall objective of the pen-test. A clear objective is essential for the pen-test. Typical objective is to demonstrate that exploitable vulnerabilities do in fact exist with the organisation computing and network infrastructure. As part of the scoping identify:

- Timing and duration allowed for the pen-tests
- Personnel involved
- Are staff being informed of the tests?
- Network and Computers involved
- Operational requirements during the pen-test
- How the results are to be presented at the conclusion of the test.

After this scoping meeting the pen-testers need to develop a ***Penetration Test Plan*** which should be shared with the client company. It must include:

- The detailed test plan itself. What tests are to be performed and on what.
- A ***Confidentiality statement*** that is signed by both the pen-testers and the client.
- A clear ***Acceptance sign-off sheet*** that the ***Penetration Test Plan*** is acceptable to the client and affords legal protection to the pen-testers.

Remember the pen-testers are actually conducting tests that are deemed illegal and therefore require the indemnity of the Acceptance sign-off from the client company.

3.1.2 Information Gathering and Analysis

Gathering of as much information as possible as a reconnaissance is essential.

- What does the network look like?
- What devices are on the network?
- Who works at the company?
- What does the organogram of the company look like?

3.1.3 Vulnerability detection

Once a picture of the target organisation has been compiled a scan of vulnerabilities is the next step.

3.1.4 Penetration attempt

Once a list of vulnerabilities have been identified and logged it is time to attempt a penetration. Identifying the best targets from the machines showing vulnerability is important particularly if the time given is short. Identifying the juicy targets may be as simply as looking at the machine names as it is a habit of IT personnel to use functional names like MAILSVR or FTPSERVER etc…

Define the list of machines that are to be given special additional treatment. Try password cracking tools, dictionary, brute force and hybrid attacks.

3.1.5 Analysis and Reporting

A detailed report must be furnished to the client at the conclusion of the tests. It should include:

- A summary of successful penetration tests.
- A list of all information gathered during the pen-test.
- A complete list and description of vulnerabilities found (including on machines not singled out for a penetration attempt).
- A suggested list of next steps to close the vulnerabilities and increase security at the client company.

3.1.6 Tidy up

During the pen-testing a detailed list of steps taken should be maintained. On the conclusion of the testing the pen-testers work with the client staff ensure that the steps have not left and residual issues, like entries in configuration files, new users or groups etc..

4. Kali Linux

The GNU/Linux operating system includes a vast array of tools for each step of the pen-testing activity. All of the tools described here can be installed on any GNU/Linux distribution. Kali Linux, derived from Debian GNU/Linux is a distribution specifically designed for digital forensics and penetration testing. It is maintained and funded by Offensive Security Ltd. Kali Linux comes pre-installed with over 600 penetration-testing programs.

4.1 Root user

GNU/Linux distributions generally recommend the use of a non-privileged account while running the system and use a utility like ***sudo*** when and if escalation of privileges is required. As Kali Linux is a security and auditing platform it contains tools that can only be ran under root privileges and therefore the root account is used. As a result care should be taken and is not the GNU/Linux distribution for Linux beginners.

4.2 System update

Before looking at any of the programs it is important to perform a update of the system.

```
root@kali:~# apt update
Get:1 http://security.kali.org sana/updates InRelease [11.9 kB]
Get:2 http://http.kali.org sana InRelease [20.3 kB]
Get:3 http://http.kali.org sana-proposed-updates InRelease [14.1 kB]
Get:4 http://security.kali.org sana/updates/main Sources [74.5 kB]
Get:5 http://http.kali.org sana/main Sources [9,089 kB]
Ign http://security.kali.org sana/updates/contrib Translation-en_US
. . . .
. . . .
Ign http://http.kali.org sana-proposed-updates/non-free Translation-en
Fetched 22.7 MB in 1min 41s (222 kB/s)
Reading package lists... Done

root@kali:~# apt dist-upgrade
```

5. Information Gathering and Analysis

One of the oldest tools and still one of the most effective for security administration is the Network exploration tool and security / port scanner (***nmap***) tool. This is a shell based network exploration and security auditing tool. It has a sister tool ***zenmap*** that gives it a graphical interface.

5.1 Fierce

Fierce is a lightweight scanner that helps locate non-contiguous IP space and host-names against specified domains. It is used as a pre-cursor to ***nmap*** as it requires knowledge of the IP already. It locates likely targets both inside and outside a corporate network. Because it uses DNS primarily you will often find miss-configured networks that leak internal address space. That's especially useful in targeted malware.

```
root@kali:~# fierce -dns adomain.com
DNS Servers for adomain.com:
      ns2.adomain.com
      ns1.adomain.com

Trying zone transfer first...
      Testing ns2.adomain.com
            Request timed out or transfer not allowed.
      Testing ns1.adomain.com
            Request timed out or transfer not allowed.

Unsuccessful in zone transfer (it was worth a shot)
Okay, trying the good old fashioned way... brute force

Checking for wildcard DNS...
      ** Found 97919448768.adomain.com at 68.95.161.145.
      ** High probability of wildcard DNS.
Now performing 2280 test(s)...
68.95.161.6       unix.adomain.com
68.95.161.93      mx.adomain.com
68.95.161.92      mx.adomain.com
68.95.161.237     www.adomain.com

Subnets found (may want to probe here using nmap or unicornscan):
      68.95.161.0-255 : 4 hostnames found.
      176.58.111.0-255 : 1 hostnames found.

Done with Fierce scan: http://ha.ckers.org/fierce/
Found 4 entries.

Have a nice day.
```

5.2 nmap

Network Mapper (***nmap***) is an open source tool for network exploration and security auditing. It forms the basis for most of the other tools that are used for penetration testing and scanning. Open a GNU/Linux distribution install ***nmap*** and ***zenmap*** as follows. On Kali Linux this step is unnecessary as it is already pre-installed.

```
ada:~$ sudo apt install nmap zenmap xprobe
```

Run ***nmap*** against a target IP address

- **-p <port ranges>**: Only scan specified ports
- **-Pn**: Treat all hosts as online, skip host discovery

If you want to record the scan simply pipe to a file, or if you also want to see the output to the screen as well as record use the ***tee*** utility in the bash shell.

```
root@kali:~# nmap -Pn 192.168.89.1 | tee /tmp/nmap-output.txt

Starting Nmap 6.40 ( http://nmap.org ) at 2015-11-03 11:41 EAT
Nmap scan report for 192.168.89.1
Host is up (0.00086s latency).
Not shown: 65530 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
80/tcp   open  http
2000/tcp open  cisco-sccp
8291/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 6.00 seconds
```

5.3 Use nmap anonymously

For anonymous use of ***nmap*** it is possible to do so using 'The Onion Router (***TOR***) and ***ProxyChains***. ProxyChains redirects TCP connections through proxy servers

```
ada:~$ sudo apt install tor proxychains
```

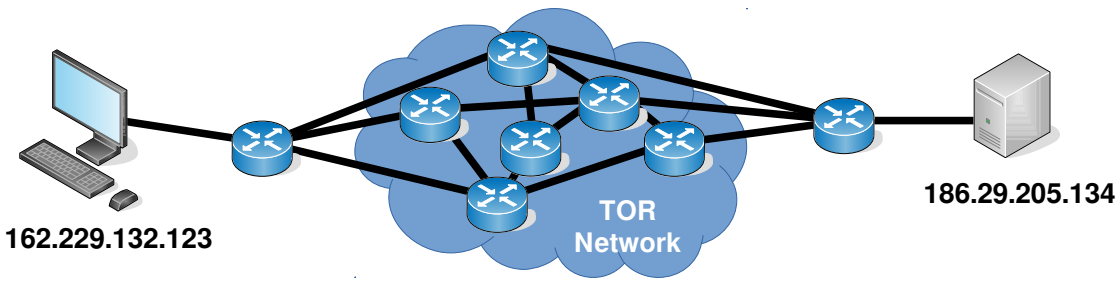


Illustration 4: Using the TOR network

Here is an Nmap scan through a proxy chain via the TOR network. Some additional options here:

- **-sT**: TCP connect scan, instead of writing raw packets as most other scan types do, Nmap asks the underlying OS to establish a connection with the target machine and port by issuing the connect system call. This more exactly simulates what network enables applications would do. Bacically Nmap is making use of the OS own Berkeley Socket API.

```
ada:~$ proxychains nmap -Pn -sT -p 22,80 186.29.205.134

ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 6.40 ( http://nmap.org ) at 2015-11-04 22:07 EAT
|S-chain|-<>-127.0.0.1:9050-<><>-186.29.205.134:80-<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-186.29.205.134:22-<><>-OK
Nmap scan report for 186.29.205.13
Host is up (0.61s latency).
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

Adding an additional option to detect the OS.:

- **-sV**: Enable version detection. It can be used to help differentiate the truly open ports from the filtered ones.

