Penetration Testing and Intrusion Detection of Firewall Policy Implementation (including SNORBY)

Part A: – Vulnerability Scanning using Zenmap and Nessus[1]

1. Overview

Penetration testing in general and vulnerability scanning in particular is the process of assessing computer systems, networks and applications for vulnerabilities (weaknesses). It is a part of the overall process required to secure a computer system, network or application. This part of the security policy, implementation and testing involves the use of tools such as Nmap – and in particular a GIU version called Zenmap, Nessus, FileZilla and SNORBY which is a detection and analysis tool. These tools provide an insight into penetration testing at the network layer thus providing verification of the accuracy of the policy implementation and associated configuration. In particular they are designed to detect errors, omissions, accidental mis-configuration and to detect vulnerabilities that might otherwise be under the radar.

1.1 Port Scanning

A port in an operating system is specific to an application or a process and serves as a communication end-point. It is associated with the IP address of the host machine as well as the protocol used in the communication. Each address and protocol can be identified by a port together with its port number. The port number is a 16-bit unsigned integer and hence ranges from 0 to 65535. The lower numbered ports are "well know" and commonly used for familiar applications (see following page). Ports above 200 may or may not be well known, they may be proprietary or they may be used at random as and when required by an application. Thus the admin host (204.137.98.145) may be scanned to check for active ports and the services running on them. This information is valuable in providing a final check that the firewall's configuration is ready for service and can be used to assist in preventing attacks on known vulnerabilities of those services.

There are two key stages in port scanning and penetration testing, viz: (i) is a port available upon which a service might operate and (ii) if we attempt to exploit such a port, can we actually run a service on the target machine. For example an FTP port might be available but that does not necessarily mean that an FTP service can be initiated externally on this port. Or – more simply – is there a door there? If so, can I open it?

Port scans can be of several types depending on their use for detection of different protocols being used or for transmitting custom packets. Tools commonly used for port scanning are *nmap* (www.nmap.org), *Zenmap*, (www.nmap.org/zenmap) and *hping* (www.hping.org).

hping is a command-line oriented TCP/IP packet assembler/analyser. The interface supports not only ICMP echo requests but TCP, UDP, and RAW-IP protocols, has a traceroute mode, the ability to send files between a specified ports.

Important components of the Penetration Testing regime include:

- Port scanning (as above)
- Firewall testing (can we penetrate the firewall - and if so - under what circumstances?)
- Network testing, using different protocols and fragmentation

[1] Ensure that the firewall is disabled on both base machine and the Virtual Machine that you are running

- Remote OS fingerprinting – determining what Operating Systems (and versions) are available.
- Fuzzing (separate workshop on this)
- TCP/IP stack analysis and penetration
- Basic network and system tools including:
  tracert, arp, netstat (e.g.netstat –a), nslookup, ifconfig/ipconfig, ping, netcat

Such tools can provide information for other vulnerability scanners such as Nessus as we will see in this part of the laboratory.

Commonly used **ports** can be found at **C:\WINDOWS/system32/drivers/etc/services** and a subset of these are:

echo	7/tcp/udp		
ftp-data	20/tcp		FTP data
ftp	21/tcp		FTP control
ssh	22/tcp		Secure Shell
telnet	23/tcp		
smtp	25/tcp		Simple Mail Transfer Protocol
time	37/tcp/udp		timserver
nameserver	42/tcp/udp		Host Name Server
nicname	43/tcp		whois
domain	53/tcp/udp		Domain Name Server
bootps	67/udp	dhcps	Bootstrap Protocol Server
bootpc	68/udp	dhcpc	Bootstrap Protocol Client
tftp	69/udp		Trivial File Transfer
finger	79/tcp		
http	80/tcp		World Wide Web
kerberos	88/tcp/udp		krb5 kerberos-sec Kerberos
hostname	101/tcp		hostnames NIC Host Name Server
rtelnet	107/tcp		Remote Telnet Service
pop2	109/tcp		Post Office Protocol - Version 2
pop3	110/tcp		Post Office Protocol - Version 3
auth	113/tc		Identification Protocol
ntp	123/udp		Network Time Protocol
epmap	135/tcp/udp		DCE endpoint resolution
imap	143/tcp	imap4	Internet Message Access Protocol
snmp	161/udp		SNMP (Simple Network Management Protocol)
irc	194/tcp		Internet Relay Chat Protocol
ldap	389/tcp		Lightweight Directory Access Protocol
https	443/tcp/udp		MCom
microsoft-ds	445/tcp/udp		
isakmp	500/udp	ike	Internet Key Exchange
exec	512/tcp		Remote Process Execution
login	513/tcp		Remote Login
who	513/udp		whod
cmd	514/tcp		shell
syslog	514/udp		
ldaps	636/tcp	sldap	LDAP over TLS/SSL
doom	666/tcp/udp		Doom Software
kerberos-adm	749/tcp/udp		Kerberos administration
wins	1512/tcp/udp		Microsoft Windows Internet Name Service
l2tp	1701/udp		Layer Two Tunneling Protocol
pptp	1723/tcp		Point-to-point tunnelling protocol
radius	1812/udp		RADIUS authentication protocol

1.2 Network Scanning/Enumeration

Network scanning or enumeration is the method of collecting information about a network, chiefly its topology, number of live hosts, services, protocols being used by the hosts etc. The goal being to start with no information and then to gather as much information as possible about the network and the connected hosts. This information can then be used to identify vulnerabilities in the network and possibly mistakes or omissions made during implementation.

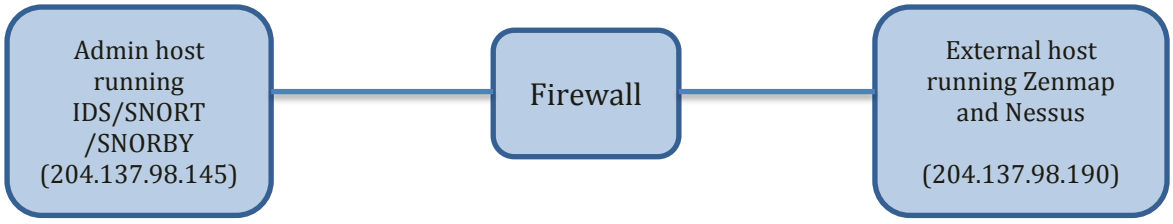
Steps involved in network scanning relevant for this workshop are:

- ping sweeps: used to check if hosts are live or not; commonly done by using ICMP pings.
- Traceroute: used to provide a basic map of the target network architecture
- OS identification: used to predict which services might be running on a system and hence the port scans can be modified accordingly
- port scanning: used to provide a list of open ports and thus potential running applications
- application enumeration: uses banner grabbing to identify the services running on a port

Most commonly used applications identify themselves and hence can be readily recognised. If the active applications are known, then their vulnerabilities can be guarded against.

1.3 Workshop Setup

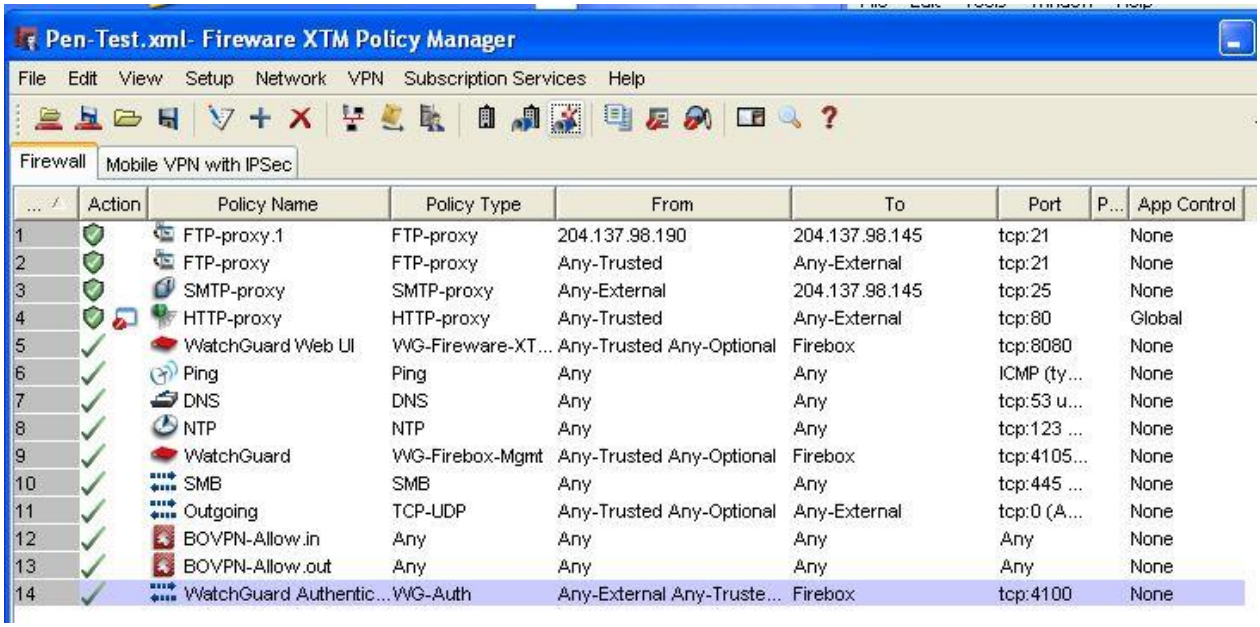
The workshop setup has will continue as before with the admin host (204.137.98.145) to be subject to penetration testing through the firewall, thus verifying the filters and proxies activated and their respective configurations. This host is essentially a bank server which runs a MYSQL database for its customer accounts and thus provides a very practical and realistic study. Thus we commence by scanning for potential vulnerabilities. Often the external host can be referred to as the "attacker" and the admin host as the "victim"[2].



To make the results of this penetration test more interesting we are going to add some additional services as shown in the following diagram. Thus if not already configured, load the file pen-test.xml on to the firewall.

Action	Policy Name	Policy Type	From	To	Port	PBR	Application Control
	FTP-proxy.1	FTP-proxy	204.137.98.190	204.137.98.145	tcp:21		None
	FTP-proxy	FTP-proxy	Any-Trusted	Any-External	tcp:21		None
	SMTP-proxy	SMTP-proxy	Any-External	204.137.98.145	tcp:25		None
	HTTP-proxy	HTTP-proxy	Any-Trusted	Any-External	tcp:80		Global
	WatchGuard Web UI	WG-Fireware-XTI	Any-Trusted Any-Optic	Firebox	tcp:8080		None
	Ping	Ping	Any	Any	ICMP (type: 8, code:		None
	DNS	DNS	Any	Any	tcp:53 udp:53		None
	NTP	NTP	Any	Any	tcp:123 udp:123		None

[2] Check that the bank's server is running. MySQL Server (MYSQL57) on Rays Bank Server (204.137.98.145) from Control Panel > Administrative Tools > Services > Start (if not already running)



...	Action	Policy Name	Policy Type	From	To	Port	P...	App Control
1		FTP-proxy.1	FTP-proxy	204.137.98.190	204.137.98.145	tcp:21		None
2		FTP-proxy	FTP-proxy	Any-Trusted	Any-External	tcp:21		None
3		SMTP-proxy	SMTP-proxy	Any-External	204.137.98.145	tcp:25		None
4		HTTP-proxy	HTTP-proxy	Any-Trusted	Any-External	tcp:80		Global
5		WatchGuard Web UI	WG-Fireware-XT...	Any-Trusted Any-Optional	Firebox	tcp:8080		None
6		Ping	Ping	Any	Any	ICMP (ty...		None
7		DNS	DNS	Any	Any	tcp:53 u...		None
8		NTP	NTP	Any	Any	tcp:123 ...		None
9		WatchGuard	WG-Firebox-Mgmt	Any-Trusted Any-Optional	Firebox	tcp:4105...		None
10		SMB	SMB	Any	Any	tcp:445 ...		None
11		Outgoing	TCP-UDP	Any-Trusted Any-Optional	Any-External	tcp:0 (A...		None
12		BOVPN-Allow.in	Any	Any	Any	Any		None
13		BOVPN-Allow.out	Any	Any	Any	Any		None
14		WatchGuard Authentic...	WG-Auth	Any-External Any-Truste...	Firebox	tcp:4100		None

The applications include the following:

Packet Filters:

- DNS (Domain Name Service) on port 53
- NTP (Network Time Protocol) on port 123
- SMB/CIFS (Server Message Block/Common Internet File System) on ports 137 & 138 (UDP) and 139 (TCP) and 445 (UDP & TCP)

Proxies:

- FTP (File Transfer Protocol) on port 21
- SMTP (Simple Mail Transport Protocol) on port 25

2. *Zenmap*

Nmap has a GUI version called **Zenmap** that is user friendly and presents information more easily read than the command line version. Use the following steps to carry out scans on your administration host in the trusted network (204.137.98.145) using Zenmap:

1. Check the respective IP address and ping to make sure that you have connectivity
2. Start Zenmap from the desktop shortcut (bottom of Windows 10 screen) on the external host (204.137.98.190)
3. Enter the admin host IP address (204.137.98.145) in the target field of Zenmap and select "Quick Scan" from the Profile drop down options.
4. The Quick Scan will show if the target machine is running and whether you have access through the firewall. The MAC address will be displayed and approximately the 100 most commonly used ports will be scanned.
5. Provide a brief summary of the information you have obtained in the box below.

   In general Quick Scan displays –

   Nmap output (starting time, date and location, DNS warning, number of filtered ports including ports open/closed and services they support.

   Ports/Hosts – Displays port number, protocol status, service and version. From this information we can analyse valid and invalid ports

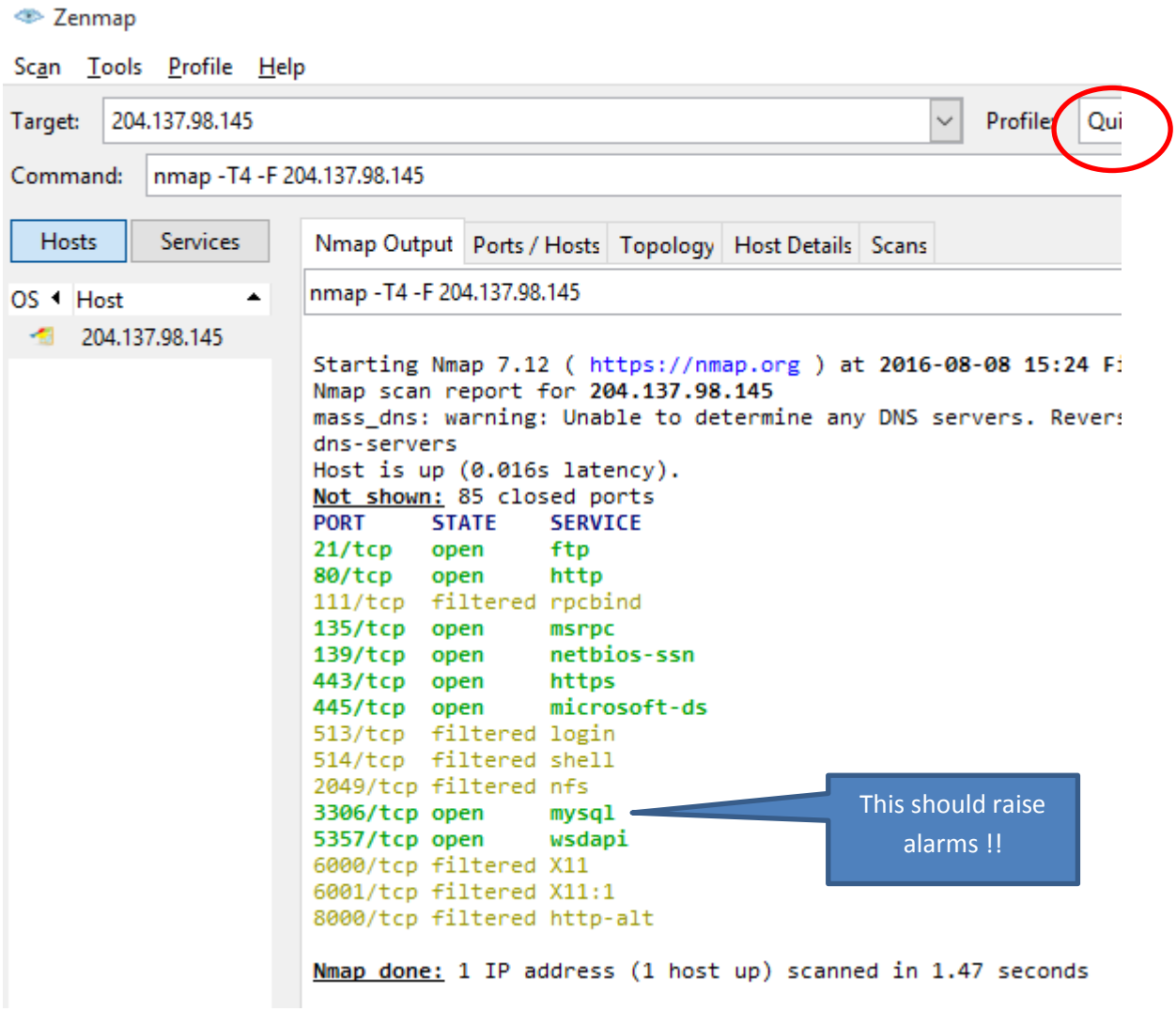
   Topology – displays topology settings and from here we can setup host viewer

   Host details – all IP information demonstrating what details might be exposed to a hacker.

