

BSc in Telecommunications Engineering

TEL3214

Computer Communication Networks

Lecture 10

Network Security

Eng Diarmuid O'Briain, CEng, CISSP



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

Copyright © 2017 Diarmuid Ó Briain

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

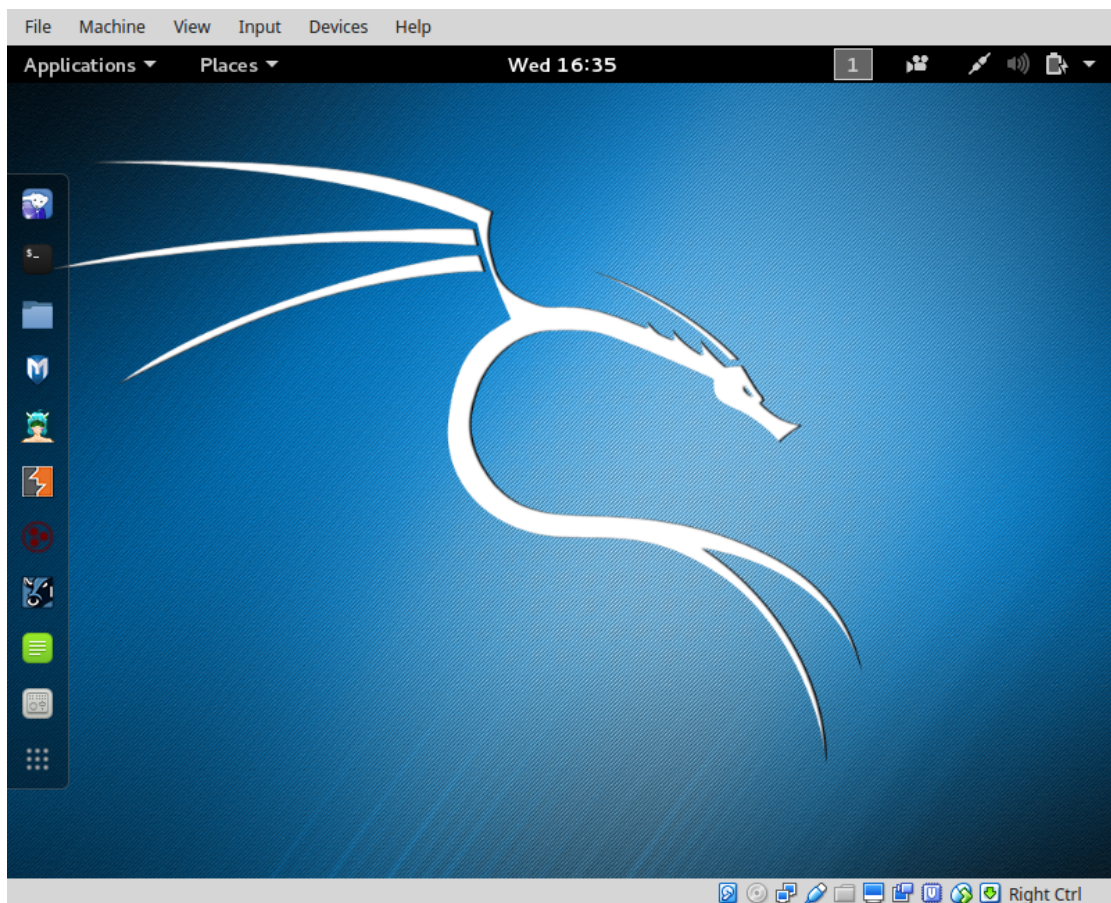
Table of Contents

| | |
|---|-----------|
| 1. LAB EXERCISE..... | 5 |
| 2. NETWORK SECURITY AND PENETRATION TESTING..... | 8 |
| 2.1 PENETRATION TESTING STEPS..... | 8 |
| 3. KALI LINUX..... | 10 |
| 3.1 ROOT USER..... | 10 |
| 3.2 SYSTEM UPDATE..... | 10 |
| 4. INFORMATION GATHERING AND ANALYSIS..... | 11 |
| 4.1 FIERCE..... | 11 |
| 4.2 NMAP..... | 12 |
| 4.3 USE NMAP ANONYMOUSLY..... | 12 |
| 4.4 ZENMAP..... | 16 |
| 5. VULNERABILITY DETECTION AND EXPLOITATION..... | 17 |
| 5.1 OPENVAS..... | 17 |
| 5.2 METASPLOIT..... | 22 |
| 5.3 ARMITAGE..... | 24 |
| 5.4 TESTING WEB SERVERS AND WEB APPLICATIONS..... | 28 |
| 5.5 NIKTO..... | 28 |
| 5.6 OPEN WEB APPLICATION SECURITY PROJECT (OWASP)1..... | 29 |
| 5.7 OWASP ZED ATTACK PROXY (ZAP)..... | 29 |
| 5.8 REPORTING..... | 31 |
| 6. DETECTION SYSTEMS..... | 32 |
| 6.1 P0F..... | 32 |
| 6.2 PORT SCAN ATTACK DETECTOR (PSAD)..... | 33 |
| 6.3 PASSIVE ASSET DETECTION SYSTEM (PADS)..... | 36 |
| 7. SUMMARY..... | 36 |
| 8. LAB EXERCISE..... | 36 |