```
ada:~$ proxychains nmap -Pn -sV -sT -p 22,80 186.29.205.134

ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 6.40 ( http://nmap.org ) at 2015-11-10 12:13 EAT
|S-chain|-<>-127.0.0.1:9050-<><>-186.29.205.134:22-<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-186.29.205.134:80-<><>-OK
Nmap scan report for li489-237.members.linode.com (186.29.205.134)
Host is up (0.71s latency).
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
80/tcp open  http    Apache httpd 2.2.22 ((Debian))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Attempt at SSH connection using user root failed but as it passed through the TOR network the attempt was anonymous.

```
ada:~$ proxychains ssh root@186.29.205.134

ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-127.0.0.1:9050-<><>-186.29.205.134:22-<><>-OK
root@186.29.205.134's password:
Permission denied, please try again.
root@186.29.205.134's password:
Permission denied, please try again.
root@186.29.205.134's password:
Permission denied (publickey,password).
```

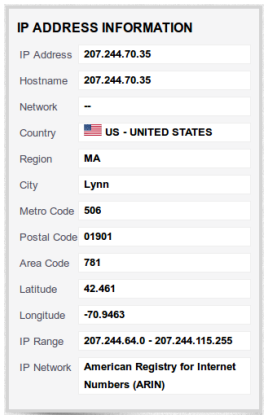
On the server that the compromise attempt occurred check the authentication logs.

```
root@ece:~# tail /var/log/auth.log

Nov  4 19:09:26 ece sshd[1146]: Failed password for root from 207.244.70.35
port 45909 ssh2
Nov  4 19:09:33 ece sshd[1146]: Failed password for root from 207.244.70.35
port 45909 ssh2
Nov  4 19:09:40 ece sshd[1146]: Failed password for root from 207.244.70.35
port 45909 ssh2
Nov  4 19:09:40 ece sshd[1146]: Connection closed by 207.244.70.35 [preauth]
Nov  4 19:09:40 ece sshd[1146]: PAM 2 more authentication failures; logname=
uid=0 euid=0 tty=ssh ruser= rhost=207.244.70.35  user=root
```

Note the IP Address from where the attempted connection originated, it is not from the actual source 162.229.132.123 but from 207.244.70.35 which is the edge of the TOR at that time for that connection.

http://www.ipaddress-finder.com



IP ADDRESS INFORMATION

IP Address	207.244.70.35
Hostname	207.244.70.35
Network	--
Country	US - UNITED STATES
Region	MA
City	Lynn
Metro Code	506
Postal Code	01901
Area Code	781
Latitude	42.461
Longitude	-70.9463
IP Range	207.244.64.0 - 207.244.115.255
IP Network	American Registry for Internet Numbers (ARIN)

5.3.1 SSH Public Key as possible Identifier in TOR

One thing to consider however about making SSH connections through the TOR network is that by default the connection will attempt to authenticate using your public key first. If you have one and this has been made public then it could be an identifier if in the unlikely but possible even that someone is capturing the connection. To remove this possibility create a new public key first specify it in the SSH connection:

```
ada:~$  ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ece/.ssh/id_rsa): id_rsa_ANONY
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_rsa_ANONY.
Your public key has been saved in id_rsa_ANONY.pub.
The key fingerprint is:
bc:34:b1:23:fd:5a:f2:4b:d9:88:af:70:f7:d6:39:a2
The key's randomart image is:
+--[ RSA 2048]----+
|         .       |
|        o o      |
|       . S       |
|        o * +    |
|       . = B .. . |
|        o O .o +  |
|         o.E+.. . |
+-----------------+
```

```
ada:~$ proxychains ssh -i /home/ece/.ssh/id_rsa_ANONY root@186.29.205.134

ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-127.0.0.1:9050-<><>-186.29.205.134:22-<><>-OK
root@176.58.111.237's password: BADPASS
Permission denied, please try again.
root@176.58.111.237's password: GOODPASS
Linux www 4.1.5-x86_64-linode61 #7 SMP Mon Aug 24 13:46:31 EDT 2015 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov  9 03:20:34 2015 from 160.242.131.178

root@ece:~# tail /var/log/auth.log
Nov 10 09:46:10 ece sshd[21706]: Failed password for root from 43.229.53.25
port 11978 ssh2
Nov 10 09:46:12 ece sshd[21706]: Failed password for root from 43.229.53.25
port 11978 ssh2
Nov 10 09:46:12 ece sshd[21706]: Received disconnect from 43.229.53.25: 11:
[preauth]
Nov 10 09:46:12 ece sshd[21706]: PAM 2 more authentication failures; logname=
uid=0 euid=0 tty=ssh ruser= rhost=43.229.53.25  user=root
Nov 10 09:46:13 ece sshd[21708]: pam_unix(sshd:auth): authentication failure;
logname= uid=0 euid=0 tty=ssh ruser= rhost=43.229.53.25  user=root
Nov 10 09:46:15 ece sshd[21708]: Failed password for root from 43.229.53.25
port 28216 ssh2
Nov 10 09:46:17 ece sshd[21708]: Failed password for root from 43.229.53.25
port 28216 ssh2
Nov 10 09:46:19 ece sshd[21708]: Failed password for root from 43.229.53.25
port 28216 ssh2
Nov 10 09:46:19 ece sshd[21708]: Received disconnect from 43.229.53.25: 11:
[preauth]
Nov 10 09:46:19 ece sshd[21708]: PAM 2 more authentication failures; logname=
uid=0 euid=0 tty=ssh ruser= rhost=43.229.53.25  user=root
```

Each time the source is a different address as the exit point from TOR changes.

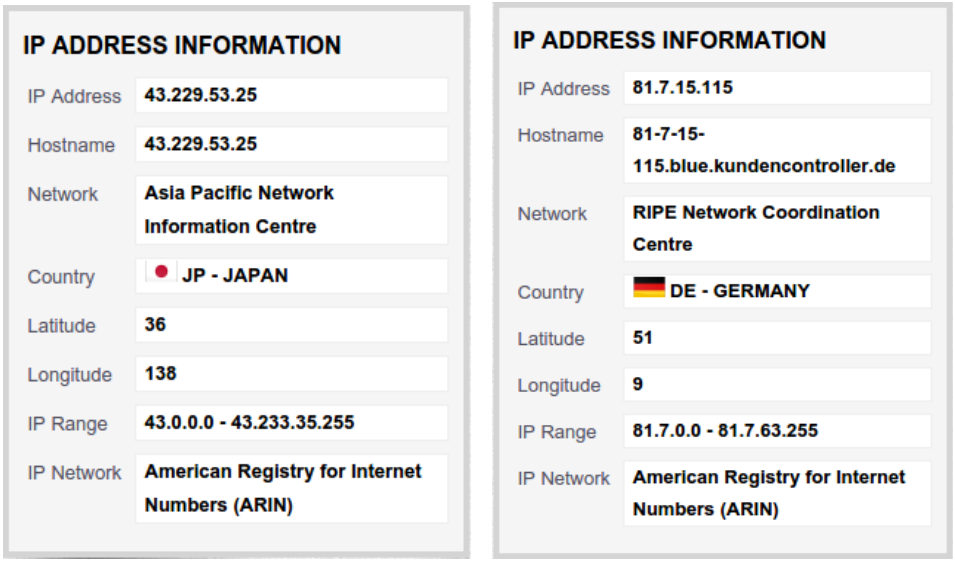


Illustration 5: Source for connections through TOR

5.4 zenmap

zenmap is a very useful tool. It gives a graphical interface to ***nmap*** and is an easy way to sort through the multitude of options within the parent tool.