The *Quick Scan* has prompted the following messages /details:

Closed ports from amongst 100 scanned ports) which include:

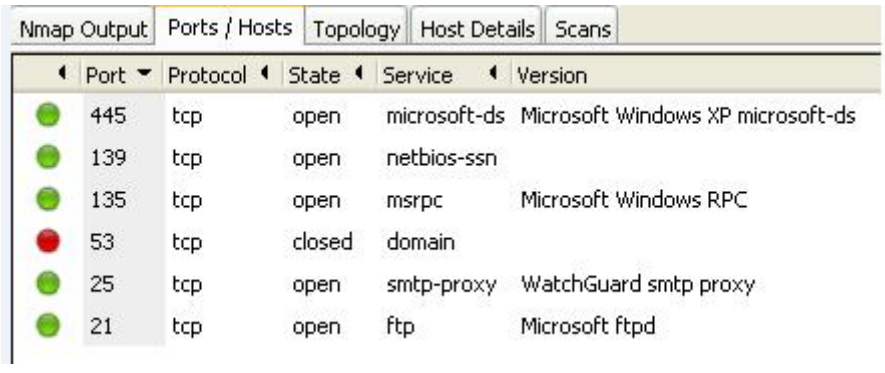
Open ports including– 21 (ftp), 80 (http), 135 (msrpc), 139 (netbios), 443 (https), 445 (Microsoft-ds) and others.



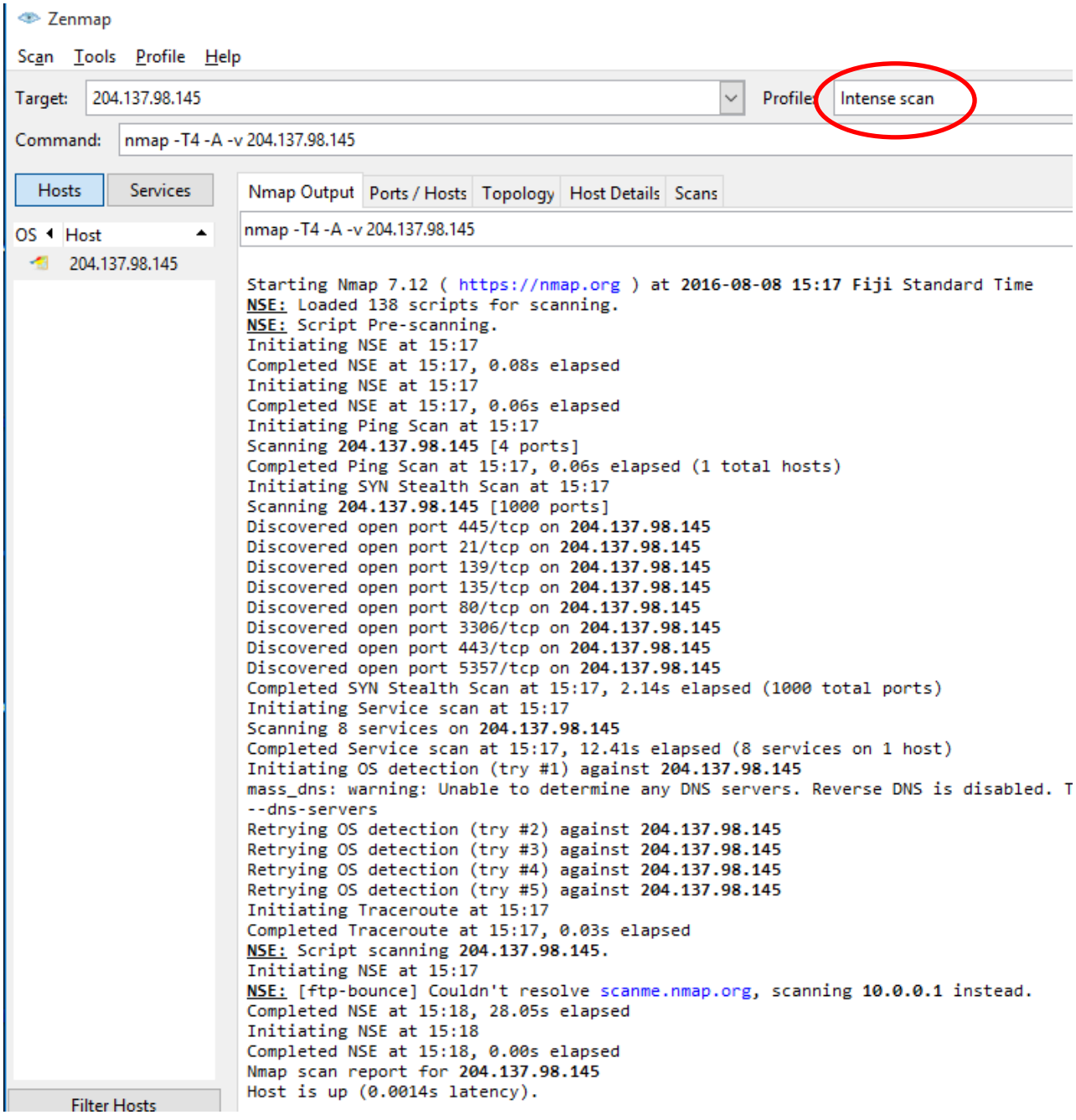
The state of a port can be: *open*, *filtered*, *closed*. *Open* means that an application on the target machine is listening for connections/packets on that port. *Filtered* means that a firewall is likely blocking the port so that Zenmap cannot tell whether it is open or closed.

It is most interesting to note that port 3306 is open as this is the clients' access to the MYSQL database and this will be examined in more detail later.

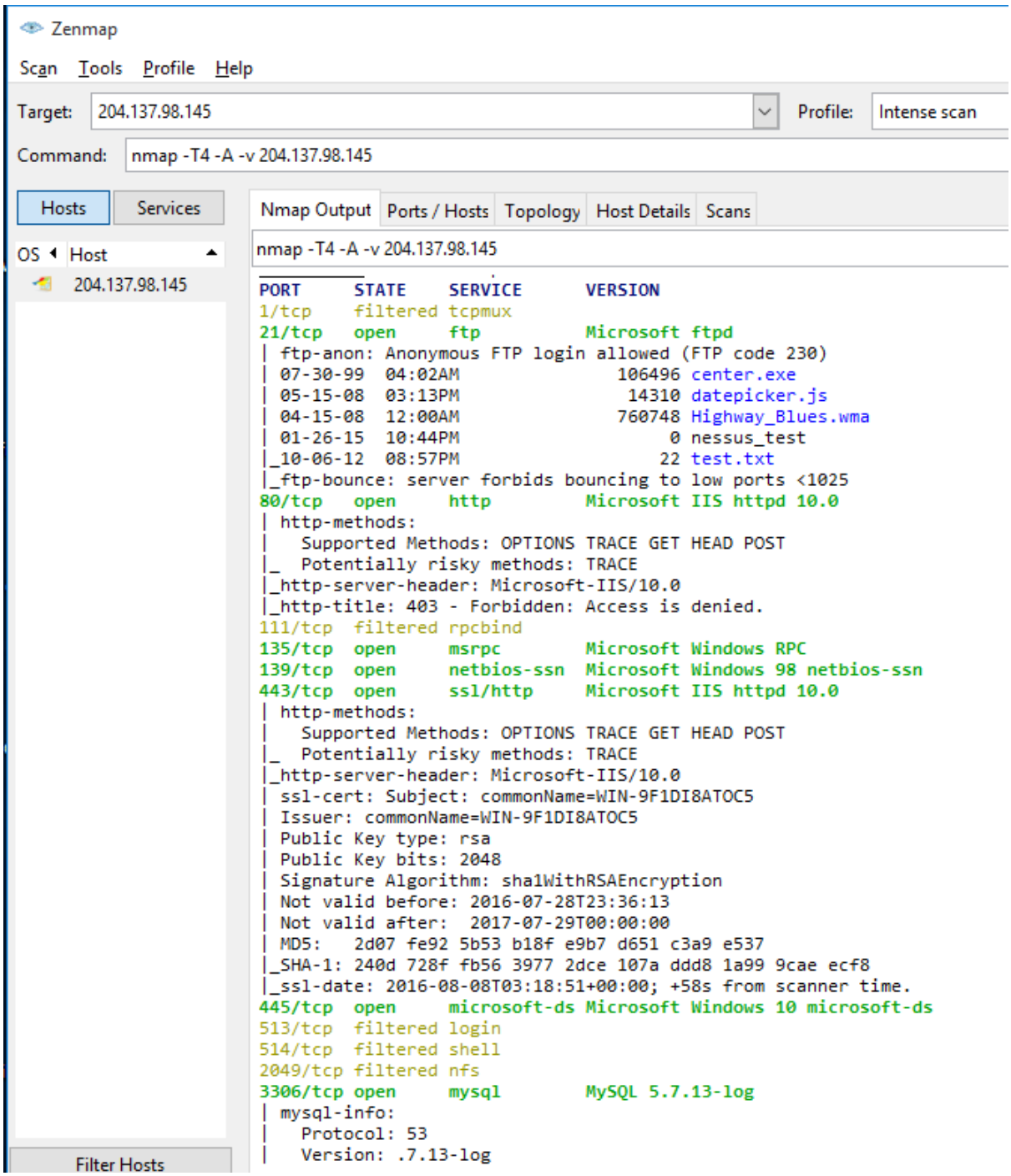
Experiment with the other buttons – Ports/Host, topology etc to see things like:

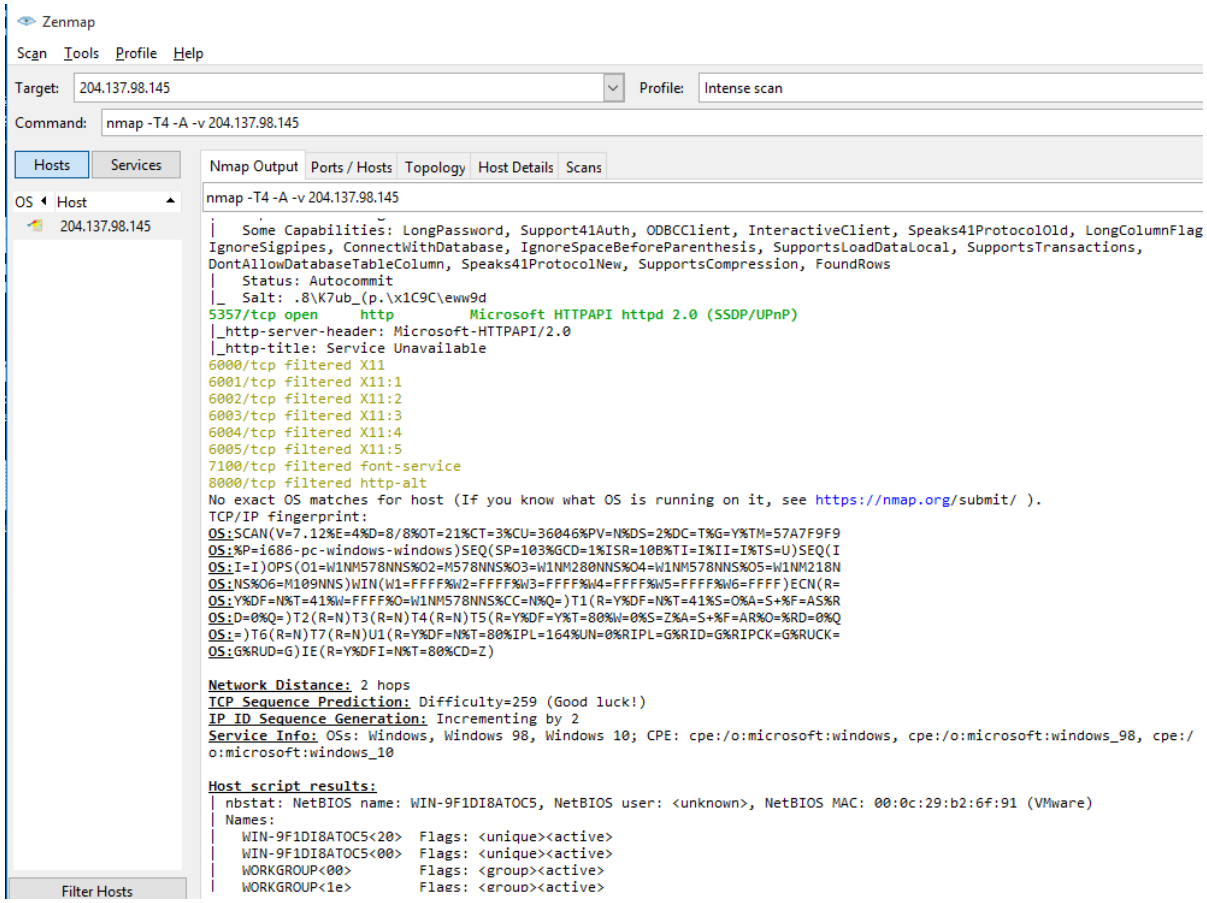


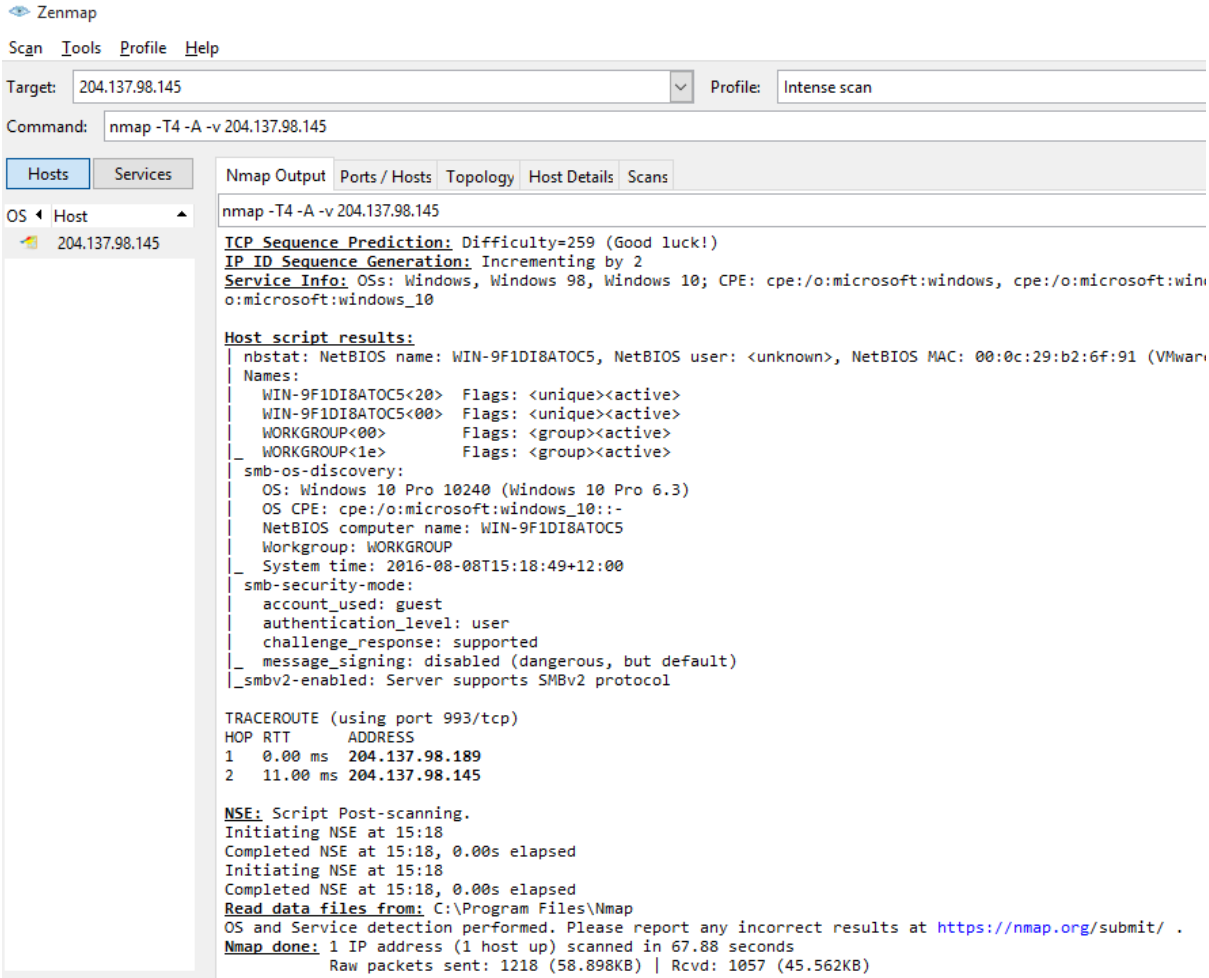
Next use the “Intensive Scan” option from the Profile drop down list and scan the administration host machine. This will take a longer time than Quick Scan (couple more minutes) to complete as more ports are scanned and further information extracted. All the information will be displayed after the scan is over. Navigate to the Ports/Hosts tab or the Host Details (see examples below) tab to see more information. The information gathered can be used to identify the vulnerabilities and take necessary steps to secure the system.