This page is intentionally blank

1. Lab Exercise

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

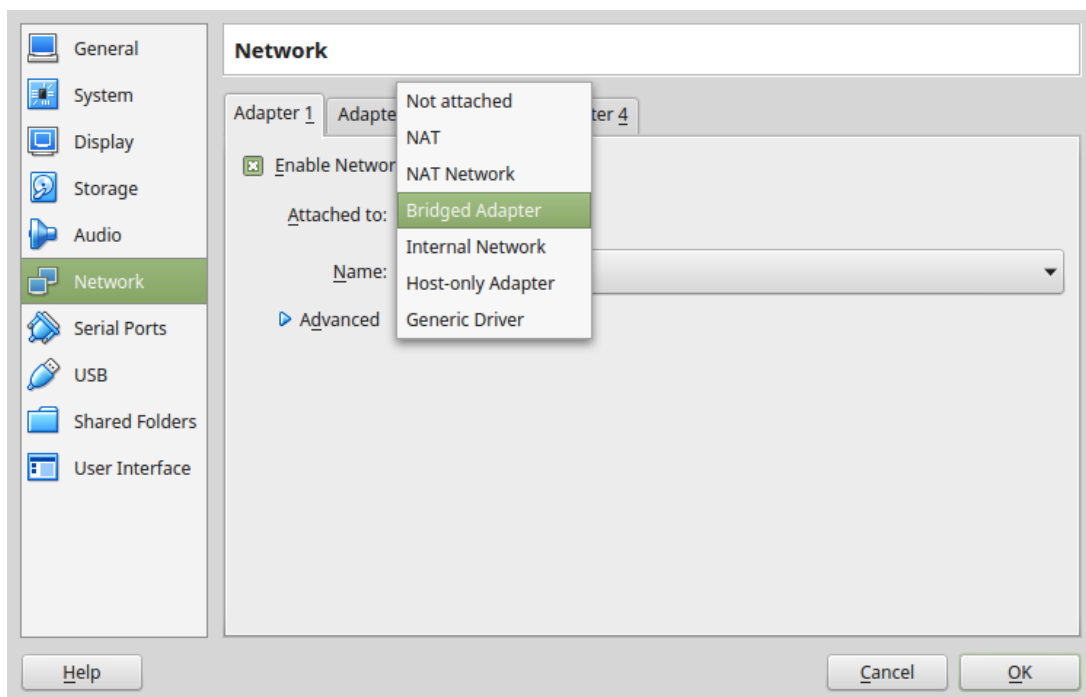


Login to the image with the default root username (**root**) and password (**toor**).

Run up a shell and confirm connectivity with the Internet.

```
root@kali:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    group default
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    state UP group default qlen 1000
        link/ether 08:00:27:1a:02:bd brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
            valid_lft 86044sec preferred_lft 86044sec
        inet6 fe80::a00:27ff:fe1a:2bd/64 scope link
            valid_lft forever preferred_lft forever
```

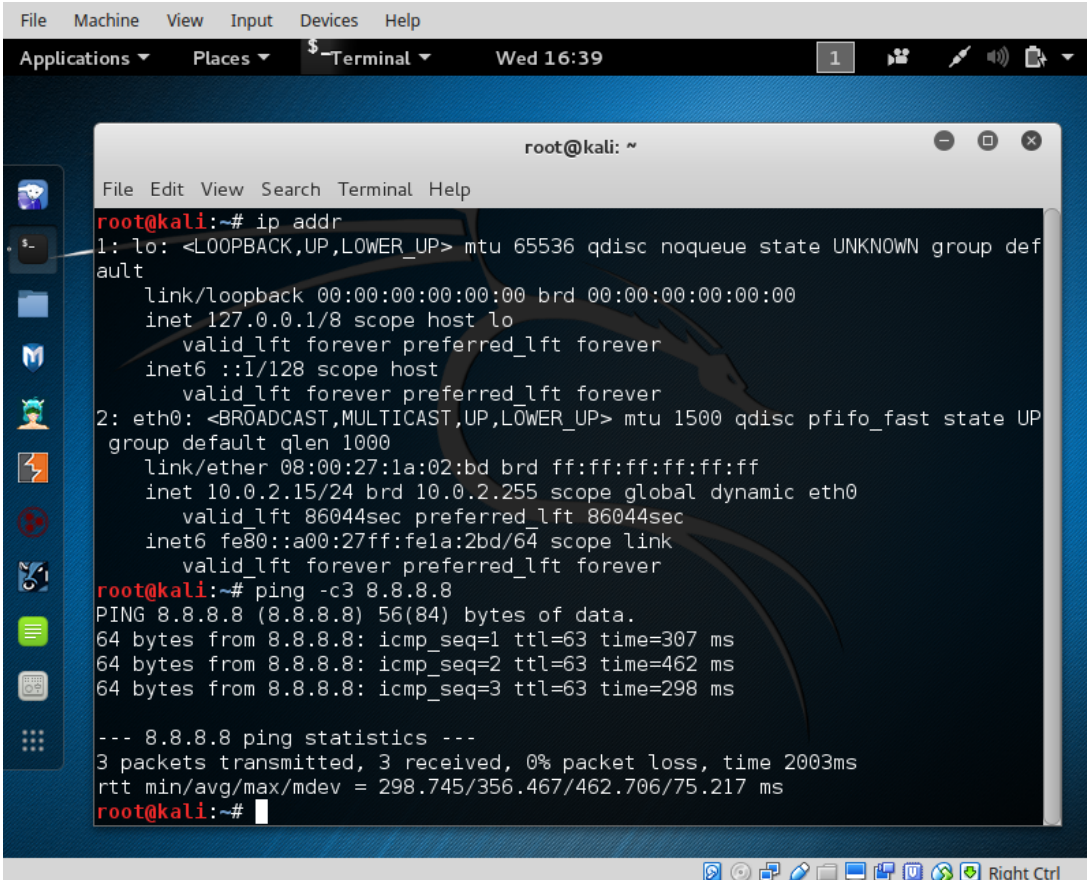
The IP Address is assigned by Network Address Translation (NAT) to the VM. It is possible to bridge the VM Ethernet interface (eth0) with the active interface on the host to get an IP address from the real world Dynamic Host Configuration Protocol (DHCP) Server.



Whichver 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:~#
```

A screenshot of a Kali Linux desktop environment. The desktop background is a dark blue wallpaper with a dragon-like creature. On the left side, there is a vertical dock with several application icons. The top of the screen features a menu bar with 'File', 'Machine', 'View', 'Input', 'Devices', and 'Help'. Below the menu bar is a panel with 'Applications', 'Places', and a terminal icon. The terminal window is open, showing the command prompt 'root@kali: ~'. The terminal output displays the results of the 'ip addr' command for the loopback interface 'lo' and the ethernet interface 'eth0'. It then shows the output of the 'ping -c3 8.8.8.8' command, which successfully pings the Google DNS server at 8.8.8.8 three times. Finally, it shows the 'ping statistics' for the test, indicating 3 packets transmitted, 3 received, and 0% packet loss. The terminal window has a title bar that says 'root@kali: ~' and standard window control buttons (minimize, maximize, close). The bottom of the screen shows a taskbar with various system icons and a 'Right Ctrl' button.

2. 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.

2.1 Penetration testing steps

2.1.1 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.

2.1.2 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?

2.1.3 Vulnerability detection

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

2.1.4 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.

2.1.5 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.

2.1.6 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..

3. 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.

3.1 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.

3.2 System update

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

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

4. 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.

4.1 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.
```