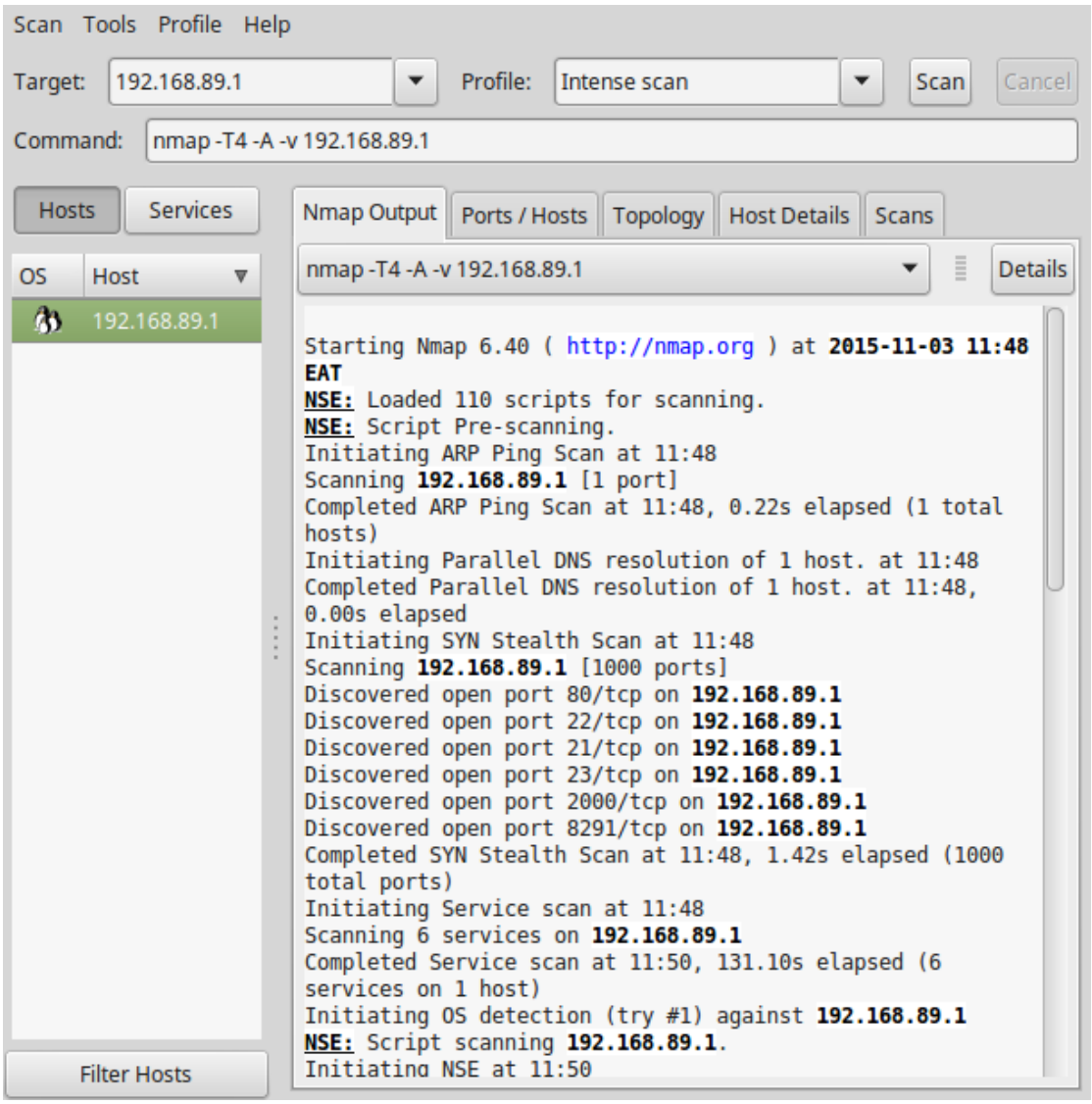


Illustration 6: Zenmap

6. Vulnerability Detection and Exploitation

6.1 OpenVAS

The Open Vulnerability Assessment System (***OpenVAS***) is a GNU General Public License (GNU GPL) framework of several services and tools offering a comprehensive and powerful vulnerability scanning and vulnerability management solution.

The actual security scanner is accompanied with a daily updated feed of Network Vulnerability Tests (NVTs), over 100,000 in total.

6.1.1 Install OpenVAS 9 on Kali

Install OpenVAS 9 on Kali Linux 2017.2. The second setup will take some time so be patient.

```
root@kali:/# apt install openvas
root@kali:/# openvvas-setup
```

6.1.2 OpenVAS User

Create an OpenVAS User and Password with Admin rights.

```
root@kali:/# openvasmd --create-user=MyOpenVASuser --role=Admin
User created with password '9cecf166-8cd0-4d31-9e09-3fe13c48eca0'.
root@kali:/# openvasmd --user=MyOpenVASuser --new-password=MyOpenVASpass
```

6.1.3 Update the database of NVTs

Update the NVT database, this step should be carried out regularly.

```
root@kali:/# openvasmd --update
root@kali:/# openvasmd --rebuild
root@kali:/# systemctl restart openvas-scanner
```

6.1.4 Greenbone assistant access

By default it is only possible to access the greenbone assistant from the localhost. To allow access from other hosts.

```
root@kali:/# sed -i.bak -e 's/--listen=127.0.0.1/--listen=0.0.0.0/'
/lib/systemd/system/greenbone-security-assistant.service
```

Reload systemd manager configuration and restart the greenbone security assistant.

```
root@kali:/# systemctl daemon-reload
root@kali:/# systemctl restart greenbone-security-assistant
```

6.1.5 Checking the OpenVAS installation

The OpenVAS installation can be checked and any problems fixed. When all is OK it should give an OK message.

```
root@kali:/# openvas-check-setup
It seems like your OpenVAS-9 installation is OK.
```

6.1.6 Run OpenVAS

Start the OpenNAS server.

```
root@kali:~# openvas-start
Starting OpenVas Services
```

At this stage the OpenVAS manager, scanner, and Greenbone Security Assistant (***GSAD***) services should be listening:

```
root@kali:/# netstat -antp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address  Foreign Address  State PID/Program name
tcp       0      0 127.0.0.1:9390   0.0.0.0:*  LISTEN  2745/openvasmd
tcp       0      0 127.0.0.1:80     0.0.0.0:*  LISTEN  4421/gsad
tcp       0      0 127.0.0.1:9392   0.0.0.0:*  LISTEN  4420/gsad
```

-a All, -n Numeric, -t TCP, -p Program

6.1.7 Using the web client

Note the webclient will only work to **https:**// not **http:**//

https://127.0.0.1:9392

It is also possible to browse to the Kali Linux host.

https://192.168.89.3:9392

Use the Username and Password created above.

```
Username: MyOpenVASuser
Password: MyOpenVASpass
```

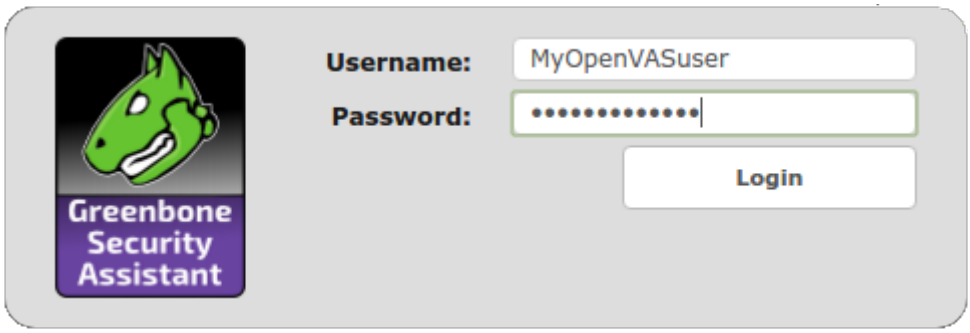


Illustration 7: Greenbone login

Illustration 8: Initial dashboard

- Define a target in ***Scans >> Tasks***.
- Select the Task Wizard icon.

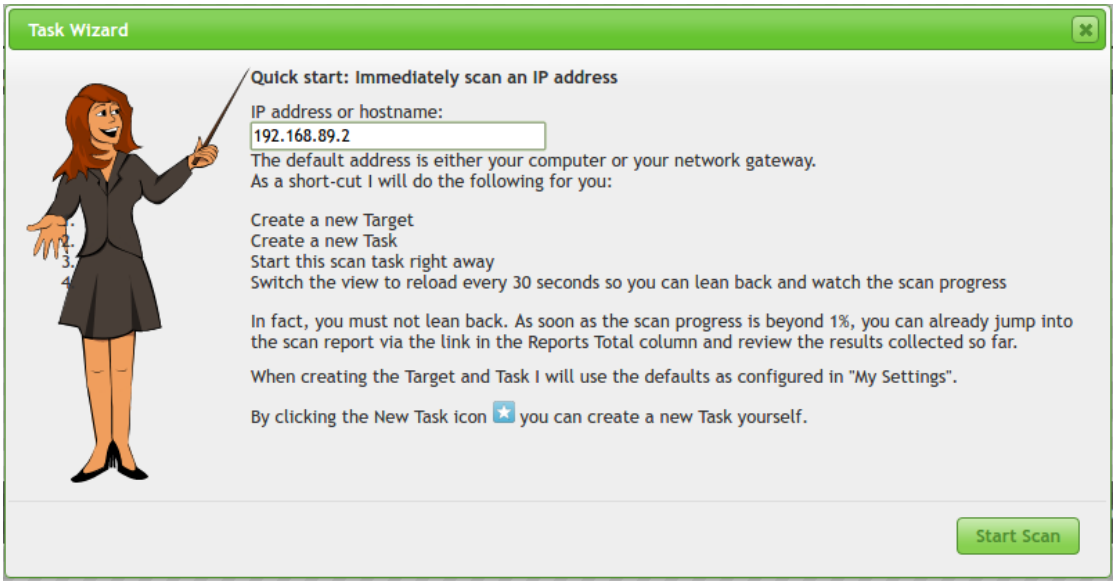


Illustration 9: OpenVAS Task Wizard

- You don't need to even wait for the scan to complete before looking at it.
  - ***Scans >> Reports***