NSE: Nmap Scripting engine


Zenmap
Scan Tools Profile Help
Target: 204.137.98.145
Profile: Intense scan
Command: nmap -T4 -A -v 204.137.98.145
Hosts Services
OS Host
204.137.98.145
Filter Hosts
Nmap Output Ports / Hosts Topology Host Details Scans
nmap -T4 -A -v 204.137.98.145
PORT STATE SERVICE VERSION
1/tcp filtered tcpmux
21/tcp open ftp Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| 07-30-99 04:02AM 106496 center.exe
| 05-15-08 03:13PM 14310 datepicker.js
| 04-15-08 12:00AM 760748 Highway_Blues.wma
| 01-26-15 10:44PM 0 nessus_test
|_10-06-12 08:57PM 22 test.txt
|_ftp-bounce: server forbids bouncing to low ports <1025
80/tcp open http Microsoft IIS httpd 10.0
| http-methods:
| Supported Methods: OPTIONS TRACE GET HEAD POST
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: 403 - Forbidden: Access is denied.
111/tcp filtered rpcbind
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows 98 netbios-ssn
443/tcp open ssl/http Microsoft IIS httpd 10.0
| http-methods:
| Supported Methods: OPTIONS TRACE GET HEAD POST
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
| ssl-cert: Subject: commonName=WIN-9F1DI8ATOC5
| Issuer: commonName=WIN-9F1DI8ATOC5
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2016-07-28T23:36:13
| Not valid after: 2017-07-29T00:00:00
| MD5: 2d07 fe92 5b53 b18f e9b7 d651 c3a9 e537
|_SHA-1: 240d 728f fb56 3977 2dce 107a ddd8 1a99 9cae ecf8
|_ssl-date: 2016-08-08T03:18:51+00:00; +58s from scanner time.
445/tcp open microsoft-ds Microsoft Windows 10 microsoft-ds
513/tcp filtered login
514/tcp filtered shell
2049/tcp filtered nfs
3306/tcp open mysql MySQL 5.7.13-log
| mysql-info:
| Protocol: 53
| Version: .7.13-log


Zenmap
Scan Tools Profile Help
Target: 204.137.98.145
Profile: Intense scan
Command: nmap -T4 -A -v 204.137.98.145
Hosts Services
OS Host
204.137.98.145
Filter Hosts
Nmap Output Ports / Hosts Topology Host Details Scans
nmap -T4 -A -v 204.137.98.145
| Some Capabilities: LongPassword, Support41Auth, ODBCClient, InteractiveClient, Speaks41ProtocolOld, LongColumnFlag
IgnoreSigpipes, ConnectWithDatabase, IgnoreSpaceBeforeParenthesis, SupportsLoadDataLocal, SupportsTransactions,
DontAllowDatabaseTableColumn, Speaks41ProtocolNew, SupportsCompression, FoundRows
| Status: Autocommit
|_ Salt: .8\K7ub_(p.\x1C9C\eww9d
5357/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable
6000/tcp filtered X11
6001/tcp filtered X11:1
6002/tcp filtered X11:2
6003/tcp filtered X11:3
6004/tcp filtered X11:4
6005/tcp filtered X11:5
7100/tcp filtered font-service
8000/tcp filtered http-alt
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/).
TCP/IP fingerprint:
OS:SCAN(V=7.12%E=4%D=8/8%OT=21%CT=3%CU=36046%PV=N%DS=2%DC=T%G=Y%TM=57A7F9F9
OS:%P=i686-pc-windows-windows)SEQ(SP=103%GCD=1%ISR=10B%TI=I%II=I%TS=U)SEQ(I
OS:I=I)OPS(O1=W1NM578NNS%O2=M578NNS%O3=W1NM280NNS%O4=W1NM578NNS%O5=W1NM218N
OS:NS%O6=M109NNS)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FFFF)ECN(R=
OS:Y%DF=N%T=41%W=FFFF%O=W1NM578NNS%CC=N%Q=)T1(R=Y%DF=N%T=41%S=O%A=S+%F=AS%R
OS:D=0%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q
OS:=)T6(R=N)T7(R=N)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=
OS:G%RUD=G)IE(R=Y%DFI=N%T=80%CD=Z)
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: Incrementing by 2
Service Info: OSs: Windows, Windows 98, Windows 10; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_98, cpe:/
o:microsoft:windows_10
Host script results:
| nbstat: NetBIOS name: WIN-9F1DI8ATOC5, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:b2:6f:91 (VMware)
| Names:
| WIN-9F1DI8ATOC5<20> Flags: <unique><active>
| WIN-9F1DI8ATOC5<00> Flags: <unique><active>
| WORKGROUP<00> Flags: <group><active>
| WORKGROUP<1e> Flags: <group><active>


Zenmap
Scan Tools Profile Help
Target: 204.137.98.145
Profile: Intense scan
Command: nmap -T4 -A -v 204.137.98.145
Hosts Services
OS Host
204.137.98.145
Nmap Output Ports / Hosts Topology Host Details Scans
nmap -T4 -A -v 204.137.98.145
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: Incrementing by 2
Service Info: OSs: Windows, Windows 98, Windows 10; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:win
o:microsoft:windows_10
Host script results:
| nbstat: NetBIOS name: WIN-9F1DI8ATOC5, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:b2:6f:91 (VMwar
| Names:
| WIN-9F1DI8ATOC5<20> Flags: <unique><active>
| WIN-9F1DI8ATOC5<00> Flags: <unique><active>
| WORKGROUP<00> Flags: <group><active>
|_ WORKGROUP<1e> Flags: <group><active>
| smb-os-discovery:
| OS: Windows 10 Pro 10240 (Windows 10 Pro 6.3)
| OS CPE: cpe:/o:microsoft:windows_10::-
| NetBIOS computer name: WIN-9F1DI8ATOC5
| Workgroup: WORKGROUP
|_ System time: 2016-08-08T15:18:49+12:00
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_smbv2-enabled: Server supports SMBv2 protocol
TRACEROUTE (using port 993/tcp)
HOP RTT ADDRESS
1 0.00 ms 204.137.98.189
2 11.00 ms 204.137.98.145
NSE: Script Post-scanning.
Initiating NSE at 15:18
Completed NSE at 15:18, 0.00s elapsed
Initiating NSE at 15:18
Completed NSE at 15:18, 0.00s elapsed
Read data files from: C:\Program Files\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 67.88 seconds
Raw packets sent: 1218 (58.898KB) | Rcvd: 1057 (45.562KB)

Another type of scan is the TCP Port Scan as follows:

Zenmap

Scan Tools Profile Help

Target: 204.137.98.145 Profile: Intense scan, all TCP port

Command: nmap -p 1-65535 -T4 -A -v 204.137.98.145

Hosts Services

OS Host

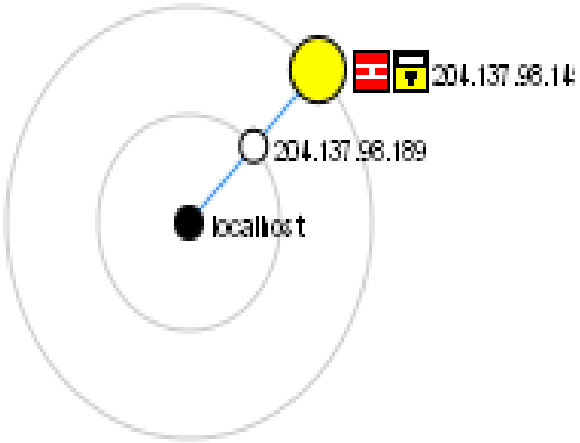
204.137.98.145

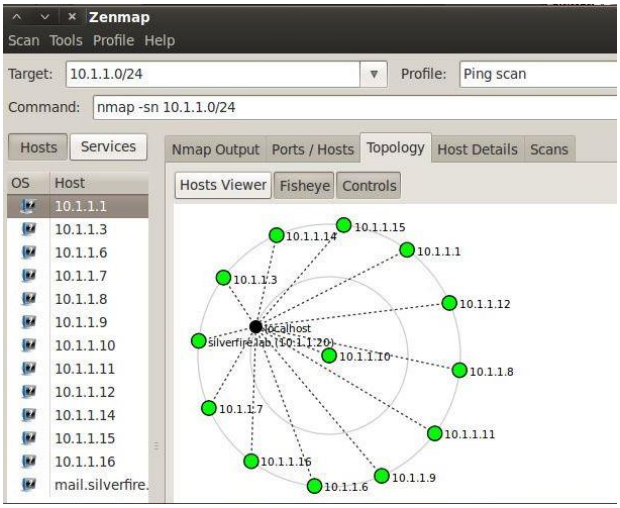
Nmap Output Ports / Hosts Topology Host Details Scans

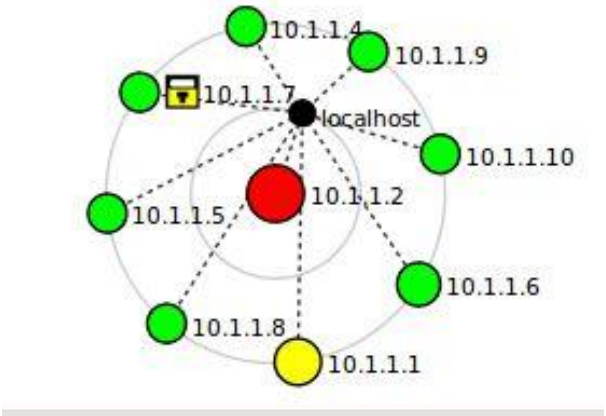
nmap -p 1-65535 -T4 -A -v 204.137.98.145

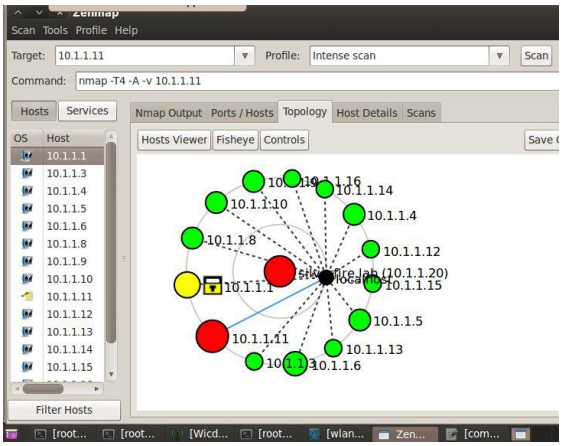
```
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-08 15:25 Fiji Standard Time
NSE: Loaded 138 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 15:25
Completed NSE at 15:25, 0.00s elapsed
Initiating NSE at 15:25
Completed NSE at 15:25, 0.00s elapsed
Initiating Ping Scan at 15:25
Scanning 204.137.98.145 [4 ports]
Completed Ping Scan at 15:25, 0.03s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 15:25
Scanning 204.137.98.145 [65535 ports]
Discovered open port 3306/tcp on 204.137.98.145
Discovered open port 80/tcp on 204.137.98.145
Discovered open port 21/tcp on 204.137.98.145
Discovered open port 139/tcp on 204.137.98.145
Discovered open port 135/tcp on 204.137.98.145
Discovered open port 445/tcp on 204.137.98.145
Discovered open port 443/tcp on 204.137.98.145
Discovered open port 49409/tcp on 204.137.98.145
Discovered open port 49408/tcp on 204.137.98.145
Discovered open port 49410/tcp on 204.137.98.145
Discovered open port 49413/tcp on 204.137.98.145
Discovered open port 49414/tcp on 204.137.98.145
Discovered open port 49411/tcp on 204.137.98.145
SYN Stealth Scan Timing: About 47.94% done; ETC: 15:26 (0:00:34 remaining)
Discovered open port 49412/tcp on 204.137.98.145
Discovered open port 7680/tcp on 204.137.98.145
Discovered open port 5357/tcp on 204.137.98.145
Completed SYN Stealth Scan at 15:26, 62.03s elapsed (65535 total ports)
```

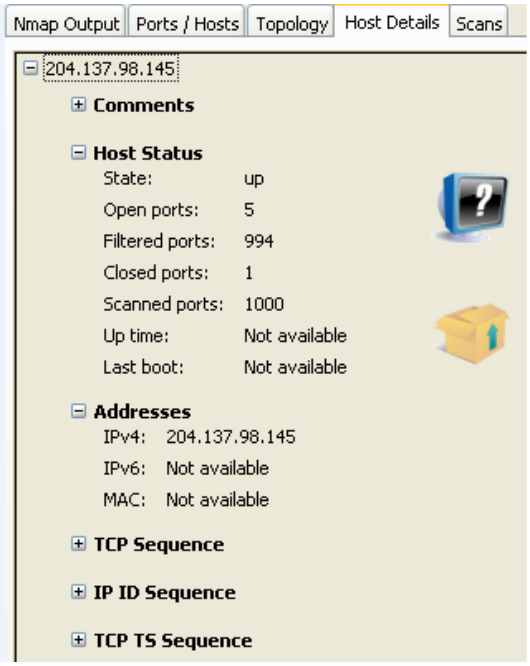
The diagrams below show the Zenmap / Topology for a set of connected devices in a subnet.











Zenmap scans showing interconnection of Backtrack/Kali engines, Androids and SMTP Server for Android Vulnerability Lab

It is necessary to be clear as to what all this scanning information means – for Quick, Intense and TCP Port scans. In particular in the box below is an explanation of what each of these scans shows you which the others do not – accepting of course that there is some degree of overlap.

In particular, your explanation must clearly include the following – saying what these mean and why they may be important (or not) which appear in some of these scans.

- Stealth Scan
- What is NSE?
- FTP Bounce
- TCP/IP Fingerprint
- OS Detection
- Anonymous FTP files
- Implications of ports 135/139 being accessible
- RSA encryption algorithm details for port 443
- MYSQL server information
- TCP Sequence Prediction
- Host Script Results

For the Intense scan (as well as Nessus to follow) firewall autoblock may be an issue here. Set an autoblocking exception for the IP address of 204.137.98.190 or equivalent. However it may be necessary to re-IP the penetration testing machine and make sure that such a new IP address is in the external range (.177 - .188). Check other notes for ping issues.