4.2 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.

```
cedat:~$ sudo apt-get 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
```

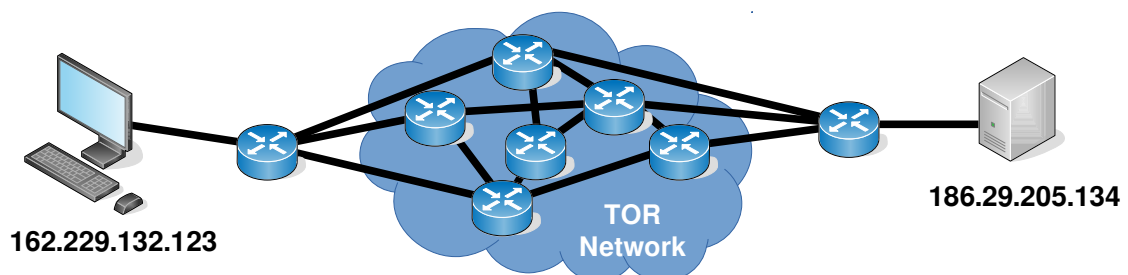
```
Starting Nmap 6.40 ( http://nmap.org ) at 2015-11-03 11:41 EAT
Nmap scan report for 192.168.89.1
Host is up (0.00086s latency).
Not shown: 65530 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
2000/tcp  open  cisco-sccp
8291/tcp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

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

4.3 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

```
cedat:~$ sudo apt-get install tor proxychains
```



Here is an Nmap scan thorough 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. Bacidally Nmap is making use of the OS own Berkeley Socket API.

```
cedat:~$ 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.

```
cedat:~$ 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.

```
cedat:~$ 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).
```



```
cedat:~$ 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
```

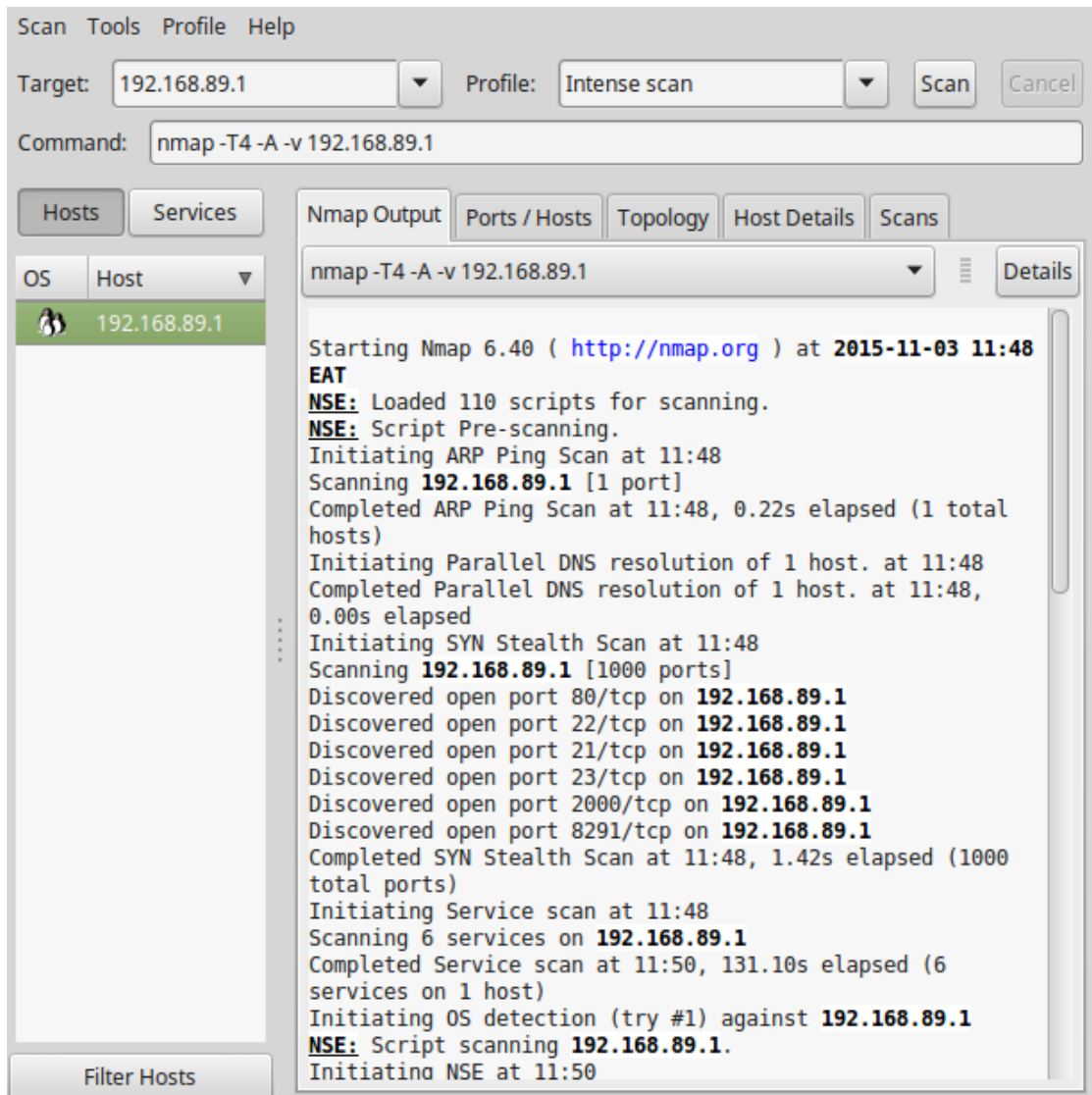
Each time the source is a different address as the exit point from TOR changes.

| IP ADDRESS INFORMATION | |
|------------------------|--|
| IP Address | 43.229.53.25 |
| Hostname | 43.229.53.25 |
| Network | Asia Pacific Network Information Centre |
| Country |  JP - JAPAN |
| Latitude | 36 |
| Longitude | 138 |
| IP Range | 43.0.0.0 - 43.233.35.255 |
| IP Network | American Registry for Internet Numbers (ARIN) |

| IP ADDRESS INFORMATION | |
|------------------------|---|
| IP Address | 81.7.15.115 |
| Hostname | 81-7-15-115.blue.kundencontroller.de |
| Network | RIPE Network Coordination Centre |
| Country |  DE - GERMANY |
| Latitude | 51 |
| Longitude | 9 |
| IP Range | 81.7.0.0 - 81.7.63.255 |
| IP Network | American Registry for Internet Numbers (ARIN) |

4.4 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.



5. Vulnerability Detection and Exploitation

5.1 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 35,000 in total.

5.1.1 Create an OpenVAS SSL Certificate

Create an OpenVAS SSL Certificate.