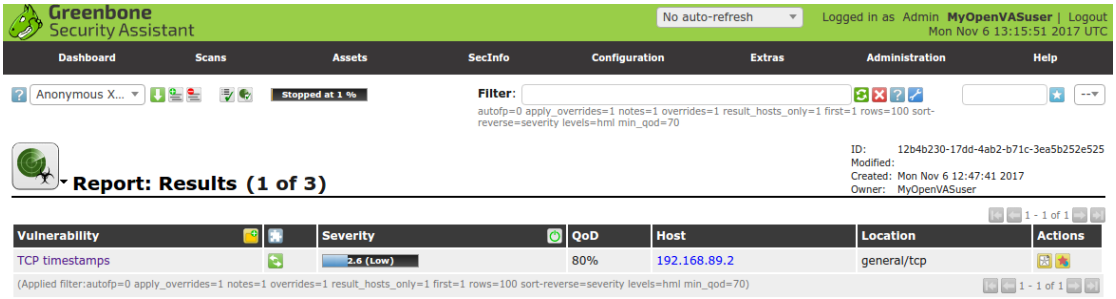


Illustration 10: OpenVAS post scan findings

- More detailed information can be gained from individual findings.

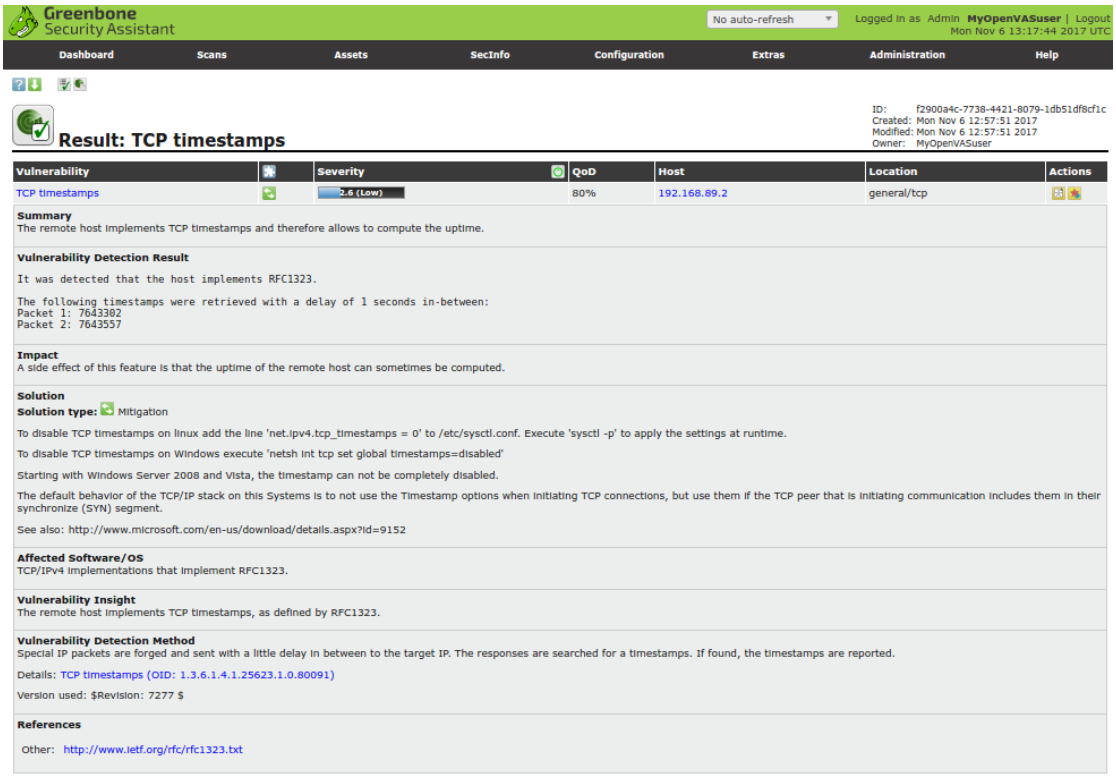


Illustration 11: OpenVAS detail

6.1.8 Stopping OpenVAS

To stop the OpenNAS server.

```
root@kali:~# openvas-stop
```

6.2 Metasploit

metasploit is a penetration testing framework from Rapid7 that enables you to find, exploit, and validate vulnerabilities.

```
root@kali:~# systemctl start postgresql


root@kali:~# msfdb init
A database appears to be already configured, skipping
initialization
```

It is important to update the Metasploit database regularly. There are typically updates weekly.

```
root@kali:~# apt update; apt install metaspoilt-framework
```

Control of Metasploit is through the ***msfconsole***.

```
root@kali:~# msfconsole

[*] Starting the Metaspoilt Framework Console ....

     ,           ,
    /             \
   ((__---,,,---__))
      (_) O O (_)_________
         \ _ /            |\
          o_o \   M S F   | \
               \   _____  |  *
                |||   WW|||
                |||     |||

       =[ metasploit v4.16.14-dev                         ]
+ -- --=[ 1699 exploits - 969 auxiliary - 299 post        ]
+ -- --=[ 503 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

Metasploit uses modules which are in effect other security tools like ***OpenVAS*** and ***Nessus***.

```
msf > load openvas
[*] Welcome to OpenVAS integration by kost and averagesecurityguy.
[*]
[*] OpenVAS integration requires a database connection. Once the
[*] database is ready, connect to the OpenVAS server using
openvas_connect.
[*] For additional commands use openvas_help.
[*]
[*] Successfully loaded plugin: OpenVAS
msf >
```

Each module has its own particular command line to manipulate it and establish a scan.

```
msf > openvas_help
[*] openvas_help                  Display this help
[*] openvas_debug                 Enable/Disable debugging
[*] openvas_version               Display the version of the OpenVAS server
[*]
[*] CONNECTION
[*] ==========
[*] openvas_connect               Connects to OpenVAS
[*] openvas_disconnect            Disconnects from OpenVAS
[*]
[*] TARGETS
[*] =======
[*] openvas_target_create         Create target
[*] openvas_target_delete         Deletes target specified by ID
[*] openvas_target_list           Lists targets
[*]
[*] TASKS
[*] =====
[*] openvas_task_create           Create task
[*] openvas_task_delete           Delete a task and all associated reports
[*] openvas_task_list             Lists tasks
[*] openvas_task_start            Starts task specified by ID
[*] openvas_task_stop             Stops task specified by ID
[*] openvas_task_pause            Pauses task specified by ID
[*] openvas_task_resume           Resumes task specified by ID
[*] openvas_task_resume_or_start  Resumes or starts task specified by ID
[*]
[*] CONFIGS
[*] =======
[*] openvas_config_list           Lists scan configurations
[*]
[*] FORMATS
[*] =======
[*] openvas_format_list           Lists available report formats
[*]
[*] REPORTS
[*] =======
[*] openvas_report_list           Lists available reports
[*] openvas_report_delete         Delete a report specified by ID
[*] openvas_report_import         Imports an OpenVAS report specified by ID
[*] openvas_report_download     Downloads an OpenVAS report specified by ID
```

6.3 Armitage

Armitage is a graphical cyber attack management tool for the Metasploit Framework that visualises targets and recommends exploits. Through ***Armitage***, a user may launch scans and exploits, get exploit recommendations, and use the advanced features of the Metasploit Framework.

Before starting ***Armitage*** the ***postgresql*** database must be running.