The Intense Scan shows the following items which the Quick Scan did not:

i) *version* of the open/closed ports being used such as by port 21 (ftp) which uses Microsoft ftpd etc. (as shown on previous page – page 4)

ii) information on TCP sequence numbers, IP ID sequence and TCP TS sequence which can be seen by expanding the entries in the diagram above. TCP Sequence (Index 258; value = drop down), IP ID Sequence (class = Increments 1); TCP TS Sequence (class: non returned (unsupported)). Also shows information about Oss and devices.

iii) host name and workgroup details – for example host script results such as NetBios name, group which can be used to clone machine.

iv) large number of scanned ports, i.e. 1000 ports, compared to Quick Scan with a total of 100 ports only.

v) displays Traceroute information

vi) shows source of read data files

Run Wireshark to show the packets originating from Nmap scans (Quick and Intensive). To select a filter right click on a source address in a trace and add it to the filter. Alternatively select "Apply a filter". We need to obtain a filter of the form source == a.b.c.d or ip.src == a.b.c.d or ip.dst == a.b.c.d Briefly detail your Wireshark results in the box below *comparing* the results obtained between the two scans. Referring to the Wireshark traces, give three examples of protocols and processes being carried out by the Intensive Scan that were not carried out by the Quick Scan.

From Wireshark, the Intensive Scan provides more comprehensive details compared to the Quick Scan which normally only checks to see if a port is open while the Intensive Scan attempts to access the port to obtain details about service and machine. The Quick Scan showed port availability, but the detailed scan was much more through. To gain information on FTP, NetBIOS, configurations, the Intensive Scan also queried the victim using the query services (read-only) of the FTP and NetBIOS protocols.

Examples of protocol details in the Intensive Scan include:

i) NBNS – query of NetBios name of victim's machine (Intensive only)

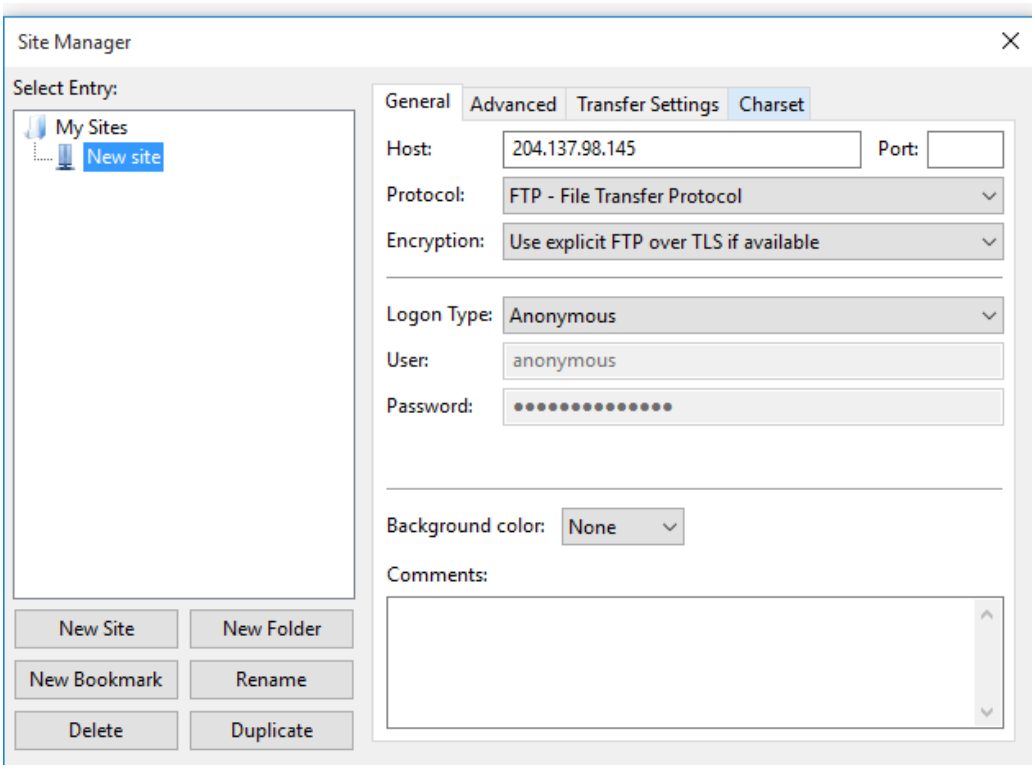
ii) SMB – session setup and response to initiate remote login (Quick & Intensive but more detail, e.g. logon attempt)

iii) FTP – scanned in both Quick and Intensive but tested in Intensive

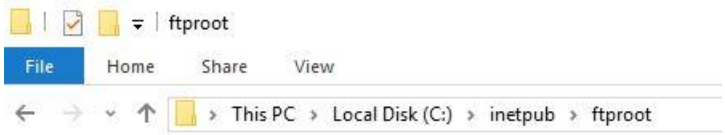
iv) Also possible are UDP / TCP / ICMP

3. *Filezilla*

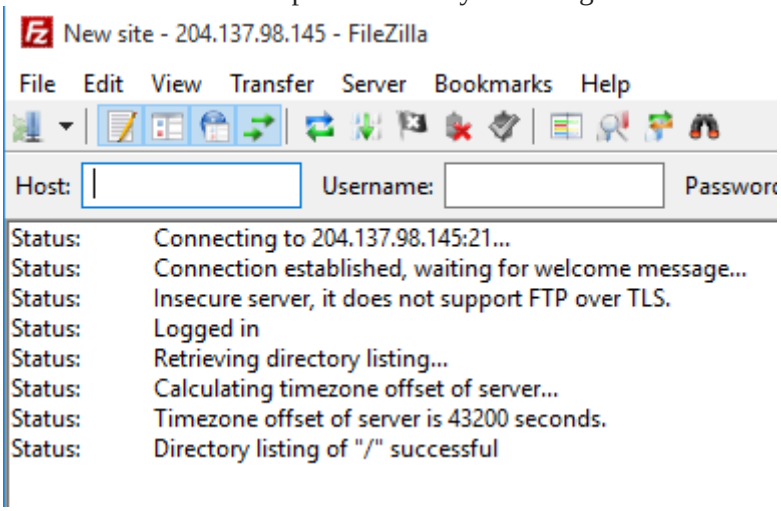
FileZilla is a cross-platform FTP application which can be used for Penetration Testing. It can test for anonymous access as well as the uploading/downloading of files. FTP is a very common service founds on almost every server and the risks resulting from this need to be careful tested.

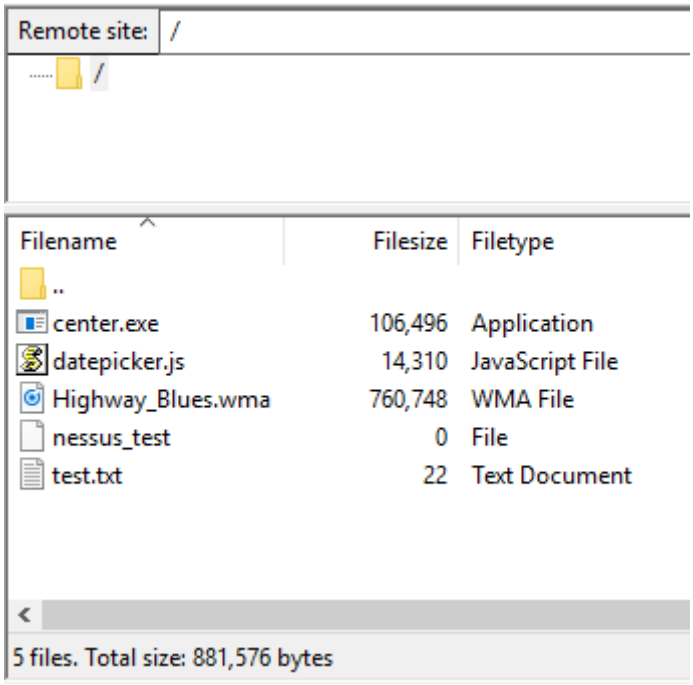


From the external Penetration testing machine open Filezilla in the bottom row on Windows 10 and select File > Site Manager > New site and enter the necessary parameters as in the diagram above (204.137.98.145, FTP and anonymous/guest) and connect. If the FTP service is exposed then it may be possible to access ftproot via Port 21 as follows:



Now see if you can detect the files in the ftp root directory as in diagrams below:



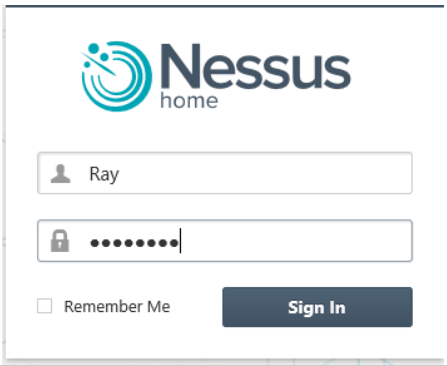


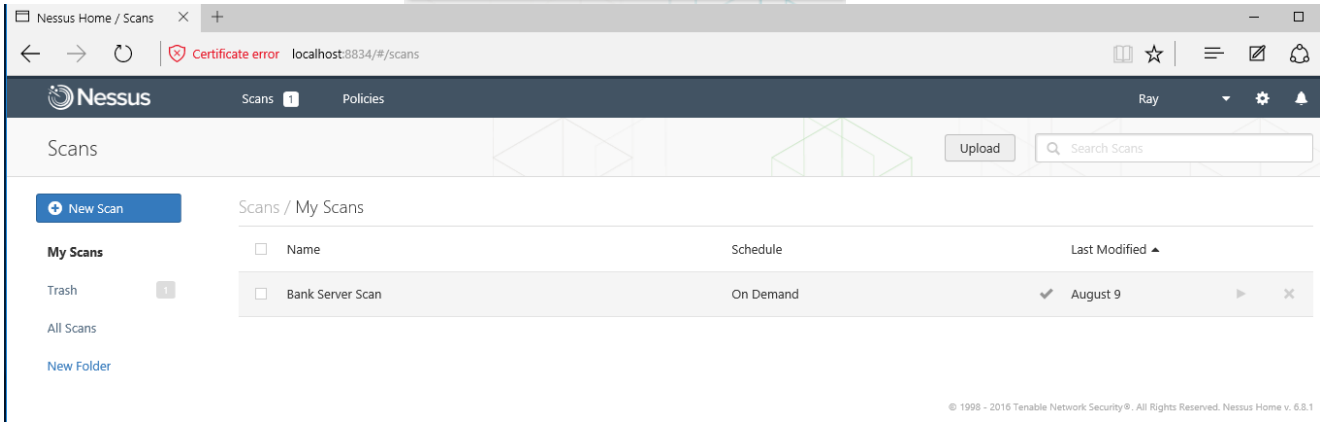
4. *Nessus*

Login: Ray / P#ssw0rd

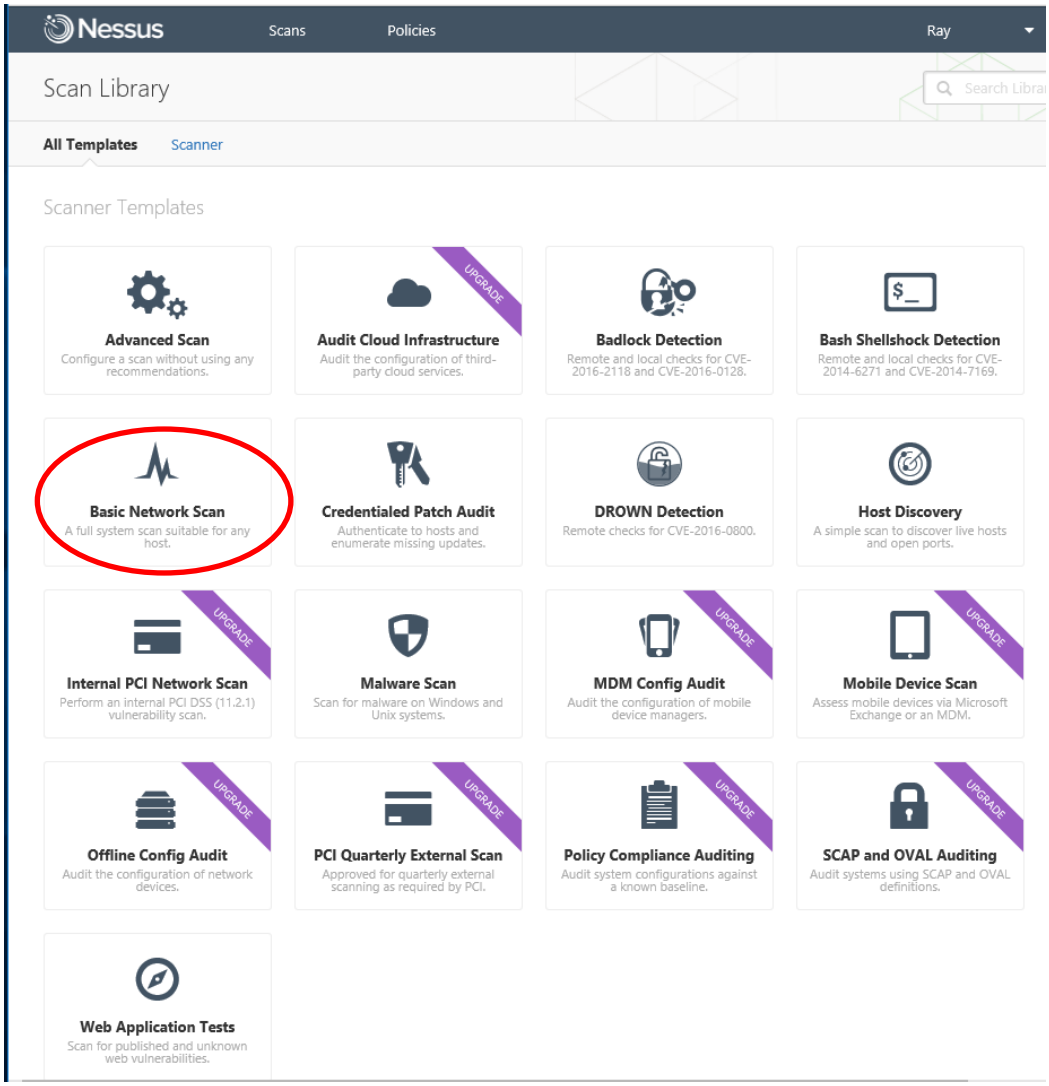
Having gained initial experience with Nmap (Zenmap) we will now use Nessus which is a valuable and widely used commercial comprehensive vulnerability scanner. It can be used for performing not just port scans but detailed vulnerability assessments using custom policies. The following steps show how to use Nessus:

On the external host machine (204.137.98.190) click on the desktop Nessus Web Client (If Microsoft Edge does not work try Chrome and enter https://localhost:8834) and login in with Ray / P#ssw0rd. You will see an earlier scan entitled "Bank Server Scan" which can be referred to later.





Select *New Scan* and then choose *Basic Network Scan* from the Scan library.



Consider these scan options which are available (above) and which may be important to your industry or commercial operation.