```
root@kali:/# openvas-mkcert -f
```

```
-----  
                        Creation of the OpenVAS SSL Certificate  
-----
```

This script will now ask you the relevant information to create the SSL certificate of OpenVAS.

Note that this information will ***NOT*** be sent to anybody (everything stays local), but anyone with the ability to connect to your OpenVAS daemon will be able to retrieve this information.

```
CA certificate life time in days [1460]:  
Server certificate life time in days [365]:  
Your country (two letter code) [DE]: UG  
Your state or province name [none]:  
Your location (e.g. town) [Berlin]: Kampala  
Your organization [OpenVAS Users United]: Makerere
```

Congratulations. Your server certificate was properly created.

The following files were created:

```
. Certification authority:  
  Certificate = /var/lib/openvas/CA/cacert.pem  
  Private key = /var/lib/openvas/private/CA/cakey.pem  
  
. OpenVAS Server :  
  Certificate = /var/lib/openvas/CA/servercert.pem  
  Private key = /var/lib/openvas/private/CA/serverkey.pem
```

Press [ENTER] to exit

5.1.2 client certificate file of OpenVAS Manager

Create an SSL client certificate with the **-i** option to install client certificates for use with OpenVAS manager.

```
root@kali:/# openvas-mkcert-client -i
```

This script will now ask you the relevant information to create the SSL client certificates for OpenVAS.

```
Client certificates life time in days [365]:
Your country (two letter code) [DE]: UG
Your state or province name [none]:
Your location (e.g. town) [Berlin]: Kampala
Your organization [none]: Makerere
Your organizational unit [none]:
*****
```

We are going to ask you some question for each client certificate.

If some question has a default answer, you can force an empty answer by entering a single dot '.'

```
*****
```

```
Client certificates life time in days [365]:
Country (two letter code) [UG]:
State or province name []:
Location (e.g. town) [Kampala]:
Organization [Makerere]:
Organization unit []:
e-Mail []:
Generating RSA private key, 4096 bit long modulus
.....++
.....++
e is 65537 (0x10001)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [DE]:State or Province Name (full name) [Some-
State]:Locality Name (eg, city) []:Organization Name (eg, company) [Internet
Widgits Pty Ltd]:Organizational Unit Name (eg, section) []:Common Name (eg,
your name or your server's hostname) []:Email Address []:Using configuration
from /tmp/openvas-mkcert-client.9696/stdC.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName          :PRINTABLE:'UG'
localityName         :ASN.1 12:'Kampala'
organizationName     :ASN.1 12:'Makerere'
commonName           :ASN.1 12:'om'
Certificate is to be certified until Nov  2 11:13:51 2016 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
```

5.1.3 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
```

5.1.4 OpenVAS setup

As an initial step on the first occasion to run OpenVAS run the setup tool. This script synchronises with the OpenVAS collection of Network Vulnerability Tests (NVTs). As it needs to upload all the NVTs on record this can take some time.

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

Once up and running it is important to regularly sync with the OpenVAS NVT database. To carry out that task:

```
root@kali:~# openvas-nvt-sync
```

5.1.5 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-8 installation is OK.
```

5.1.6 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 10409/openvasmd
tcp      0      0 127.0.0.1:9391   0.0.0.0:*        LISTEN 10383/openvassd: Wa
tcp      0      0 127.0.0.1:9392   0.0.0.0:*        LISTEN 10414/gsad
```

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

5.1.7 Using the web client

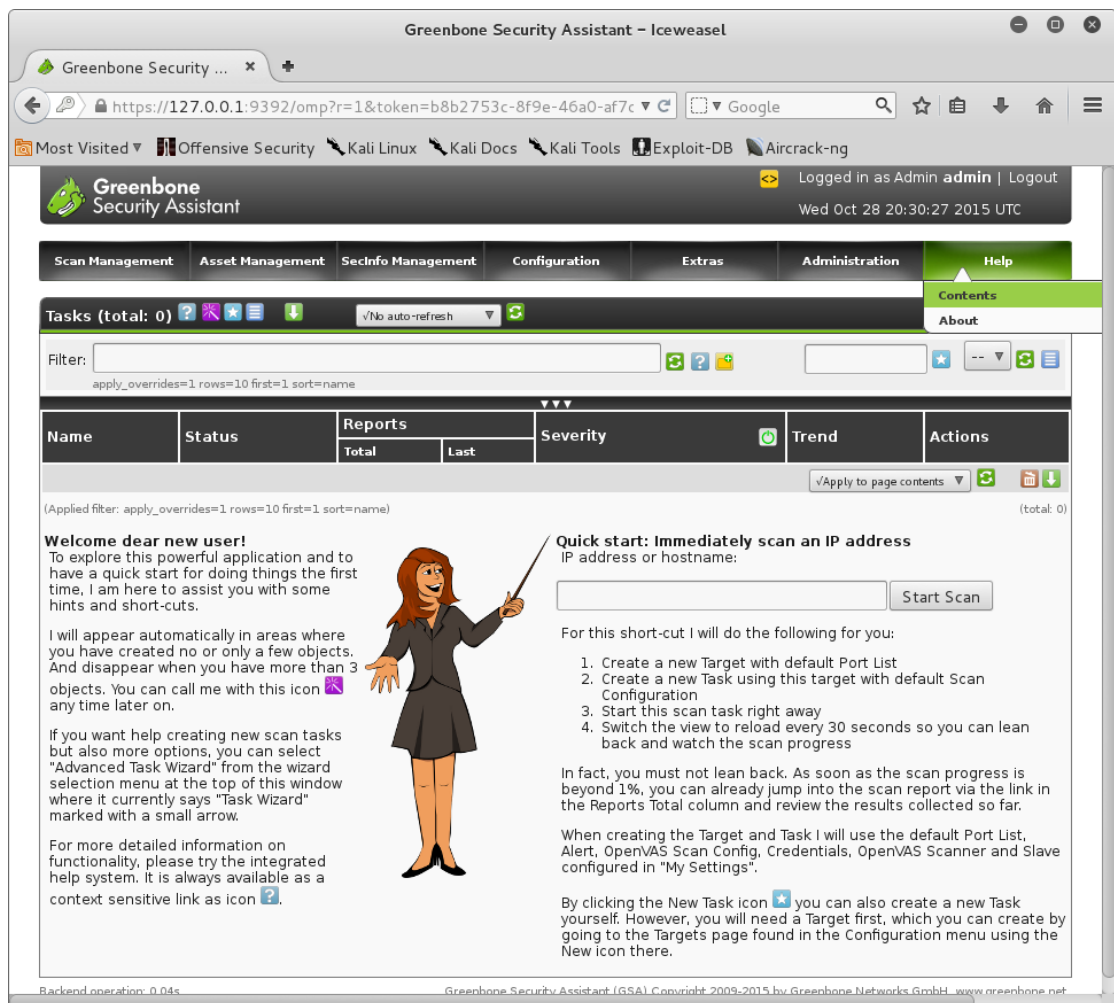
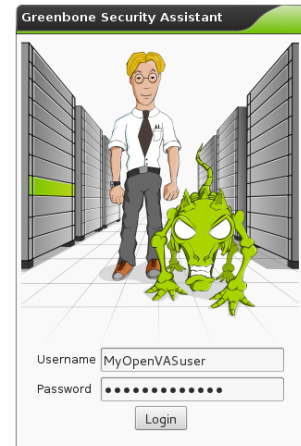
Note the webclient will only work to **https://** not **http://**