```
root@kali:~# systemctl start postgresql
```

If the ***Metaspoilt RPC Server*** is not running or accepting connections, ***armitage*** will start it before connecting to it. Simply click **Yes** at the prompt on the issue.

From another shell run **armitage**.

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

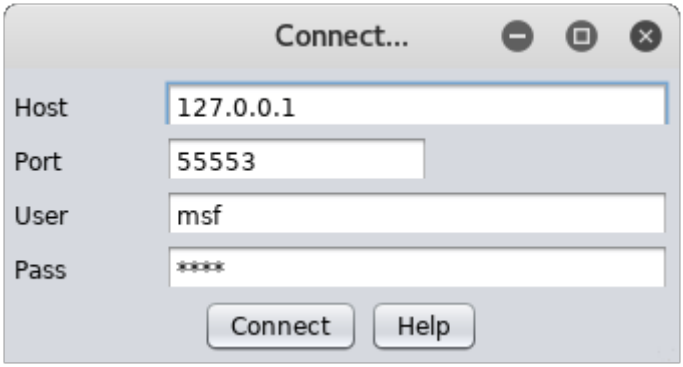


Illustration 12: Armitage connect to database

```
Start Metaspoilt? No | Yes : Yes
```

From the menu select:

Hosts → nmap Scan → Quick Scan (OS Detect)

Enter the IP addresses of the hosts that are to be scanned. For example a full range of IP address in the 192.168.89.0/24 subnet.

The system will scan and attempt to detect the Operating System of each using ***nmap***. It will display the discovered units in the top right window pane.

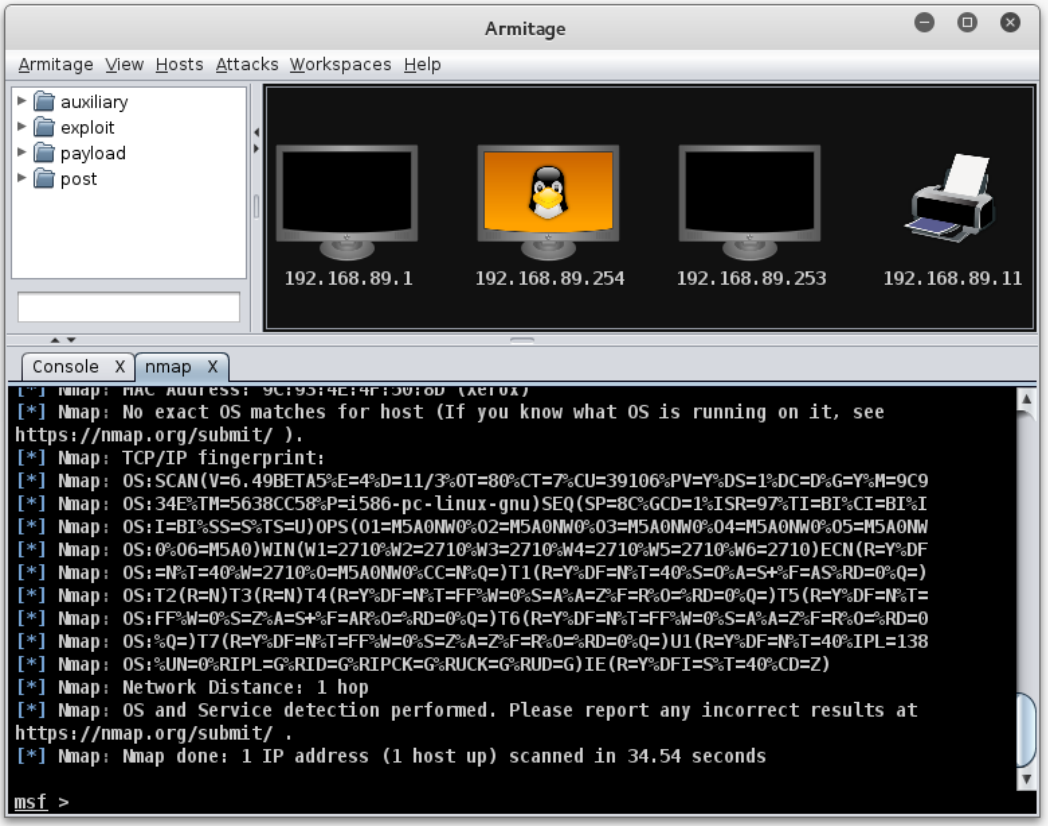


Illustration 13: Metaspoilt via Armitage

6.3.1 Scanning

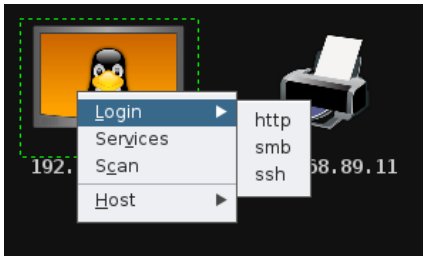


Illustration 14: Armitage, scanning

On any of the icons a scan can be carried out by right clicking and selecting ***Scan***. Or to perform for all hosts select:

Hosts → MSF Scans

When you right click now additional options will appear;

- ***Services*** if the device has services running on ports; and
- ***Login*** if login style services like SSH, Telnet, FTP or SMB are available.

6.3.2 Attack vectors

To build a set of attack vectors for each device select:

Attacks → Find Attacks

That will query exploits based on the services the scans have discovered.

A new menu will have appeared giving the potential exploit for each service.

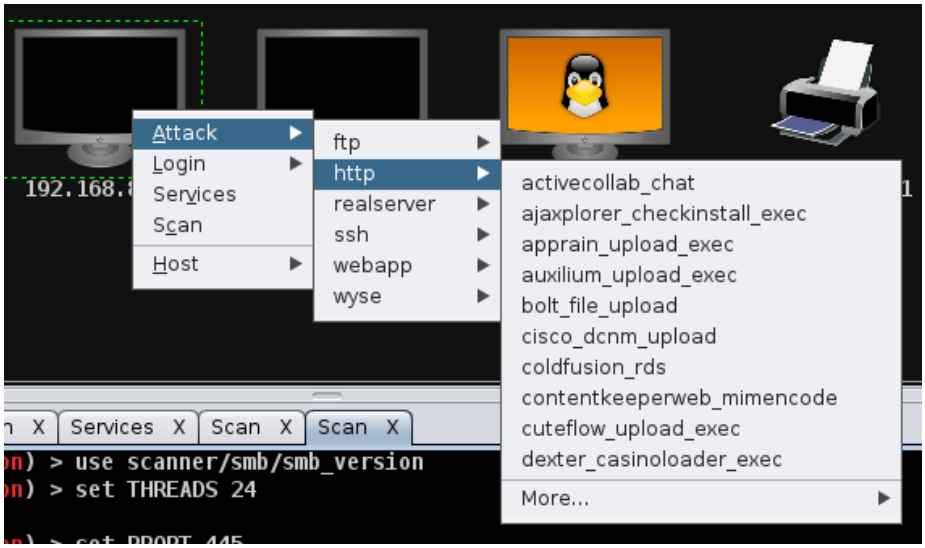


Illustration 15: Armitage, attack

6.3.3 Making the attack

Clicking on any of the potential attacks will give a detailed description of the attack and offer the option to add values like username, password, etc.. Click ***Launch*** to execute.

Illustration 16: Armitage, making the attack

6.3.4 Hail Mary attack

It is possible to flood a target with exploits. This is a clumsy attack and can potentially cause the target to crash.

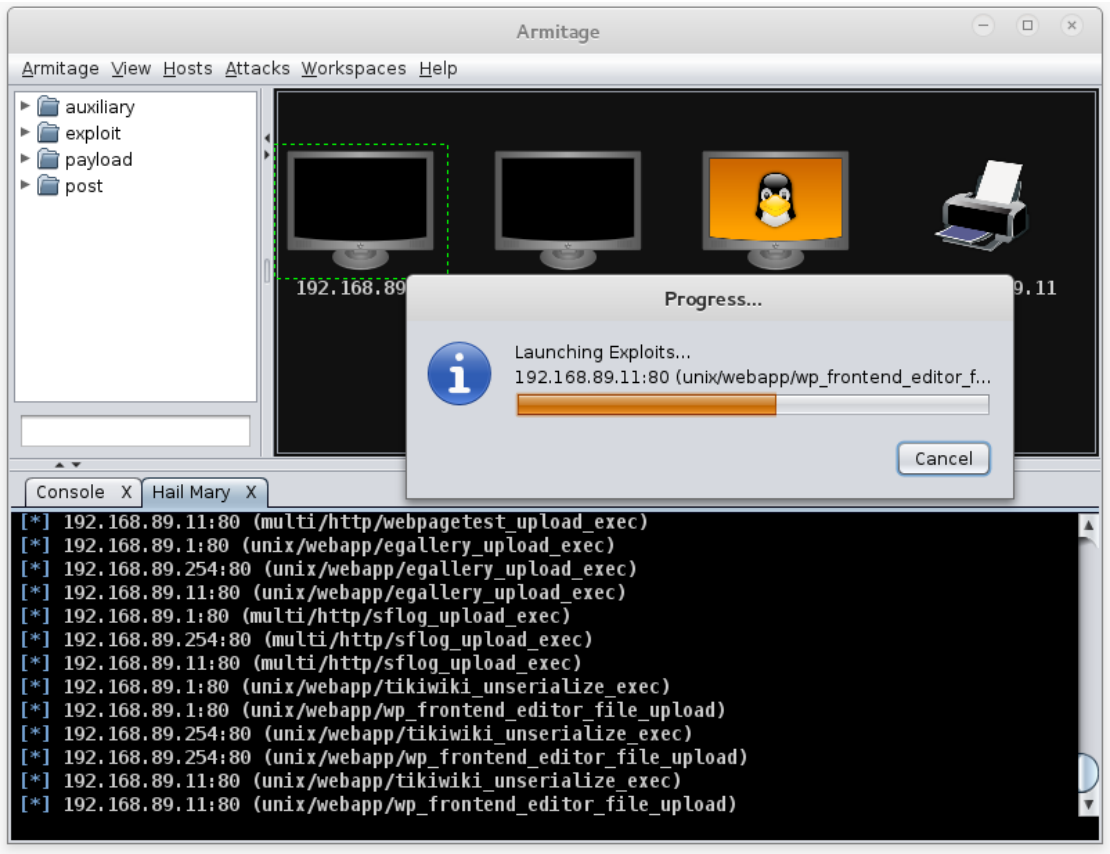


Illustration 17: Armitage, Hail Mary attack

6.3.5 Reporting

To access exploit reports select:

View → Reporting

This will give you direct access to the reports for each host as well as offer a the ability to download the reports in ***.csv*** format for spreadsheets.

host	port	state	proto	name	created_at	updated_at	info
192.168.89.1	21		tcp	ftp		1446562557662	220 MikroTik FTP server (MikroTik 6.0rc13) ready\x0d\x0a
192.168.89.1	22		tcp	ssh		1446562560331	SSH-2.0-ROSSSH
192.168.89.1	23		tcp	telnet		1446562606093	MikroTik v6.0rc13\x0aLogin:
192.168.89.1	80		tcp	http		1446562306810	
192.168.89.1	2000		tcp	bandwidth-test		1446562306824	MikroTik bandwidth-test server
192.168.89.254	22		tcp	ssh		1446562372449	SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.3
192.168.89.254	80		tcp	http		1446562369503	Apache/2.4.7 (Ubuntu)
192.168.89.254	139		tcp	netbios-ssn		1446562306916	Samba smbd 3.X workgroup: DOBRIAIN-THINKPAD-E550
192.168.89.254	445		tcp	smb		1446562375424	Unix (Samba 4.1.6-Ubuntu)
192.168.89.11	80		tcp	http		1446563548065	HTTP server (302-http://192.168.89.11/index.asp)
192.168.89.11	515		tcp	printer		1446562904812	
192.168.89.11	631		tcp	ipp		1446562904836	
192.168.89.11	9100		tcp	jetdirect		1446562904854	

Illustration 18: Armitage, .csv report

6.4 Testing Web Servers and Web Applications

6.5 Nikto

This is a shell utility to scan web servers for known vulnerabilities.

6.5.1 Install and update Nikto

Install ***nikto*** and before use it is important to update the plugins and databases directly from ***cirt.net***.