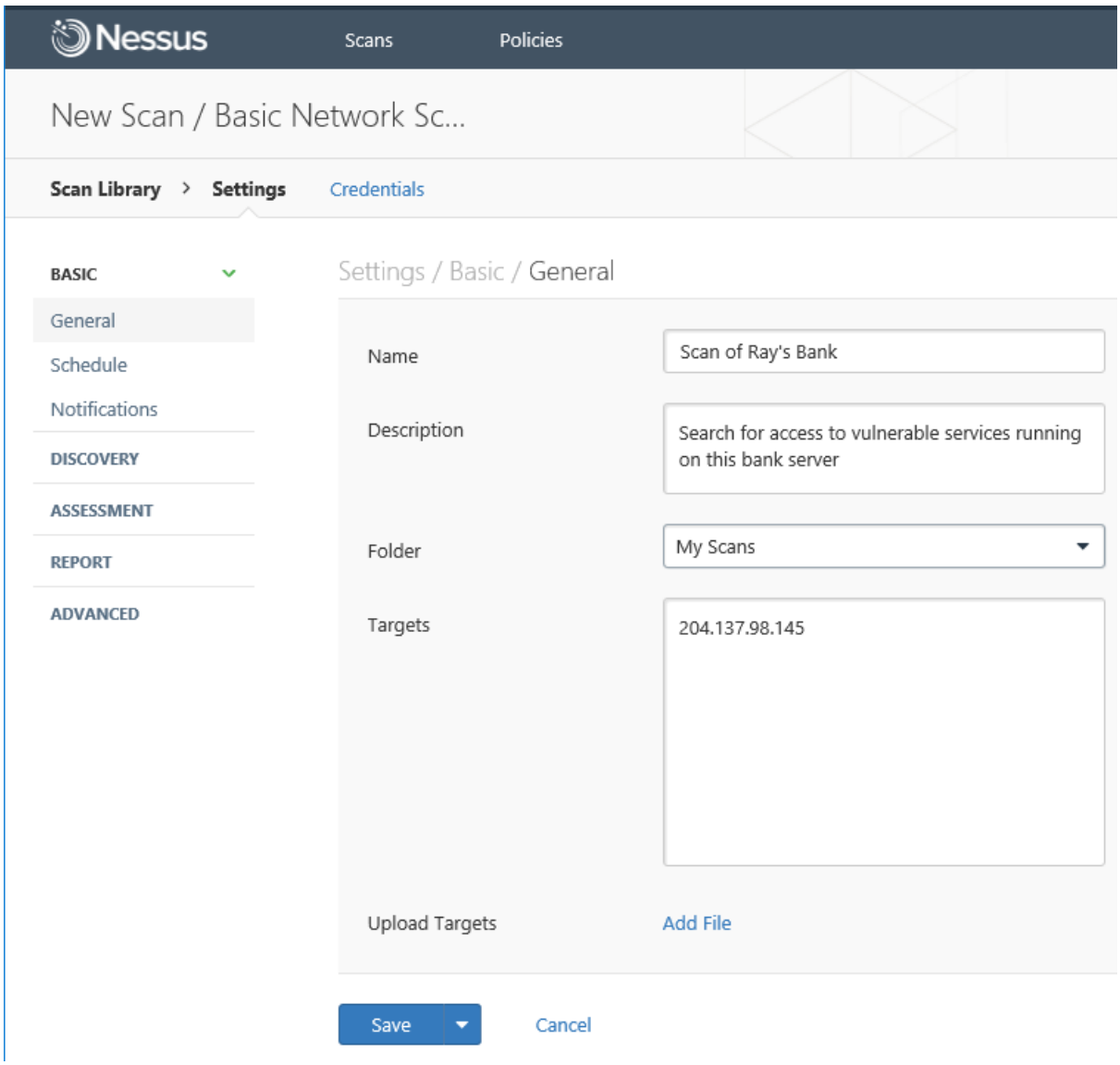
A number of the CVEs (Common Vulnerabilities and Exposures) can be found at www.cve.mitre.org – for example:

CVE-2016-2118 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-2118)

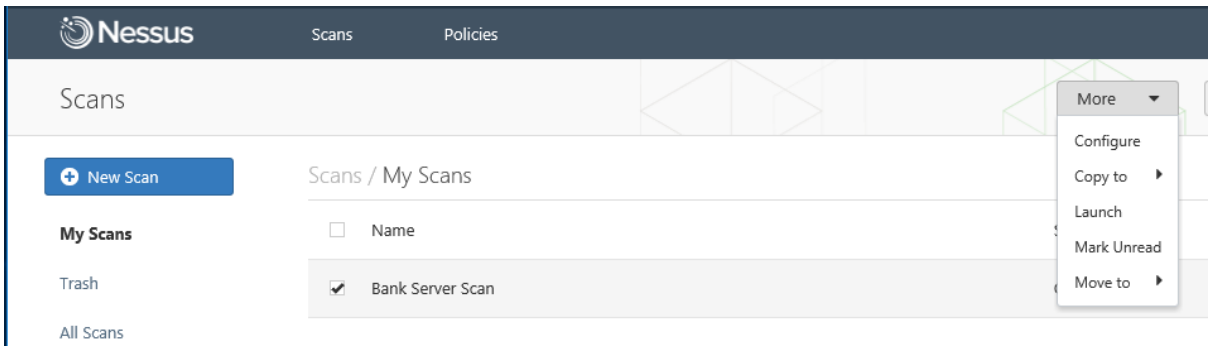
CVE-2016-0128 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-0128)

CVE-2016-0800 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-0800)

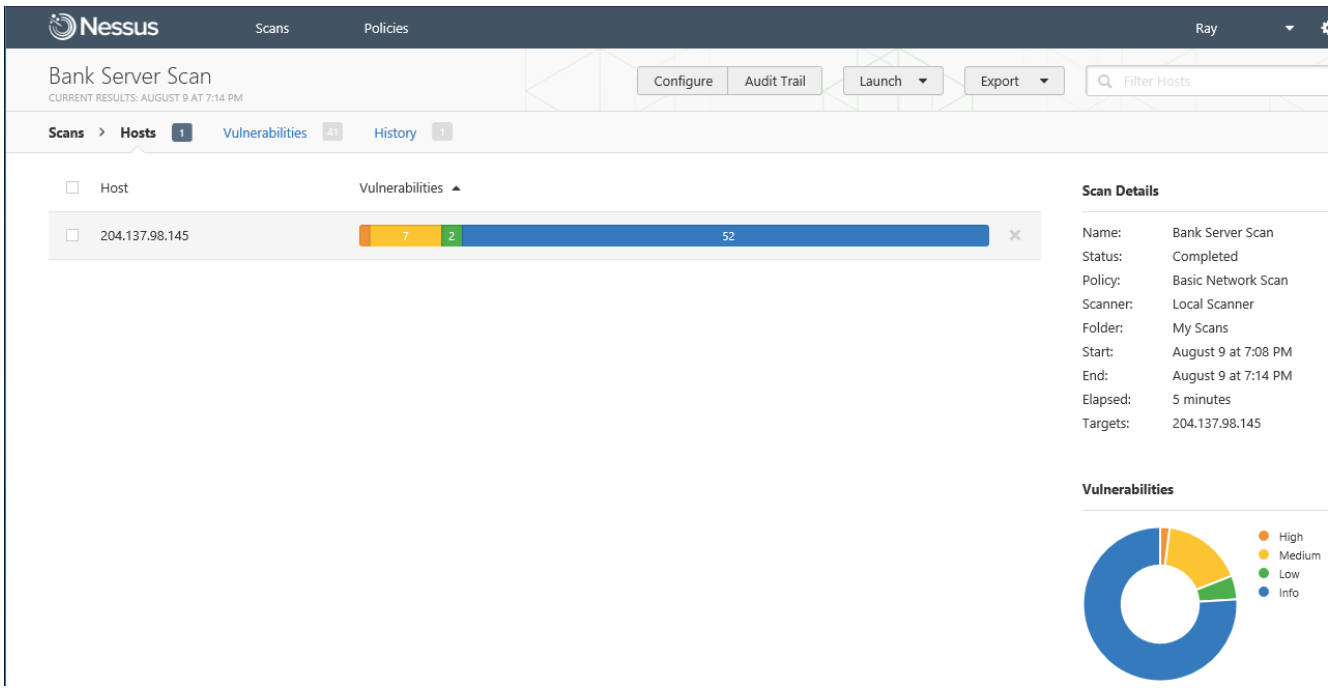
Now provide the credentials for your scan of Ray's Bank server and save:



Select the Scan then > More > Launch as in diagram below



When the scan is complete (typically takes 10-15 minutes) examine the vulnerability window and explain what you see.



Click on the colour bar showing the number of severity at each of the levels and explain what you see. Rather than explain each potential vulnerability in turn, they are grouped to provide an explanation of the relevance and importance for each of the vulnerabilities in the group.

For each category of severity – High, Medium, Low, Information – consider the meaning and significance of a vulnerability and in what way it does/does not affect your banks security.

Now go back to your Zenmap scan and compare with this Nessus scan. Are there issues raised in one which have not been discovered in the other? You may have to examine the Nessus scans in more detail to identify port numbers which are explicitly stated in the Zenmap scan but are more buried within each vulnerability description in Nessus.

Feel free to *export* your scan so that you can analyse it later.

Bank Server Scan

CURRENT RESULTS: AUGUST 9 AT 7:14 PM

Configure | Audit Trail | Launch | Export

Hosts > 204.137.98.145 > Vulnerabilities 41

Severity	Plugin Name	Plugin Family	Count
HIGH	FTP Privileged Port Bounce Scan	FTP	1
MEDIUM	Anonymous FTP Enabled	FTP	1
MEDIUM	SMB Signing Disabled	Misc.	1
MEDIUM	SSL Certificate Cannot Be Trusted	General	1
MEDIUM	SSL Certificate Signed Using Weak Hashing Algorithm	General	1
MEDIUM	SSL Self-Signed Certificate	General	1
MEDIUM	SSL Version 2 and 3 Protocol Detection	Service detection	1
MEDIUM	SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)	General	1
LOW	FTP Supports Cleartext Authentication	FTP	1
LOW	SSL RC4 Cipher Suites Supported (Bar Mitzvah)	General	1
INFO	DCE Services Enumeration	Windows	9
INFO	Nessus SYN scanner	Port scanners	7
INFO	Service Detection	Service detection	4
INFO	HTTP Methods Allowed (per directory)	Web Servers	2

Nessus | Scans | Policies

Bank Server Scan

CURRENT RESULTS: AUGUST 9 AT 7:14 PM

Configure

Hosts > 204.137.98.145 > Vulnerabilities 41

MEDIUM SSL Self-Signed Certificate

Description

The X.509 certificate chain for this service is not signed by a recognized certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone could establish a man-in-the-middle attack against the remote host.

Note that this plugin does not check for certificate chains that end in a certificate that is not self-signed, but is signed by an unrecognized certificate authority.

Solution

Purchase or generate a proper certificate for this service.

Output

```
The following certificate was found at the top of the certificate
chain sent by the remote host, but is self-signed and was not
found in the list of known certificate authorities :

|-Subject : CN=WIN-9F1DI8ATOC5
```

Port	Hosts
443 / tcp / www	204.137.98.145

Both Zenmap and Nessus identify significant vulnerabilities associated with ports upon which SMB/CIFS operate (135, 137, 139, 445) as detailed below.

SMB/CIFS (Server Message Block/Common Internet File System) on ports 137 & 138 (UDP) and 139 (TCP) and 445 (UDP & TCP) all allow file and print sharing – frequently with no log in requirements. While convenient – this is security vulnerability?

The significant vulnerabilities are:

a. A printer sharing service using CIFS provides shared access to the port which an attacker can exploit.

b. Retrieving operating system details of the host remotely.

c. Remotely logging into a host running a CIFS server without authentication (under the correct conditions).

d. Remote registry access is disabled using *winrg* meaning the Nessus tool cannot scan the registry (in penetration testing situations).

e. It is possible to gain a list of nearby Windows hosts.

Other possibilities include:

Remote host can be accessed by a NULL session

OS version, LAN manager, host name can all be retrieved from remote machine

Browse list can be obtained from remote host, i.e. list of shared files and directions are available.

Notes: You may find a few unfamiliar acronyms in the scan report:
CVSS – Common Vulnerability Scoring System
CVE – Common Vulnerabilities and Exposures
BID – Bugtraq ID
OSVDB – Open Source Vulnerability Database

Part B: – Intrusion Detection System using SNORT/SNORBY

1 Overview

Most attacks on computers were determined by manually checking the audit logs for unusual or malicious behaviour. However, with the increase in sophistication and volume of data generated, real-time analysis of these logs was needed. Intrusion detection systems were created to analyse audit logs as they were generated. As we covered in the accompanying lecturers, based on the method used for detection, the approaches can be either misuse detection or anomaly detection.

- Misuse detectors look for sets of events that match a predefined pattern of a known attack. These patterns are also called signatures and so misuse detectors are also called signature-based detectors. Misuse detectors generate a low rate of false-positives. However, they need to be constantly upgraded with the signatures of the attacks that need to be detected.
- Anomaly detectors operate on the assumption that attacks are significantly different from legitimate activity and by detecting these differences the attacks can be identified. They construct profiles representing legitimate activity of users, hosts and network devices over a period of time. Currently threshold detection and statistical measures are widely used in the anomaly detection. Other measures like neural networks, genetic algorithms etc are less frequently used. However, anomaly detectors produce a large number for false positives and also require a large number of legitimate activities to be trained to recognise valid actions.

Based on their information source intrusion detection systems (IDS) can be classified into:

- network-based IDSs
- host-based IDSs
- hybrid IDSs
- application based IDSs (subset of host-based IDSs)

Most commercial IDSs are network-based. They detect attacks by capturing and analysing network packets. They have the advantage of covering a large network with a few well-placed IDSs. They are usually passive devices and can be made invisible to attackers and hence are very secure against attacks. However, network-based IDSs do not always perform well under high network loads.

Host-based IDSs are installed on an individual computer system and collect information from that system with greater precision and reliability. They typically use system logs and operating system audit trails to check for intrusions. They can view attacks that a network based IDS cannot. However, host-based IDS are harder to manage and have a processing cost on the system that they monitor.

Application-based IDSs are actually a subset of Host-based IDSs that monitor a software application by analysing its transaction log files. Application-based IDSs can directly trace unauthorised activity to the responsible users. However, they are also more vulnerable to false analysis as the application logs are not well-protected. Also, they cannot detect and guard against Trojan horse malware. They are always used in combination with Host-based and Network-based IDSs.

2. SNORT

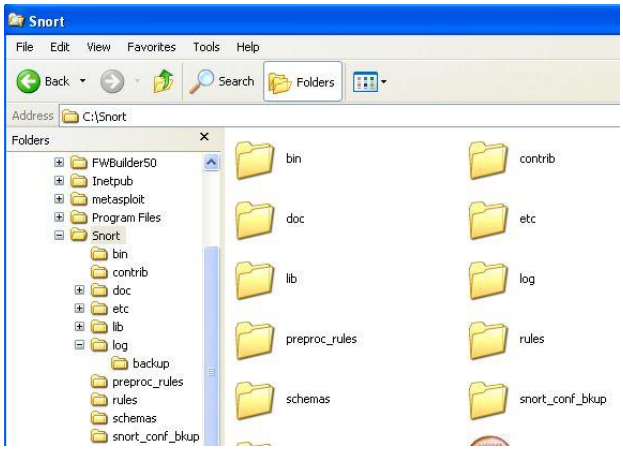
Snort is an open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS), with the ability to perform real-time traffic analysis and packet logging. Snort performs protocol analysis, content searching, and content matching. The program can also be used to detect probes and attacks, including operating system fingerprinting attempts, common gateway interface attacks, buffer overflows, server message block probes, and stealth port scans.

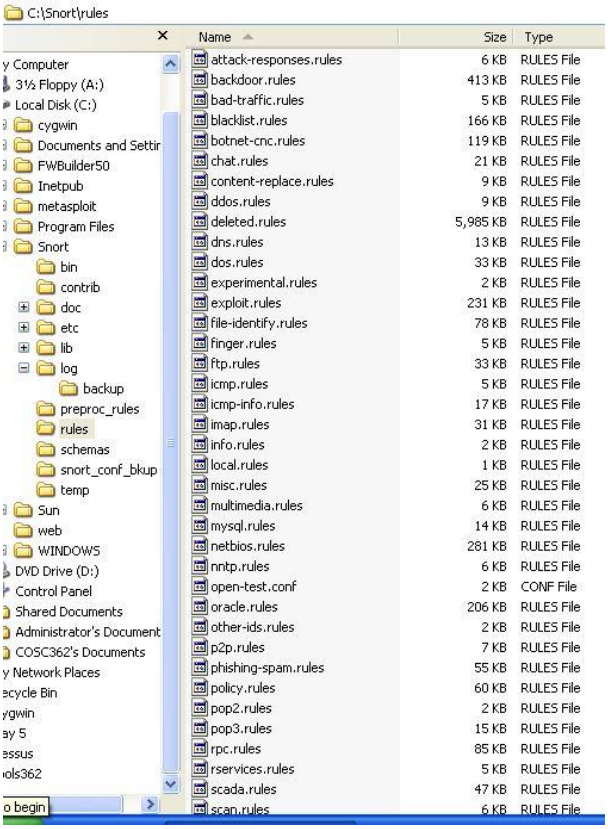
Snort can be configured in three main modes: - sniffer, packet logger, and network intrusion detection but we will use it in intrusion detection mode in this workshop.