https://127.0.0.1:9392

Use the Username and Password created above.

Username: **MyOpenVASuser**

Password: **MyOpenVAsspass**



- Define a target and click **Start Scan**.
- You don't need to even wait for the scan to complete before looking at it.

Report: Results 1 - 14 of 14 (total: 14) PDF 52 %

Filter: sort-reverse=severity result_hosts_only=1 min_cvss_base= min_qod=70

| Vulnerability | Severity | QoD | Host | Location | Actions |
|---|-----------|-----|--------------|-------------|---------|
| OS fingerprinting | 0.0 (Log) | 70% | 192.168.89.1 | general/tcp | |
| FTP Banner Detection | 0.0 (Log) | 80% | 192.168.89.1 | 21/tcp | |
| Services | 0.0 (Log) | 75% | 192.168.89.1 | 21/tcp | |
| SSH Protocol Versions Supported | 0.0 (Log) | 95% | 192.168.89.1 | 22/tcp | |
| SSH Server type and version | 0.0 (Log) | 80% | 192.168.89.1 | 22/tcp | |
| Services | 0.0 (Log) | 75% | 192.168.89.1 | 22/tcp | |
| Detect Server type and version via Telnet | 0.0 (Log) | 80% | 192.168.89.1 | 23/tcp | |
| Services | 0.0 (Log) | 75% | 192.168.89.1 | 23/tcp | |
| Services | 0.0 (Log) | 75% | 192.168.89.1 | 80/tcp | |
| Web mirroring | 0.0 (Log) | 80% | 192.168.89.1 | 80/tcp | |
| Directories used for CGI Scanning | 0.0 (Log) | 75% | 192.168.89.1 | 80/tcp | |
| wapiti (NASL wrapper) | 0.0 (Log) | 75% | 192.168.89.1 | 80/tcp | |
| Check for Telnet Server | 0.0 (Log) | 80% | 192.168.89.1 | 2000/tcp | |
| Detect Server type and version via Telnet | 0.0 (Log) | 80% | 192.168.89.1 | 2000/tcp | |

(Applied filter: sort-reverse=severity result_hosts_only=1 min_cvss_base= min_qod=70 levels=hmlg autofp=0 notes=1 overrides=1 first=1 rows=100 delta_states=gn) 1 - 14 of 14 (total: 14)

Connect...

Host: 127.0.0.1

Port: 55553

User: msf

Pass: ****

Connect Help

- More detailed information can be gained from individual findings.

Result Details

Task: Immediate scan of IP 192.168.89.1 ID: 64cec0e9-ef7e-4b95-b1d9-3ff113f22676

| Vulnerability | Severity | QoD | Host | Location | Actions |
|---------------|-----------|-----|--------------|----------|---------|
| Services | 0.0 (Log) | 75% | 192.168.89.1 | 23/tcp | |

Summary
This plugin attempts to guess which service is running on the remote ports. For instance, it searches for a web server which could listen on another port than 80 and set the results in the plugins knowledge base.

Vulnerability Detection Result
A telnet server seems to be running on this port

Log Method
Details: [Services \(OID: 1.3.6.1.4.1.25623.1.0.10330\)](#)
Version used: \$Revision: 69 \$

5.1.8 Stopping OpenVAS

To stop the OpenNAS server.

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


Trouble managing data? List, sort, group, tag and search your pentest data in Metasploit Pro -- learn more on <http://rapid7.com/metasploit>

```

      =[ metasploit v4.11.4-2015102801                ]
+ -- --=[ 1498 exploits - 862 auxiliary - 251 post      ]
+ -- --=[ 432 payloads - 37 encoders - 8 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

```

5.3 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:~# service postgresql start
```

If the **Metasploit 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
```