```
root@kali:~# nikto -update

+ Retrieving 'db_tests'
+ Retrieving 'db_variables'
+ Retrieving 'db_tests'
+ Retrieving 'db_outdated'
+ Retrieving 'db_server_msgs'
+ Retrieving 'nikto_robots.plugin'
+ Retrieving 'nikto_cookies.plugin'
+ Retrieving 'db_favicon'
+ Retrieving 'CHANGES.txt'
```

6.5.2 Running Nikto

Here is an example running the test against a host.

```
root@kali:~# nikto -host 192.168.89.1

- Nikto v2.1.4
---------------------------------------------------------------------------
+ Target IP:          192.168.89.1
+ Target Hostname:    192.168.89.1
+ Target Port:        80
+ Start Time:         2015-10-29 22:55:58
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ No CGI Directories found (use '-C all' to force check all possible
dirs)
+ robots.txt contains 1 entry which should be manually viewed.
+ 6456 items checked: 1 error(s) and 1 item(s) reported on remote
host
+ End Time:           2015-10-29 23:02:37 (399 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

6.6 Open Web Application Security Project (OWASP)[1]

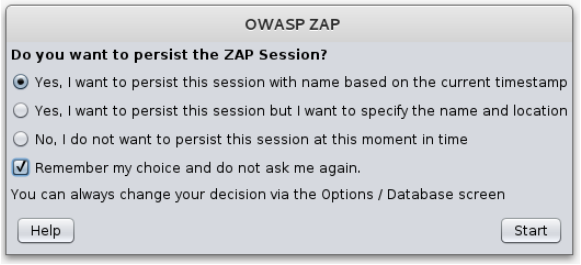
OWASP is an open community dedicated to enabling organisations to conceive, develop, acquire, operate, and maintain applications that can be trusted. All of the OWASP tools, documents, forums, and chapters are free and open to anyone interested in improving application security.

6.7 OWASP Zed Attack Proxy (ZAP)

The OWASP ZAP is an integrated penetration testing tool for finding vulnerabilities in web applications.

It can be used by developers and function test engineers to carry out penetration testing to identify and close vulnerabilities on their web developments.

```
root@kali:~# zaproxy
Found Java version 1.7.0_79
Available memory:  2021 MB
Setting jvm heap size: -Xmx512m
```



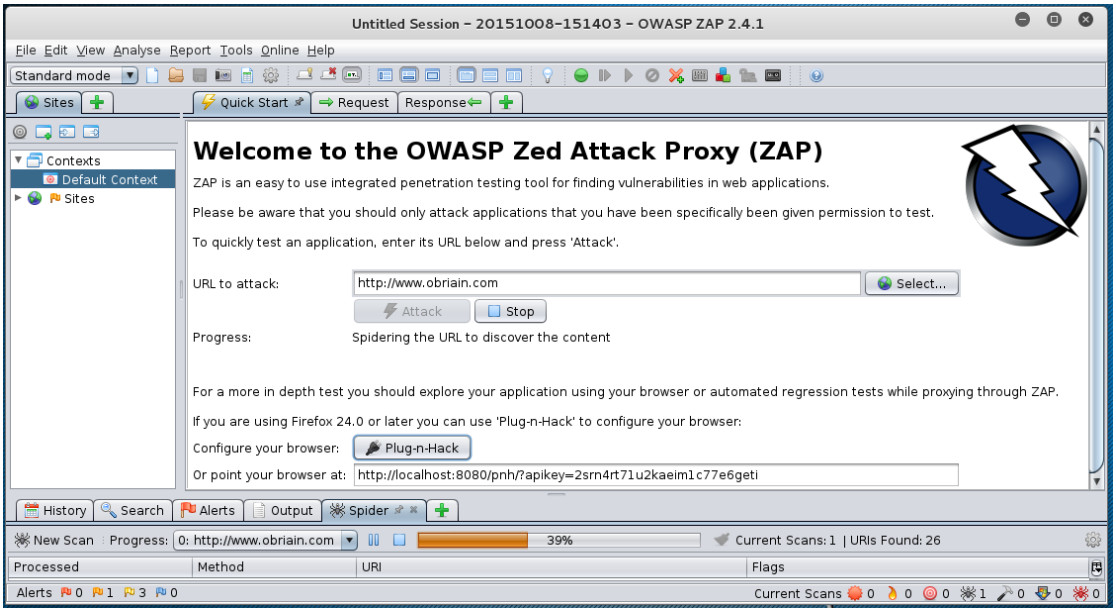


Illustration 19: Zed Attack Proxy (zap)

1 OWASP https://www.owasp.org

When the attack is complete a list of alerts are displayed for the attack vector and any links spidered from it on the site. For each alert it proposes a solution.