- In *sniffer mode*, the program will read network packets and display them on the console.
- In *packet logger mode*, the program will log packets to the disk.
- In *intrusion detection mode*, the program will monitor network traffic and analyse it against a rule set defined by the user. The program will then perform a specific action based on what has been identified.

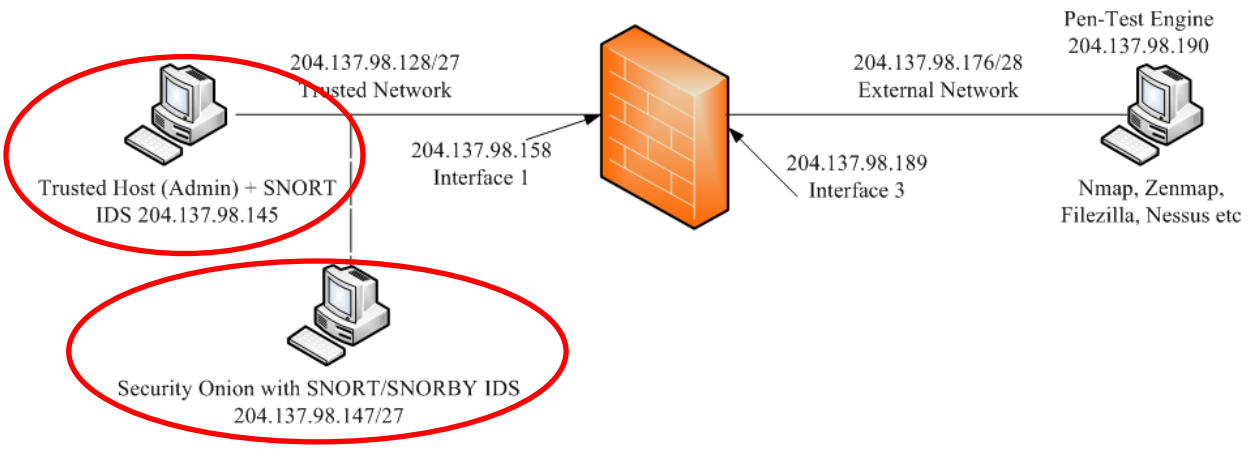
There are two ways that we can use SNORT in this workshop:

1. Installation on current server. This is a 100 MB IDS engine and only produces text-based (non-graphical) output. This installation includes a detailed set of schemes and rules which need to be regularly updated to remain current and a sample is shown below.





2. SNORBY which is a graphical version of SNORT and a commercial tool widely used in practice. It is run on a separate machine in the same network as the server being monitored by the IDS process and monitors the server's interfaces for intrusions. It is a 5GB Graphical engine compared with the SNORT Installation in 1 above which is only 100MB in comparison.



This lab will involve running both of these IDS options for experience and comparison. In both cases the server under investigation is Ray's Bank (204.137.98.145). In each case we will analyse the basic concepts of intrusion detection systems – involving penetration testing and analysis at both the network and application level. SNORT is a very useful tool as it combines both network vulnerability analysis which then can then lead to vulnerabilities in the upper (application) layer.

3. Configuring SNORT on the Ray's Bank on the trusted network

Use the following steps to set up SNORT on Ray's Bank (204.137.98.145):

1. From the previous penetration testing laboratories, both 204.137.98.190 and 204.137.98.145 should be able to ping each other thus confirming connectivity.
2. For background information the following has been done but is included as background. On the bank machine ensure that the snort.conf file has the correct IP Address (trusted network). C: > SNORT > etc. Navigate to the snort.conf file in the snort folder and about line 45 check that the IP Address is the same as your current machine upon which SNORT is running. If it is not, right click and edit with Notepad++.

   (Note also for editing ipvar HOME_NET 204.137.98.128/27 and ipvar EXTERNAL_NET any)
3. We will now set up SNORT to run as an IDS on this bank server. Click on the Command Prompt icon on the desktop to start up a terminal. Next navigate to C:\snort\bin using the commands:

   ```
   > cd\
   > cd C:\snort\bin
   ```
4. Now enter the following command but enter (do not copy and paste this command as hidden characters can cause errors) carefully into the console ***but do not press enter yet***:

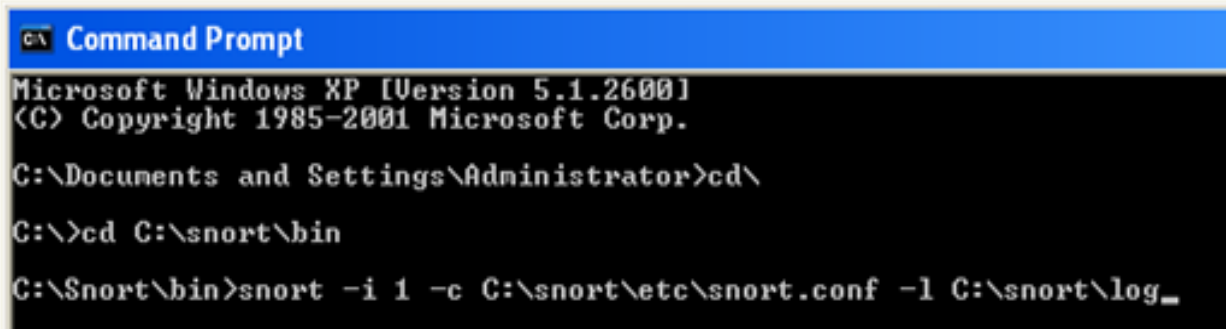
   ```
   > snort –i 1 –c C:\snort\etc\snort.conf –l C:\snort\log
   ```

   -i 1 specifies interface number **1**

   -c C:\snort specifies the configuration file

   -l C:\snort specifies the *log* directory (note that it is **l** and not 1 !!)

   Note: be very clear where we are referring to a "1" one interface and "l" for log fie location.



Snort startup Command on bank server in trusted network)

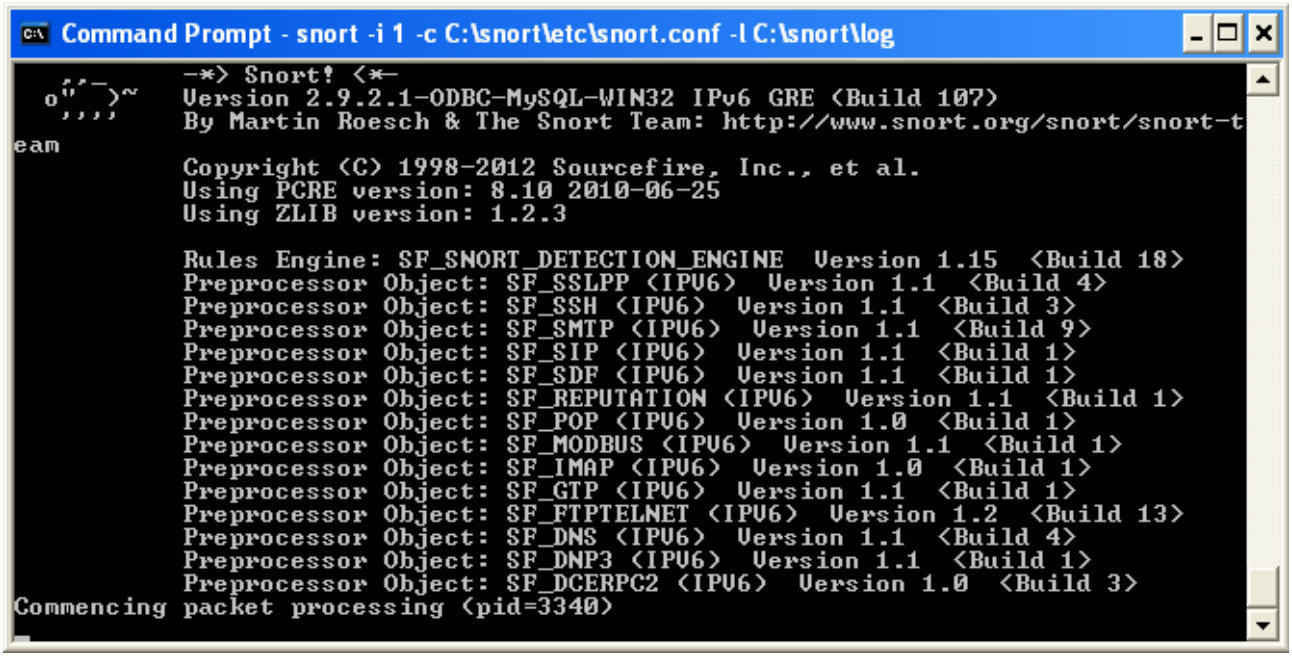
5. Next navigate to C:\snort\log in the windows explorer and delete any files you find there – apart from the Backup Folder as these will have resulted from previous IDS activity and new files will be created when SNORT runs this time.
6. Now go to the external machine (204.137.98.190) and start-up FileZilla (this is a useful FTP program with a GUI to make operation easier) and enter the IP address of the bank server machine in the host field of the initial window in FileZilla ***but do not attempt to connect yet***. Also start up Zenmap (for the port scan) on this same external machine and enter the IP

address of 204.137.98.145 in the target field, choose to run a profile - *Intense scan* ***but do not start the scan yet.***

7. *At this stage you can review what is being set up. The external machine will be running a port scan (Zenmap) and the victim bank server will detect and log this activity using SNORT. Next – having discovered vulnerabilities in this part of the exercise – we will further investigate sample vulnerabilities which will have become identified in the IDS port scans and alertlogs. Bear in mind these techniques and approaches are directed to network rather than application security which will be better addressed using the SNORBY GUI engine.*

Part I: (IDS tracking of port scan by SNORT)

8. From the bank server enter to commence the SNORT pre-processor engine. When SNORT starts up you should see a similar output to the screen image below and finally it will start processing the network packets and display as shown in the following diagram. SNORT has not fully loaded until you see "Commencing packet processing (pid=xxxx)". (See in window below).



SNORT Startup Screen

9. You are about to connect to the bank server using FileZilla which requires anonymous FTP access to be allowed on this machine (and will already be available from your earlier experiments). Check that this is in fact the case by (on bank server) going to IIS services from the shortcut on the desktop or by going Start → Control Panel → Administrative Tools → IIS Server (XP) or the equivalent on Windows 10. Naturally anonymous FTP should have been allowed for this to work.
10. Next go to the external host and click on Quickconnect in FileZilla to use anonymous FTP to connect to the FTP folder in the bank server. After the connection is successful start the Zenmap scan from the external machine.
11. Do a normal ping from a DOS prompt and then start the Zenmap scan and allow it to run to completion (takes some minutes) then navigate to the SNORT log folder on the bank server.
12. C:\Snort\log and see what information has been captured by SNORT and logged in the *postscan* and *alertlog* file as these are the most significant of the four files created.

Part II: (IDS tracking by SNORT of exploits resulting from discovery in Part I)

13. Navigate to the log folder on the bank server (204.137.98.145), i.e. C:\Snort\log but SNORT will have to be terminated (CTL C) before viewing *alertlog*.

    There will be folders/files named:

    - *alertlog (open this file with notepad and not wordpad)*
    - *Portscan (open with notepad)*
    - *merged*
    - *tcpdump.log.xxxxxxx (open this file with Wireshark which is on your server)*

14. Draw a diagram showing the machine upon which Zenmap and Nessus runs, the firewall and the machine upon which SNORT runs. Include all IP addresses so you demonstrate a clear and logical picture of the penetration testing and intrusion detection activities.

> Victim machine 204.137.98.145 machine running SNORT
>
> .. connected to ....
>
> Firewall interface 204.137.98.158 – internal and 204.137.98.189 – external
>
> Attack machine 204.137.98.190 running Zenmap and Nessus

15. Open the *Portscan* file using Notepad and examine the results resulting from SNORT's detection of the Zenmap scan and operation of FileZilla. List the open ports through which this scan has come and which have been detected by SNORT noting the source and destination addresses.

> Open ports from 204.137.98.190 (source) to 204.137.98.145 (destination) are:
>
> 21, 445, 139, 25, 135, 1062
>
> Two other open ports 1157 and 1158 likely to be logging ports to firewall administration logging host.
>
> There is a UDP port sweep at end on 169.254.163.160 -> 169.254.255.255 but this is likely to be related to underlying VM platform.

16. Next open up *alertlog* using Notepad.

    i) Observe how SNORT has reported on your ping as well as the ping resulting from running Zenmap. Use Wireshark and examine the protocol details of the two ping methods. From the alertlog file how does SNORT identify that a particular ping emanates from a tool like Zenmap in comparison with a normal ping?

    ii) What has SNORT observed about the use of FTP anonymous access and what port numbers were used at each end?

    iii) For what reason does SNORT class SMB (ports 137-8, 445) as potentially bad traffic?

    iv) SNORT has a classification of vulnerabilities by priority number. What priority levels did you observe? (See Snort Priority Classifications on following page)

(i) In alertlog a normal ping has a payload of about 32 bytes (abcdefg....) of padded data and the hex code is 0x0200. For the ping from Zenmap there is no payload and the identifier is 0x02e9. Check which is 0x2d49 too.SNORT also provided a warning for ICMP pings as possible "attempted information leakage"

(ii) SNORT has detected FTP anonymous access between ports 4604 (varies) on 204.137.98.190 and port 21 on 204.137.98.145

(iii) File sharing possible without login ID and password

(iv) See further details below but examples are classification of activities, UDP port sweep filtered at end of report, port scan with classification – attempted information leak – priority =2, NMAP ping with classification – attempted information leak – priority =2. There is also a Priority 0 for the FTP anonymous login!

17. Open the final file in the SNORT log folder (tcpdump.log.xxxxx) with Wireshark. Explain what you see from the traces of the packet exchanges and how this relates to what you observed in the Portscan and alertlog files.

Wireshark reports the details of each packet captured and transferred between source and destination IP addresses in a structured form which cannot be seen in the SNORT port scan and alertlog files, although detected connection attempts can be seen in both cases. There are some details captured by SNORT but not displayed in Wireshark.

The Wireshark packet exchanges show the actual information whereas SNORT port scan and alertlog files report on the potential harm / issues with these packets.

While SNORT alerted us to the different "perceived attacks" in the alertlog and port scanning operations in the portscan file, the corresponding Wireshark packet capture includes the actual traffic which flowed through the firewall.

The actual ICMP ping packets and the FTP traffic which arrived can be seen in the SNORT files. SNORT thus takes a packet capture at the time of perceived malicious activity while Wireshark provides a continuous window on all traffic which flows between specified IP addresses.