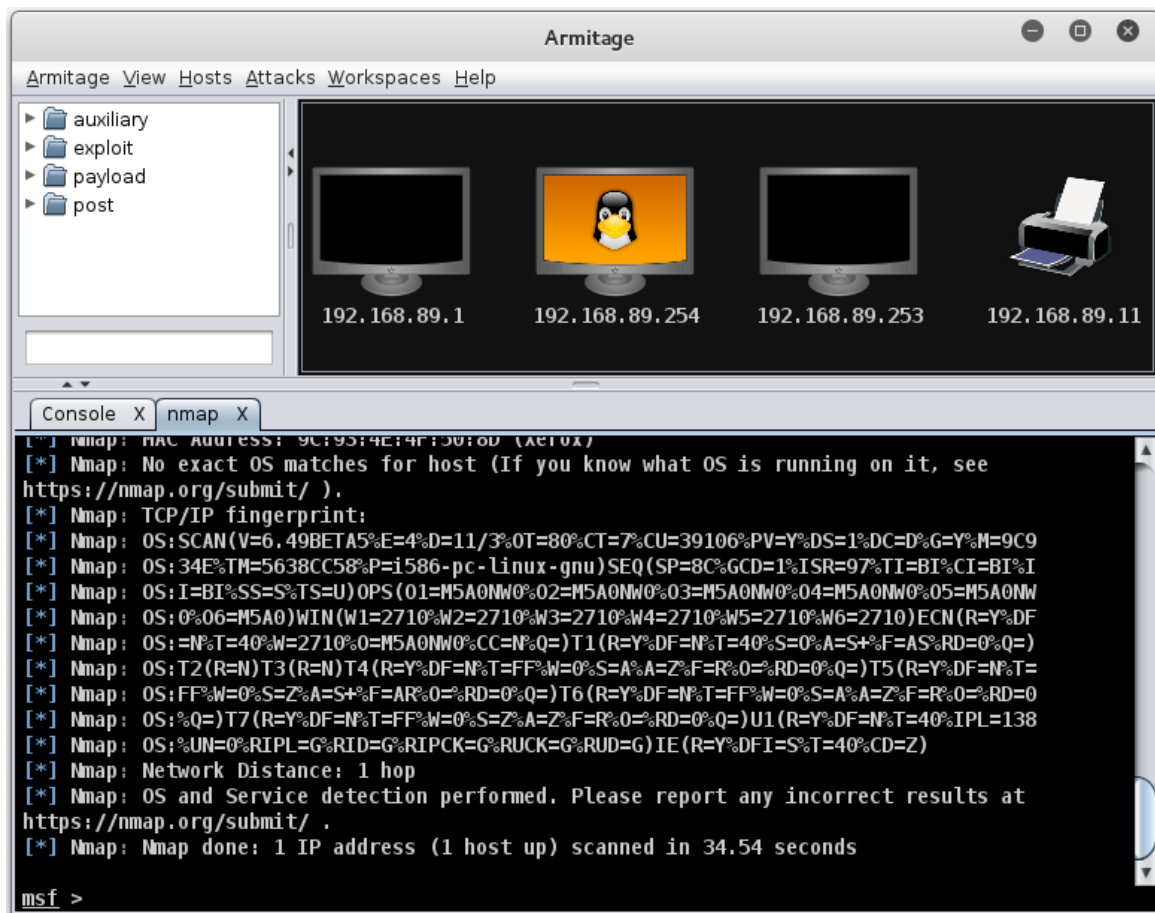


From the menu select:

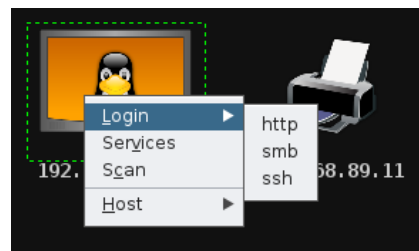
Hosts → nmap Scan → Quick Scan (OS Detect)

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.



5.3.1 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.

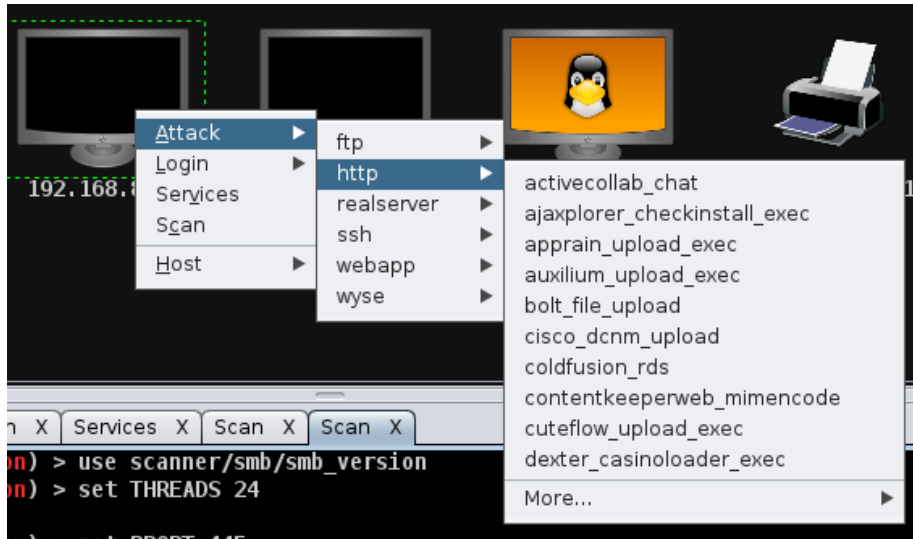
5.3.2 Attack vectors

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

Attacks → Find Attacks

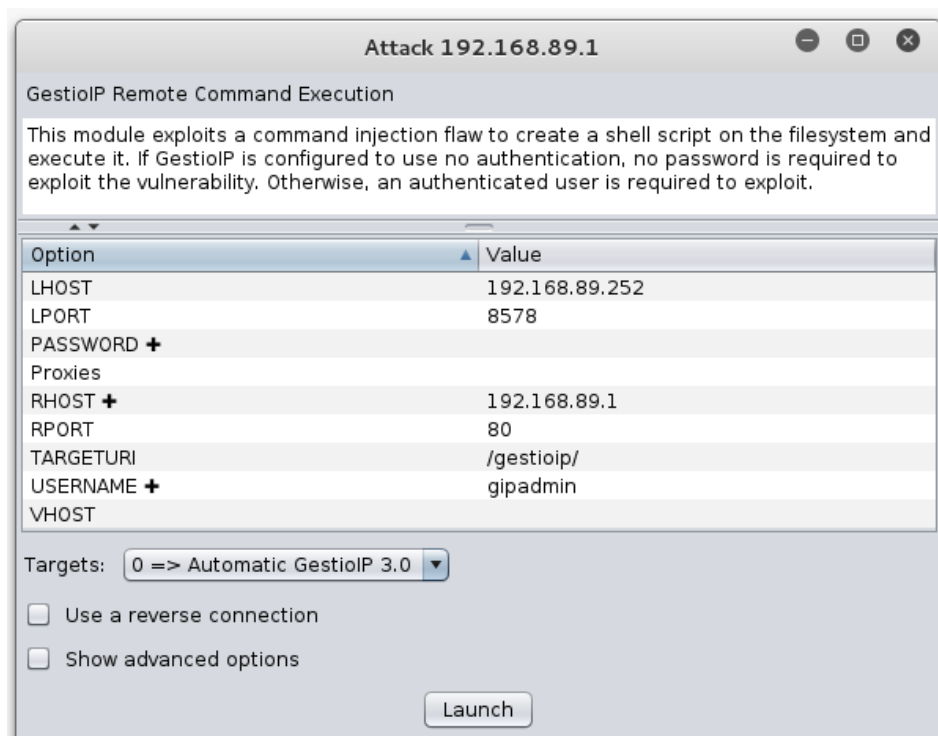
That will query exploits based on the services the scans have discovered.