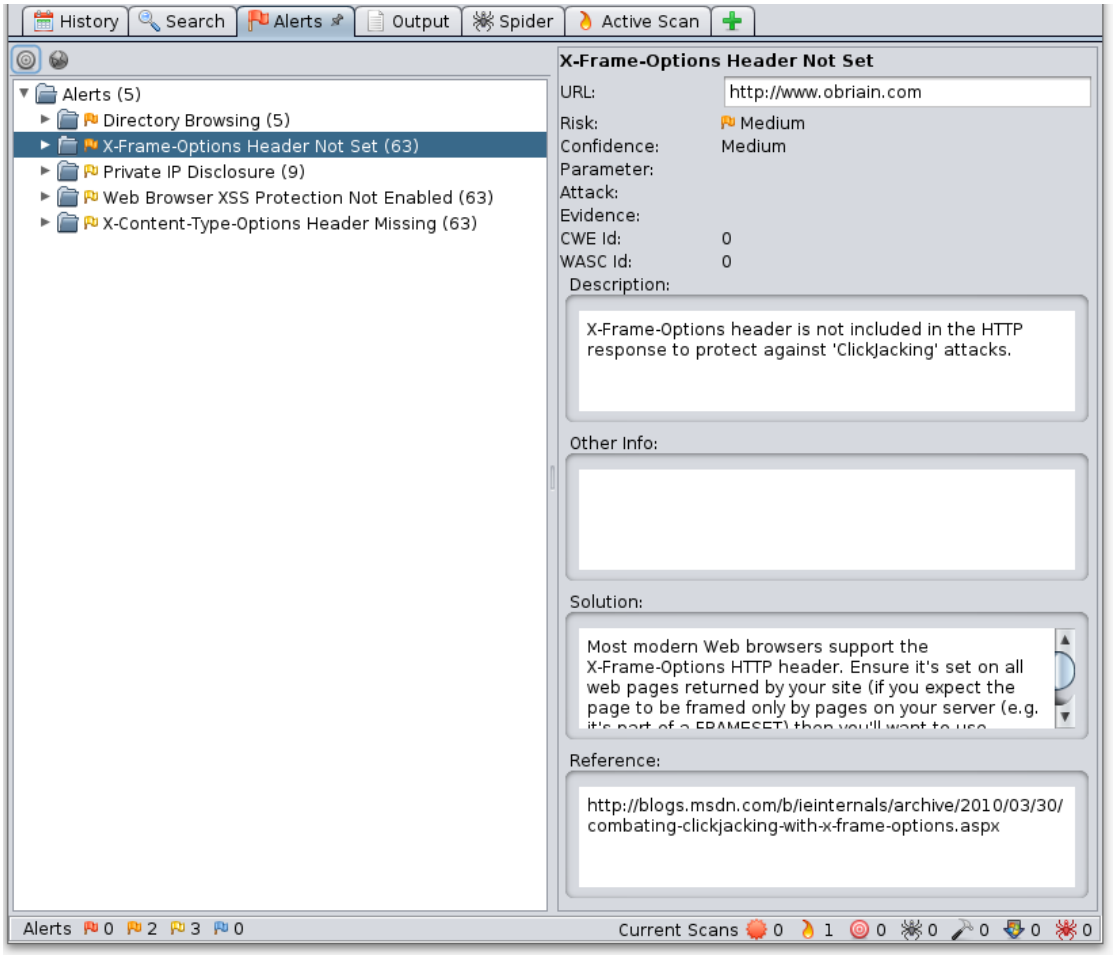


Illustration 20: Zap post scan alerts

6.8 Reporting

Zap has an excellent reporting tool. Simply select Report from the top toolbar and once can be generated in a number of formats. Here is an example of the HTML formatted report.

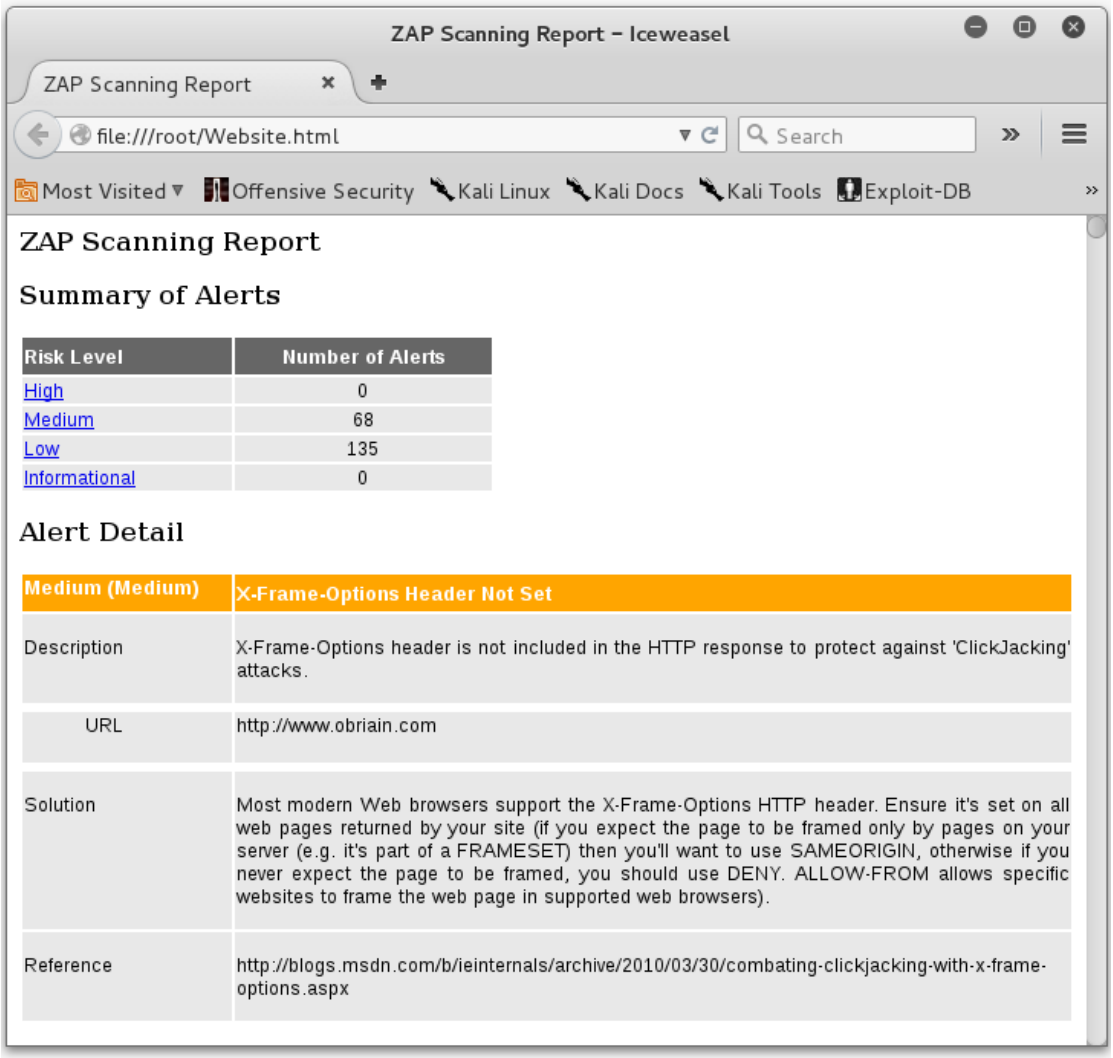


Illustration 21: Zap reporting

7. Detection Systems

7.1 p0f

p0f is a passive OS fingerprinting tool. ***p0f*** uses a fingerprinting technique based on analysing the structure of a TCP/IP packet to determine the operating system and other configuration properties of a remote host.

Install ***p0f*** on a server as follows:

```
ada:~$ sudo p0f -i eth0 -do /tmp/p0f-output.txt
```

Run the p0f server to monitor the Ethernet interface and output results to a file. It runs in daemon mode in the background.

- **-i** Interface
- **-d** Daemon mode, Fork in the background
- **-o** Output file

```
ada:~$ sudo p0f -i eth0 -do /tmp/p0f-output.txt
--- p0f 3.07b by Michal Zalewski <lcamtuf@coredump.cx> ---

[!] Consider specifying -u in daemon mode (see README).
[+] Closed 1 file descriptor.
[+] Loaded 320 signatures from 'p0f.fp'.
[+] Intercepting traffic on interface 'eth0'.
[+] Default packet filtering configured [+VLAN].
[+] Log file '/tmp/p0f-output.txt' opened for writing.
[+] Daemon process created, PID 3191 (stderr not kept).

Good luck, you're on your own now!

ada:~$ tail /tmp/p0f-output.txt
[2015/11/03 03:59:41] mod=syn|cli=10.0.2.15/51461|
srv=192.168.89.1/50501|subj=cli|app=NMap SYN scan|dist=<= 21|
params=random_ttl|raw_sig=4:43+21:0:1460:1024,0:mss::0
[2015/11/03 03:59:41] mod=syn|cli=10.0.2.15/51461|
srv=192.168.89.1/57509|subj=cli|app=NMap SYN scan|dist=<= 8|
params=random_ttl|raw_sig=4:56+8:0:1460:1024,0:mss::0
[2015/11/03 03:59:41] mod=syn|cli=10.0.2.15/51461|
srv=192.168.89.1/40296|subj=cli|app=NMap SYN scan|dist=<= 9|
params=random_ttl|raw_sig=4:55+9:0:1460:1024,0:mss::0
[2015/11/03 03:59:41] mod=syn|cli=10.0.2.15/51462|
srv=192.168.89.1/57509|subj=cli|app=NMap SYN scan|dist=<= 20|
params=random_ttl|raw_sig=4:44+20:0:1460:1024,0:mss::0
[2015/11/03 03:59:41] mod=syn|cli=10.0.2.15/51461|
srv=192.168.89.1/63300|subj=cli|app=NMap SYN scan|dist=<= 25|
params=random_ttl|raw_sig=4:39+25:0:1460:1024,0:mss::0
```

In this example the ***p0f*** utility detected an ***nmap*** scan.