18. When you have finished using SNORT – navigate to the terminal window running SNORT and use Ctrl+C to shut it down.

Snort Priority Classifications (1 to 4)
SNORT User Manual 2.9.6 (http://manual.snort.org/snort_manual.html)

Classtype	Description	Priority
attempted-admin	Attempted Administrator Privilege Gain	high (1)
attempted-user	Attempted User Privilege Gain	high
inappropriate-content	Inappropriate Content was Detected	high
policy-violation	Potential Corporate Privacy Violation	high
shellcode-detect	Executable code was detected	high
successful-admin	Successful Administrator Privilege Gain	high
successful-user	Successful User Privilege Gain	high
trojan-activity	A Network Trojan was detected	high
unsuccessful-user	Unsuccessful User Privilege Gain	high
web-application-attack	Web Application Attack	high
attempted-dos	Attempted Denial of Service	medium (2)
attempted-recon	Attempted Information Leak	medium
bad-unknown	Potentially Bad Traffic	medium
default-login-attempt	Attempt to login by a default username and password	medium
denial-of-service	Detection of a Denial of Service Attack	medium
misc-attack	Misc Attack	medium
non-standard-protocol	Detection of a non-standard protocol/event	medium
rpc-portmap-decode	Decode of an RPC Query	medium
successful-dos	Denial of Service	medium
successful-recon-largescale	Large Scale Information Leak	medium
successful-recon-limited	Information Leak	medium
suspicious-filename-detect	A suspicious filename was detected	medium
suspicious-login	An attempted login using a suspicious username was detected	medium
system-call-detect	A system call was detected	medium
unusual-client-port-connection	A client was using an unusual port	medium
web-application-activity	Access to a potentially vulnerable web application	medium
icmp-event	Generic ICMP event	low (3)
misc-activity	Misc activity	low
network-scan	Detection of a Network Scan	low
not-suspicious	Not Suspicious Traffic	low
protocol-command-decode	Generic Protocol Command Decode	low
string-detect	A suspicious string was detected	low
unknown	Unknown Traffic	low
tcp-connection	A TCP connection was detected	very low (4)

Snorby and Security Onion

Security Onion is a Linux distribution for intrusion detection, network security monitoring, and log management. It is based on Ubuntu and contains:

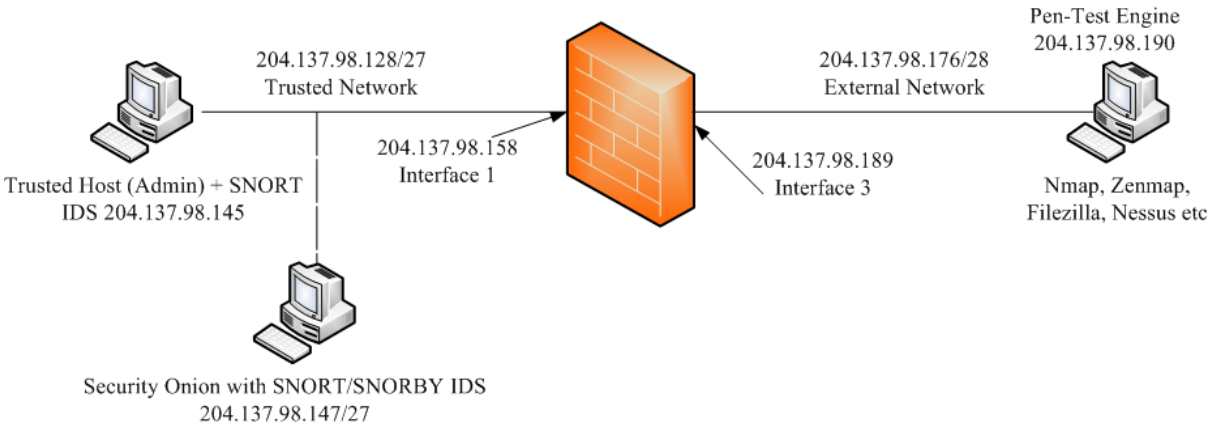
- Snort – is an open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS), with the ability to perform real-time traffic analysis and packet logging. Snort performs protocol analysis, content searching, and content matching. The program can also be used to detect probes and attacks, including operating system fingerprinting attempts, common gateway interface attacks, buffer overflows, server message block probes, and stealth port scans.
- Suricata – a recent variation on Snort.
- Bro – an open source Unix based network monitoring framework. Often compared to a Network Intrusion Detection System (NIDS), Bro can be used to build a NIDS and can also be used for collecting network measurements, conducting forensic investigations and for traffic analysis. Bro has elements of tcpdump, Snort, and netflow (CISCO).
- Sguil – a collection of software components for Network Security Monitoring (NSM) which operates by collection, analysis, and escalation of indications and warnings to detect and respond to intrusions as well as event driven analysis of IDS alerts. The Sguil client is written in Tcl/Tk. Sguil integrates alert data from Snort, session data from SANCP (Security Analyst Network Connection Profiler www.metre.net), and full content data from a second instance of Snort running in packet logger mode. Sguil's main component is an intuitive GUI that provides access to realtime events, session data, and raw packet captures.
- Squert – used to view NIDS/HIDS (Network/Host IDS) alerts and HTTP logs. Squert is a web application that is used to query and view event data stored in a Sguil database (typically IDS alert data). Squert is a visual tool that attempts to provide additional context to events through the use of metadata, time series representations and weighted and logically grouped result sets.
- Snorby – views and annotates Snort IDS alerts. It uses a Ruby on Rails application for network security monitoring and links to many of the items in this list and produces elaborate GUI displays to be studied in the workshop.
- ELSA – (Enterprise Log and Search Archive) which searches logs (IDS, Bro, syslog)
- Xplico –processes .pcap files in a similar way to NetwortkMiner and Wireshark
- NetworkMiner – is a Network Forensic Analysis Tool (NFAT) for Windows. It can be used as a passive network sniffer/packet capturing tool in order to detect operating systems, sessions, hostnames and open ports without putting any traffic on the network. NetworkMiner can also parse .pcap files for off-line analysis and to regenerate/reassemble transmitted files and certificates from .pcap files.

Security Onion is a platform that permits the monitoring of a network for security alerts. It is simple enough to run in small environments without many issues and allows advanced users to deploy distributed systems that can be used in network enterprise type environments. However its construction in to Virtual Machines and to get it operating is very complex.

Configuration

In the previous exercise the *Snort intrusion detection system* produced output via text files (Alertlog and Portscan). Although it is possible to feed these files into Snorby via the Security Onion suite – it is a complex and difficult process. Therefore we will run Snorby/Snort on a separate virtual machine (204.137.98.147) but still monitoring the trusted host (204.137.98.145). The following diagrams describe how to establish the Snort/Snorby engine.

Make sure that you still have the *pen-test.xml* configuration file loaded onto the firewall. Build and ping test this network below to match the following configuration diagram.



Start up the Snorby VM entitled ***Snorby_Master***

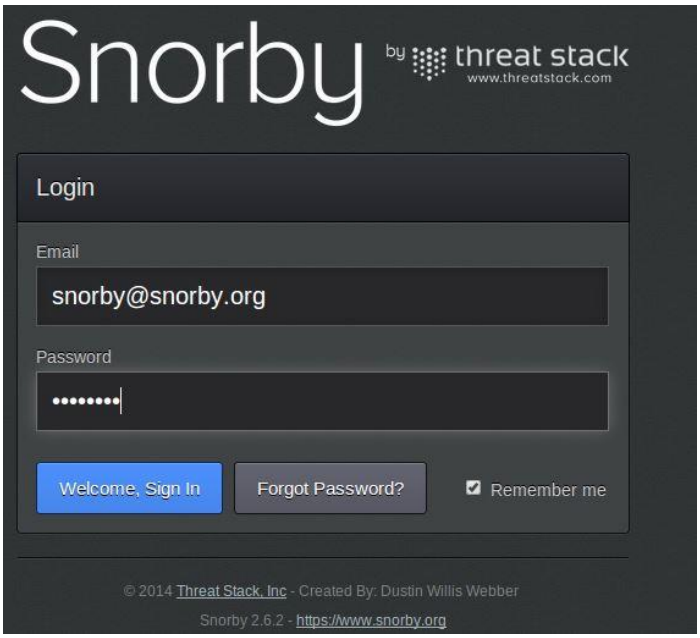
In answer to I moved it or I copied it respond: I *moved* it, as it is not a CentOS VM.

Login into Security Onion with ID/PW ***snorby/P#ssw0rd***

Ensure that the management interface is configured to: eth1: 204.137.98.147/27 and the monitoring interface (server) is configured to eth0: 204.137.98.145/27. Snorby runs a monitoring interface for traffic sent to 204.137.98.145.

Ensure that you can ping between the two Security Onion interfaces (204.137.98.145, 204.137.98.147), and the firewall interfaces (204.137.98.158). As a rule, Snorby does not respond to pings from external devices.

Click on and open the Snorby engine in Security Onion and login with ID/PW of ***snorby@snorby.org/password*** (different from previous password above!). You should now have the Snorby Dashboard displayed. With graphs displaying counts of High, Medium and Low Snort severity events.

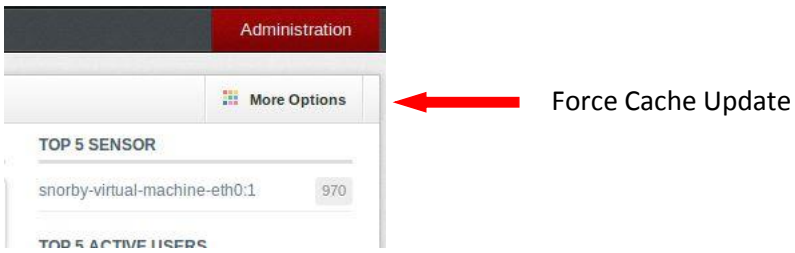


Re running the FileZilla and Zenmap

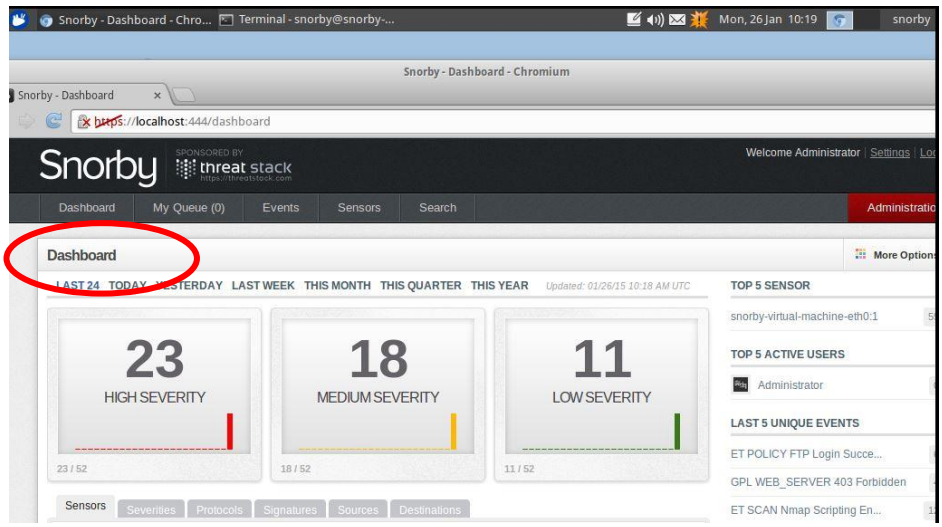
Go back to what you did before prepare FileZilla and Zenmap but without starting immediately. This relates to Steps 6 & 7 on p21-22. Now the FileZilla and Zenmap tests are directed at the server (204.137.98.145) and Snorby (204.137.98.147) is monitoring the 204.137.98.145 address and detecting intrusion according to the Snort rules and will now produce some interesting graphical output.

You may not see the High, Medium and Low Snort severity events graphs start immediately as there is always delay because of the very complex processing. If the message "current caching" is being displayed then processing is underway and you will have to wait. Remember that this is a 4.5GB full commercial graphical process in engine.

However it is possible to generate updating by selecting:

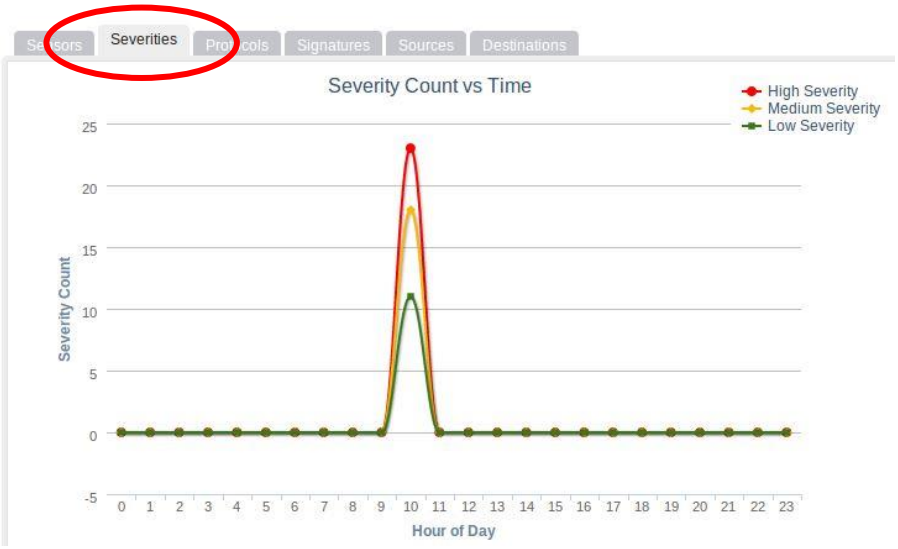


Once FileZilla and Intense Nmap scan has finished, you should see graphs along the lines of:

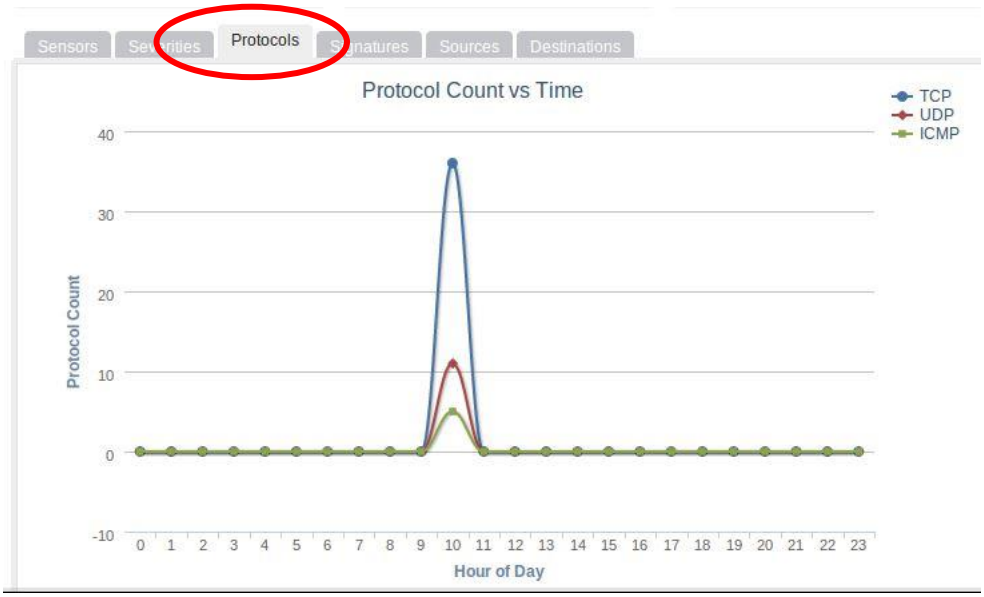


Dashboard – High, Medium, Low Severity

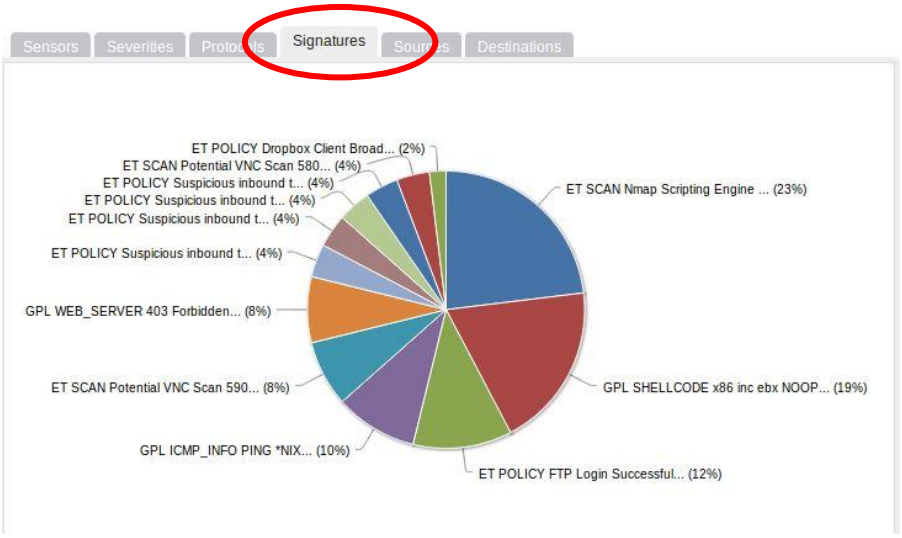
Event Count from eth0 Interface (204.137.98.145)