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



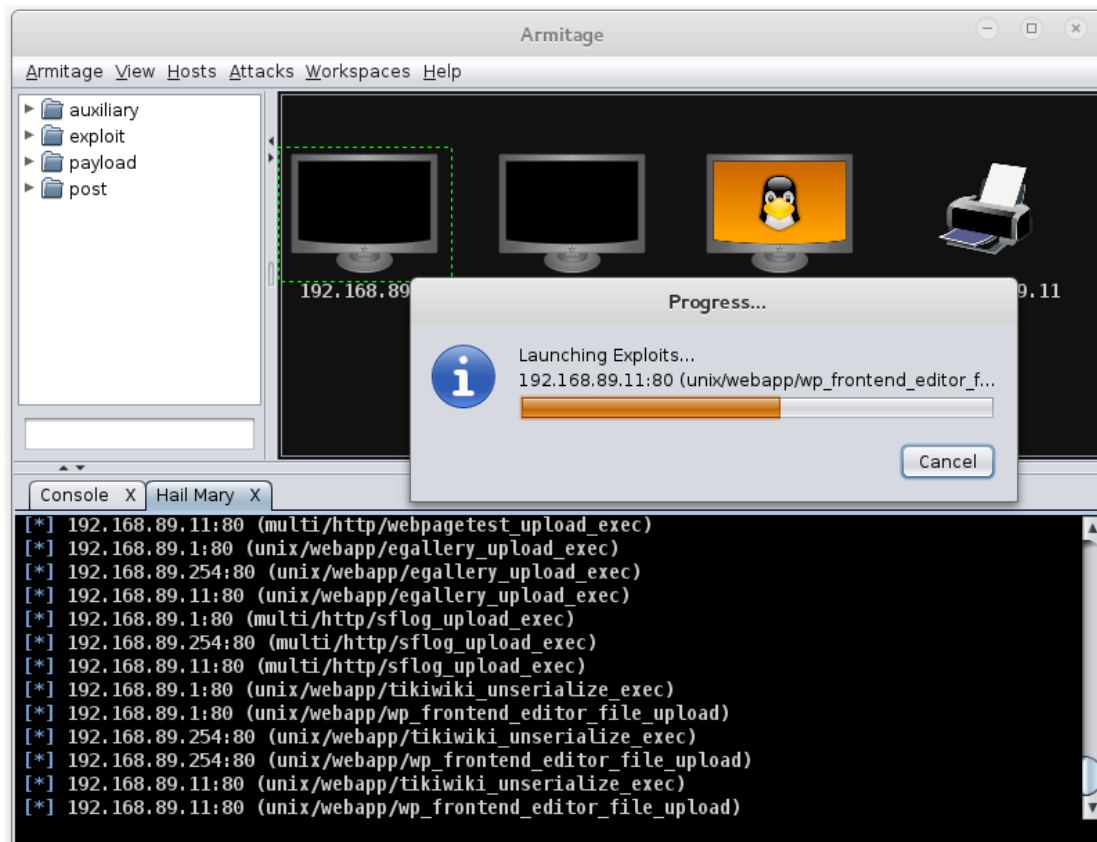
5.3.3 Making the attack

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.



5.3.4 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.



5.3.5 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-ROSSH |
| 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 | |

5.4 Testing Web Servers and Web Applications

5.5 Nikto

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

5.5.1 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'
```

5.5.2 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
```



5.6 Open Web Application Security Project (OWASP)¹

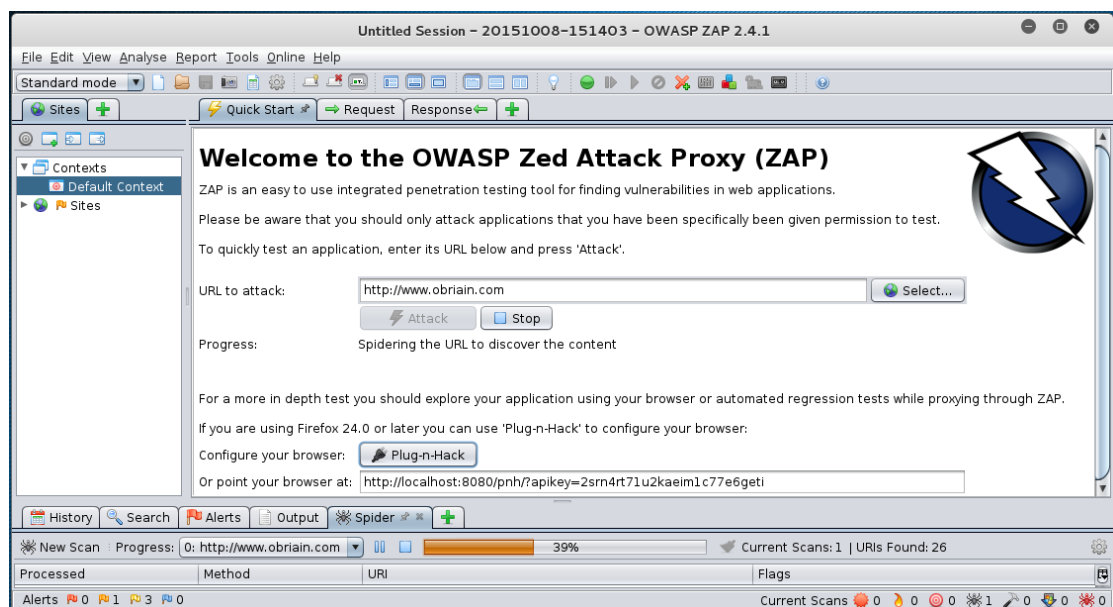
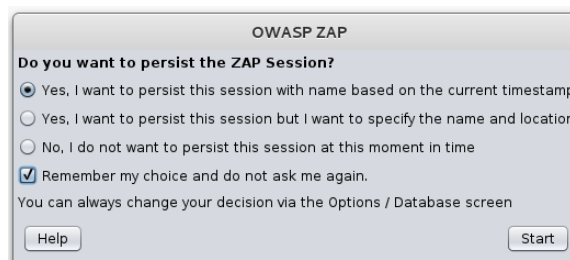
OWASP is an open community dedicated to enabling organisations to conceive, develop, acquire, operate, and maintain applications that can be trusted. All of the OWASP tools, documents, forums, and chapters are free and open to anyone interested in improving application security.