This scan continues in the background filling the output file until you stop it. To finish the scan. List the current processes and ***grep*** for those with ***p0f*** in the name (***-e*** *= All processes,* ***-f*** *= Perform full format listing*). Returned is the ***p0f*** daemon that was ran plus the grep process established in the command to find ***p0f***.

```
ada:~$ ps -ef| grep p0f
root  3191  1  0 03:55 ?   00:00:00 ./p0f -i eth0 -do /tmp/p0f-output.txt
root  3218  3138  0 04:02 pts/1  00:00:00 grep p0f
```

Send the daemon via its process ID the SIGKILL signal. This terminates the daemon. A ***grep*** of the processes confirms this.

```
ada:~$ kill -SIGKILL 3191
ada:~$ ps -ef | grep p0f
root      3231  3138  0 04:06 pts/1    00:00:00 grep p0f
```

7.2 Port Scan Attack Detector (psad)

The Port Scan Attack Detector (***psad***) makes use of ***iptables*** log messages from the ***/var/log/messages*** file to detect, alert, and optionally block port scans and other suspect traffic.

Variables can be adjusted in the ***/etc/psad/psad.conf***. In the example below ***psad*** detects an ***nmap*** port scan from ***86.140.55.1***.

```
ada:~$ sudo apt install psad
Setting up psad (2.2-3.1) ...
[ ok ] Starting Port Scan Attack Detector: psad.
```

Set the IP Tables logging rules.

```
ada:~$ sudo iptables -F

ada:~$ sudo iptables -A INPUT -j LOG

ada:~$ sudo iptables -A FORWARD -j LOG

ada:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -j LOG
-A FORWARD -j LOG
```

Update ***psad*** signatures.

```
ada:~$ sudo psad –sig-update

ada:~$ sudo service psad restart

[info] Stopping the psadwatchd process.
[info] Stopping the kmsgsd process.
[info] Stopping the psad process.
[ ok ] Stopping Port Scan Attack Detector: psad.
[ ok ] Starting Port Scan Attack Detector: psad.
```

Check the status of ***psad***.

```
ada:~$ sudo service psad status

Status of Port Scan Attack Detector:
[+] psadwatchd (pid: 2887)  %CPU: 0.0  %MEM: 0.0
    Running since: Thu Jul  3 22:25:59 2014

[+] psad (pid: 2885)  %CPU: 1.4  %MEM: 3.0
    Running since: Thu Jul  3 22:25:59 2014
    Command line arguments: [none specified]
    Alert email address(es): root@localhost

[+] Version: psad v2.2

[+] Top 50 signature matches:
      "DDOS Trin00 Master to Daemon default password attempt"
      (udp), Count: 4,  Unique sources: 1,  Sid: 237
      "MISC Microsoft PPTP communication attempt" (tcp),  Count: 2,

      Unique sources: 1,  Sid: 100082
      "ICMP PING" (icmp),  Count: 1,  Unique sources: 1,  Sid: 384
      "ICMP traceroute" (icmp),  Count: 1,  Unique sources: 1,
      Sid: 385

[+] Top 25 attackers:
      86.140.55.1     DL: 3, Packets: 489, Sig count: 8
      78.143.141.200  DL: 2, Packets: 46, Sig count: 0

[+] Top 20 scanned ports:
      tcp 80    118 packets
      tcp 25    4 packets
      tcp 1723  2 packets
      tcp 21071 1 packets
      tcp 34978 1 packets
      tcp 143   1 packets
      tcp 9088  1 packets
      tcp 9443  1 packets

      udp 27892 9 packets
      udp 26415 9 packets
      udp 28543 8 packets
```

```
      udp 22124 8 packets
      udp 30544 8 packets
      udp 22123 6 packets
      udp 21698 6 packets
      udp 27482 6 packets
      udp 32779 6 packets
      udp 123   6 packets
      udp 24511 6 packets
      udp 24007 5 packets
      udp 32818 5 packets
      udp 25546 5 packets
      udp 31189 5 packets
      udp 30303 5 packets
      udp 34358 5 packets
      udp 32931 5 packets
      udp 36893 5 packets
      udp 21525 5 packets

[+] iptables log prefix counters:
        [NONE]

    Total packet counters: tcp: 129 udp: 408 icmp: 1

[+] IP Status Detail:

SRC:  86.140.55.1, DL: 3, Dsts: 1, Pkts: 489, Unique sigs: 2, Email
alerts: 5

DST: 192.168.89.1, Local IP Scanned ports: UDP 123-58178, Pkts:
359, Chain: INPUT, Intf: eth0 Scanned ports: TCP 25-34978, Pkts:
129, Chain: INPUT, Intf: eth0 Signature match: "MISC Microsoft PPTP
communication attempt" TCP, Chain: INPUT, Count: 1, DP: 1723, SYN,
Sid: 100082 Signature match: "DDOS Trin00 Master to Daemon default
password attempt" UDP, Chain: INPUT, Count: 1, DP: 27444, Sid: 237

SRC:  78.143.141.200, DL: 2, Dsts: 1, Pkts: 46, Unique sigs: 0,
Email alerts: 4

DST: 192.168.89.1, Local IP Scanned ports: UDP 34114-60963, Pkts:
46, Chain: INPUT, Intf: eth0

    Total scan sources: 2
    Total scan destinations: 1

[+] These results are available in: /var/log/psad/status.out
```

```
ada:~$ sudo tail -f /var/log/psad/status.out
UDP, Chain: INPUT, Count: 1, DP: 27444, Sid: 237

SRC:  78.143.141.200, DL: 2, Dsts: 1, Pkts: 46, Unique sigs: 0,
Email alerts: 4

DST: 192.168.89.1, Local IP Scanned ports: UDP 34114-60963, Pkts:
46, Chain: INPUT, Intf: eth0

    Total scan sources: 2
    Total scan destinations: 1
```

7.3 Passive Asset Detection System (pads)

Passive Asset Detection System (***pads***) is a libpcap based detection engine used to passively detect network assets. It is designed to complement IDS technology by providing context to IDS alerts. Discovered devices are logged in ***/var/lib/pads/assets.csv***. This can be changed along with many other variables in ***/etc/pads/pads.conf***.

```
ada:~$ sudo apt install pads

Setting up pads (1.2-11) ...
[ ok ] Starting Passive Asset Detection System: pads.

ada:~$ cat /var/lib/pads/assets.csv
asset,port,proto,service,application,discovered
109.106.96.153,0,0,ARP (Intel Corporation), 0:04:23:B1:8F:E2,
1404421526
```

8. Summary

This document introduces penetration testing and Kali Linux as a tool for such activity. It has only skimmed the surface as you should realise just browsing the menus of the Kali Linux applications tab.

To become proficient at pen-testing takes practice.

9. Lab Exercise

Carry out a pen-test on the IP address given to you by the instructor.

10. Bibliography

Payment Card Industry - Data Security Standard (PCI DSS), Penetration Testing Guidance, Version 1.0, March 2015.

Payment Card Industry - Data Security Standard (PCI DSS), Requirements and Security Assessment Procedures, Version 3.2, April 2016.

Karen Scarfone, Murugiah Souppaya, Amanda Cody and Angela Orebaugh (2015). Technical Guide to Information Security Testing and Assessment. National Institute of Standards and Technology Special Publication 800-115.

NIST (2014). Assessing Security and Privacy Controls in Federal Information Systems and Organizations - Building Effective Assessment Plans. National Institute of Standards and Technology Special Publication 800-53A Revision 4.

NIST (2017). Verification and Test Methods for Access Control Policies/Models. National Institute of Standards and Technology Special Publication 800-184.

NIST (2016). Guide for Cybersecurity Event Recovery. National Institute of Standards and Technology Special Publication 800-192.

Tor Project: Anonymity Online [online]. Available: https://www.torproject.org

Kali GNU/Linux distribution. Offensive Security [online]. Available: https://www.kali.org

Nmap: the Network Mapper - Free Security Scanner [online]. Available: https://nmap.org

Zenmap - Official cross-platform Nmap Security Scanner GUI [online]. Available: https://nmap.org/zenmap/

OpenVAS - Open Source vulnerability scanner and manager [online]. Available: http://www.openvas.org

Metasploit Unleashed (MSFU). Offensive Security [online]. Available: https://www.offensive-security.com/metasploit-unleashed/

Armitage - Cyber Attach Managem,ent for Metasploit [online]. Available: http://www.fastandeasyhacking.com

OWASP Zed Attack Proxy Project [online]. Available: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

Michal Zalewski (2014). p0f v3 (version 3.09b) [online]. Available: http://lcamtuf.coredump.cx/p0f3/

Port Scan Attack Detector (PSAD): Intrusion Detection and Log Analysis with iptables [online]. Available: http://cipherdyne.org/psad/

Passive Asset Detection System (PADS) [online]. Available: http://passive.sourceforge.net

This page is intentionally blank