5.7 OWASP Zed Attack Proxy (ZAP)

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

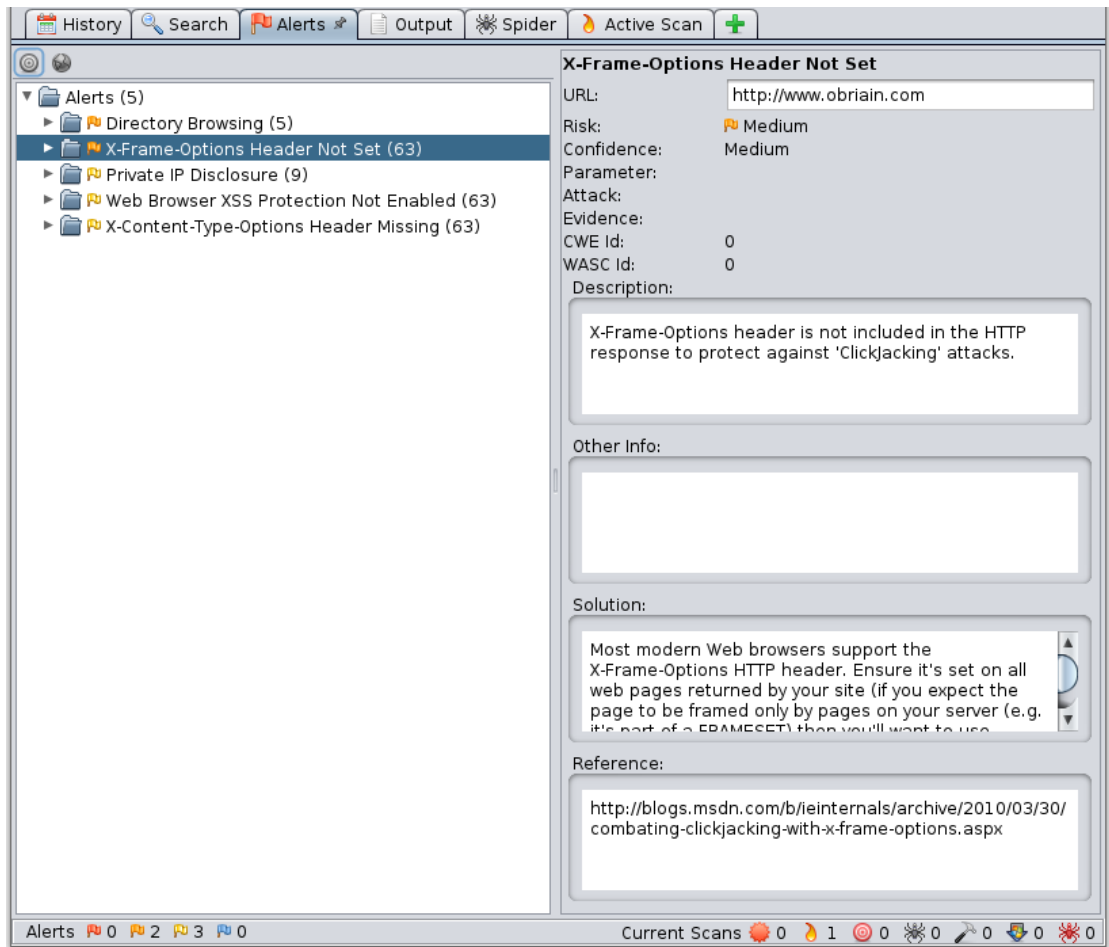
It can be used by developers and function test engineers to carry out penetration testing to identify and close vulnerabilities on their web developments.

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



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.



5.8 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.

The screenshot shows a web browser window titled "ZAP Scanning Report - Iceweasel". The address bar shows "file:///root/Website.html". The browser's bookmark bar includes "Most Visited", "Offensive Security", "Kali Linux", "Kali Docs", "Kali Tools", and "Exploit-DB".

ZAP Scanning Report

Summary of Alerts

| Risk Level | Number of Alerts |
|-------------------------------|------------------|
| High | 0 |
| Medium | 68 |
| Low | 135 |
| Informational | 0 |

Alert Detail

| Medium (Medium) | X-Frame-Options Header Not Set |
|-----------------|---|
| Description | X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks. |
| URL | http://www.obriain.com |
| Solution | Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. ALLOW-FROM allows specific websites to frame the web page in supported web browsers). |
| Reference | http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx |

6. Detection Systems

6.1 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:

```
cedat:~$ 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

```
cedat:~$ 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!

```
cedat:~$ 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**.

```
cedat:~$ 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.

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

6.2 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**.

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

Set the IP Tables logging rules.

```
cedat:~$ sudo iptables -F

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

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

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

Update **psad** signatures.

```
cedat:~$ sudo psad -sig-update
```

```
cedat:~$ 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**.

```
cedat:~$ 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
```

```
cedat:~$ 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
```

6.3 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**.

```
cedat:~$ sudo apt-get install pads

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

cedat:~$ 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
```

7. 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.

8. Lab Exercise

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