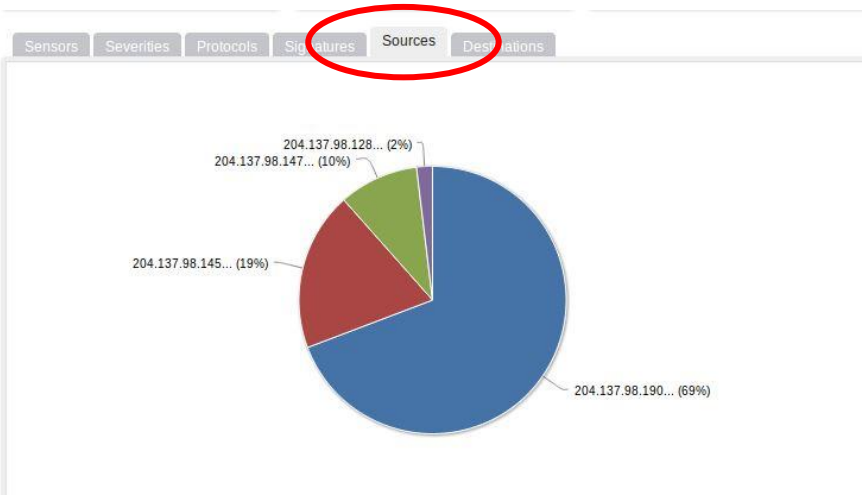
Severity (High, Medium, Low) Count from eth0 Interface



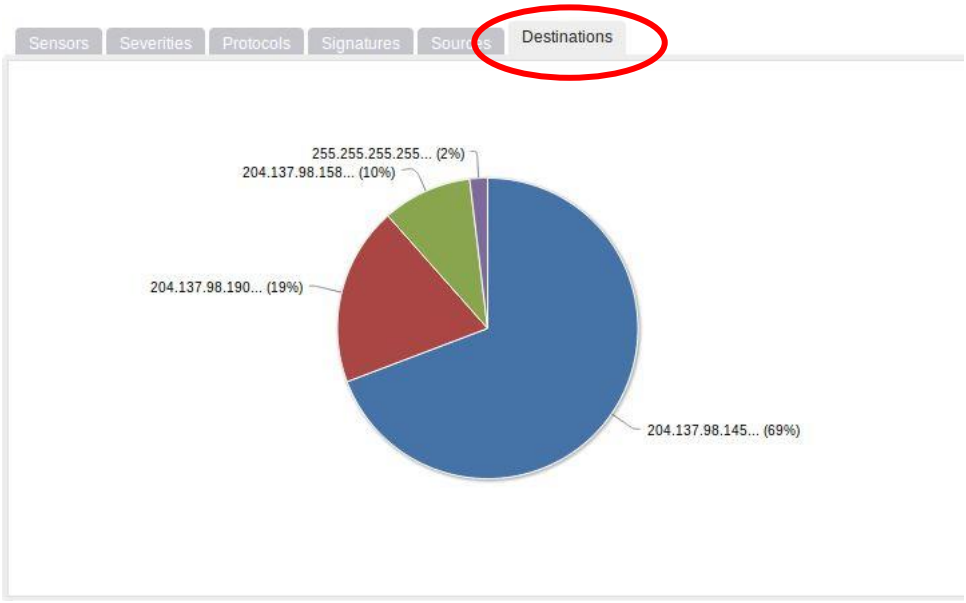
Protocols of Intrusion Scans at eth0 Interface



Signatures of Intrusion Scans at eth0 Interface



IP Source Addresses of Intrusion Scans at eth0 Interface



IP Destination Addresses of Intrusion Scans at eth0 Interface

Note 1: Because these statistics are cumulative and it is very difficult to reset Snorby counters and statistics, it is best to make a copy of your graph with the snipping tool and then compare it with the results at a later stage.

Note 2: ET Scan (Emerging Threat Scan)

Note 3: NoOp (NOOP) is the short form of No Operation and is a simple command that does not do anything to the results list, but can assist as a blind helper for some operations. Examples:

The first example checks the environment variable greeting to see whether it is set to HelloWorld, and if so, lets the scanner skip the following 10 lines during scans. The second example operates similarly, but tests only for the existence of an environment variable before skipping.

NoOp:"chkenv=greeting:HelloWorld,skipcount=10"

NoOp:"isenv=greeting,skipcount=10"

Exercises to be carried out.

1. Each scan e.g. intense, quick, regular, slow comprehensive, ping, quick traceroute etc …. and different uses of FileZilla will produce different responses from the Snorby engine. Capture your own dashboard statistics along with event severity count, protocols, signatures and IP addresses. Explain the risks (low, medium, high) based upon the statistics that you have collected in order that you can make a quantitative risk assessment. Try not to just produce a replica scan of the examples above. Now start the Nessus scan (page 7) and noting the new time, see if different or similar intrusions can be detected. Are there new signatures detected for example?

2. Include your own captured Severity Settings table (example below)

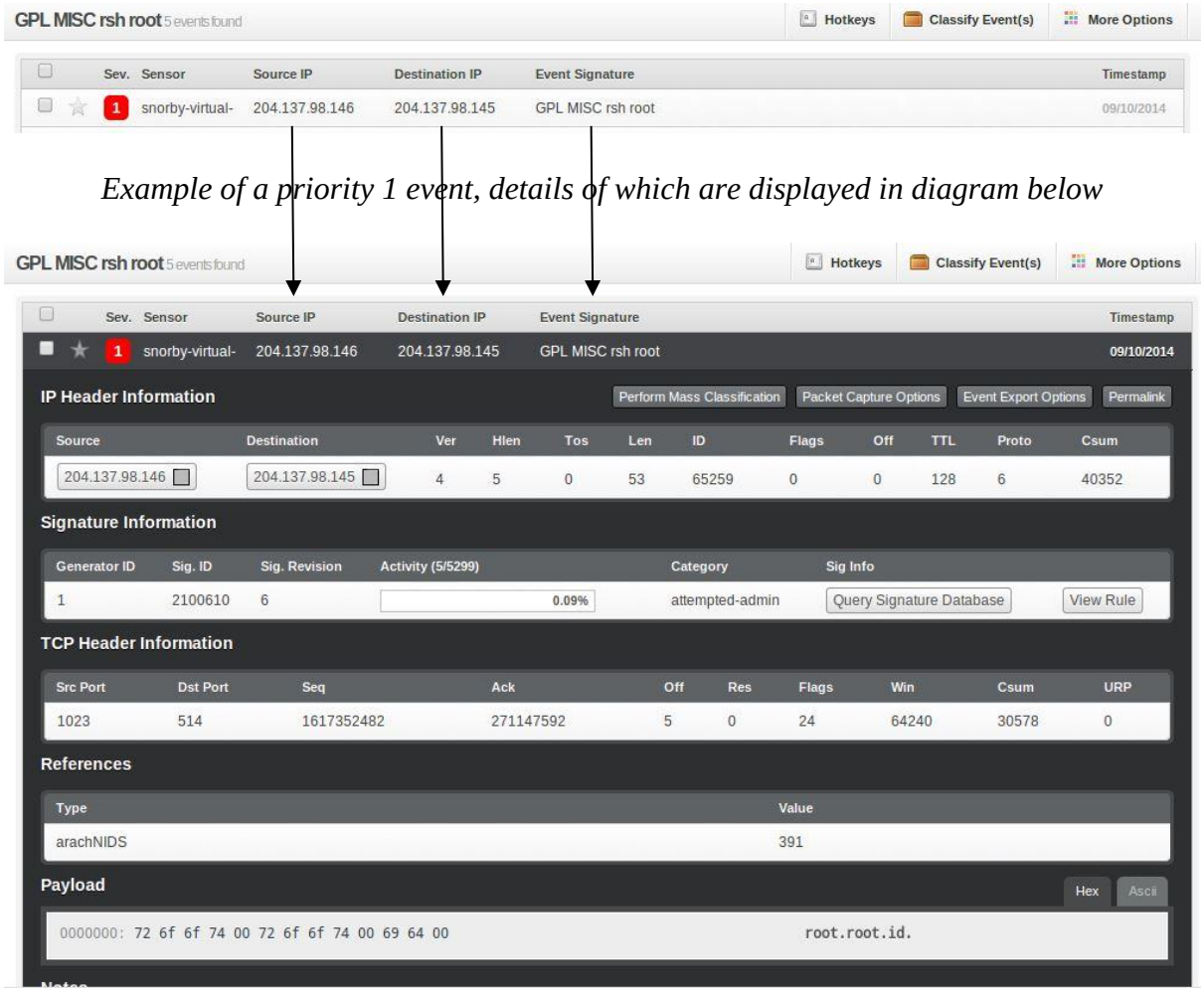
Severity Settings

ID	Name	Background	Text	Signature Count	Event Count	Example
1	High Severity	#ff0000	#ffffff	15	4,649	1
2	Medium Severity	#fab908	#ffffff	34	633	2
3	Low Severity	#3a781a	#ffffff	4	17	3

Snorby's three security settings (cf Snort Priority Classifications (1 to 4))

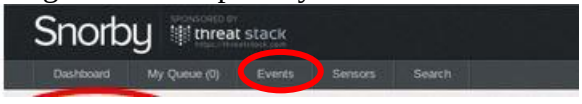
Now choose *three different* types of high, medium and low security events and explain why these intrusions might cause our network to be at risk. Make sure that each such event is of a different type. You will likely have to carry out some research to answer this question adequately.

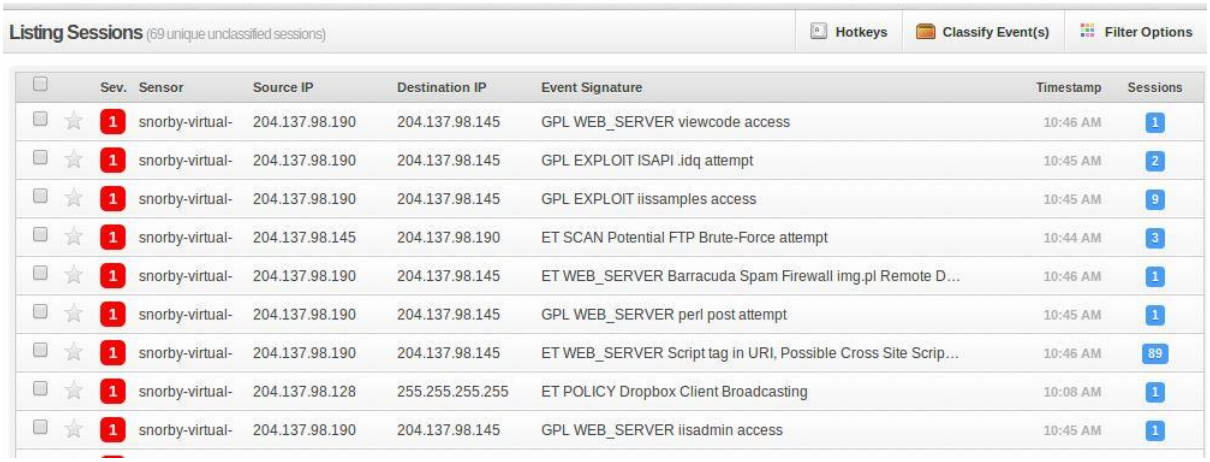
The diagram below demonstrates one such example – viz that of a high (Priority 1) security event involving the GPL MISC RSH root signature.

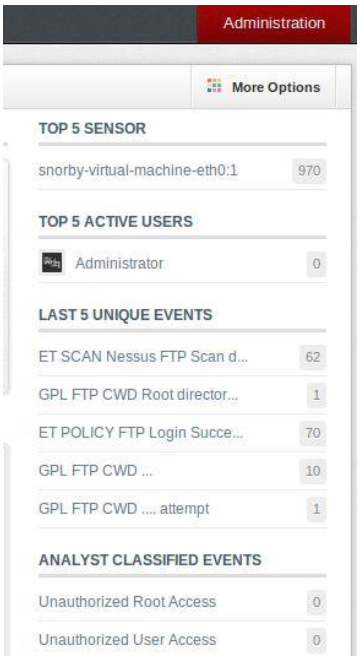


The first diagram below shows an example of Priority 1 events which need careful attention while the second diagram shows the last five categories of unique events and how many such events there were in each of these categories.

To get Events in priority order and click on "sev" in the top row to reorder:







Use “more options” and select *Force Cache Update* to refresh dashboard.

Speed Guide Port Analysis for SNORBY – you will need access to the Internet for this, e.g.

http://www.speedguide.net/port.php?port=80

Put in port number:

1433 Microsoft SQL Server
3306 MYSQL database server connection
5432 Postgres SQL database
1521 Oracle database default listener port for SQL

These ports can be seen from Administration → Signatures and they are grouped. Various port numbers can be seen. Also the pdf report (More Options → Export to pdf).

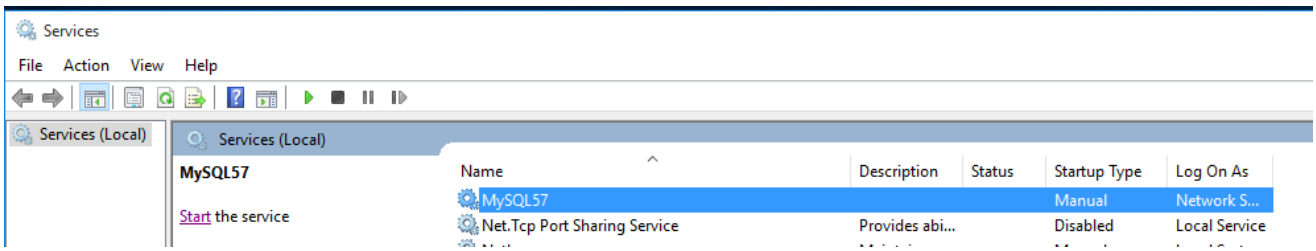
To get Events in priority order – select Events from the top row and click on “sev” in the top row of this screen to reorder.

An interesting questions is – “What response do we get from these ports on attempted connection?”

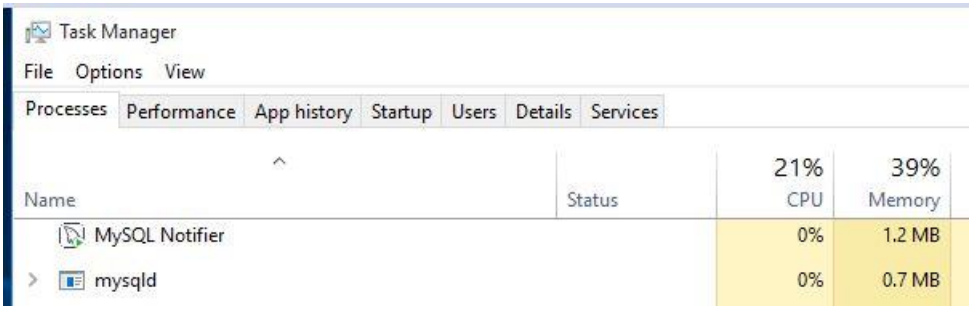
Experiment for access to an accessible service. Port 3306. MYSQL database server connection

Ensure you can ping between 204.137.98.145 and 204.137.98.190 through firewall.

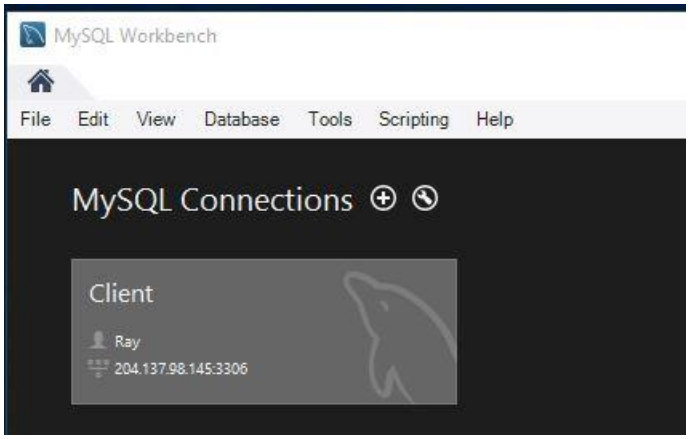
If not already running, startup MySQL Server (MYSQL57) on Rays Bank Server (204.137.98.145) from Control Panel > Administrative Tools > Services



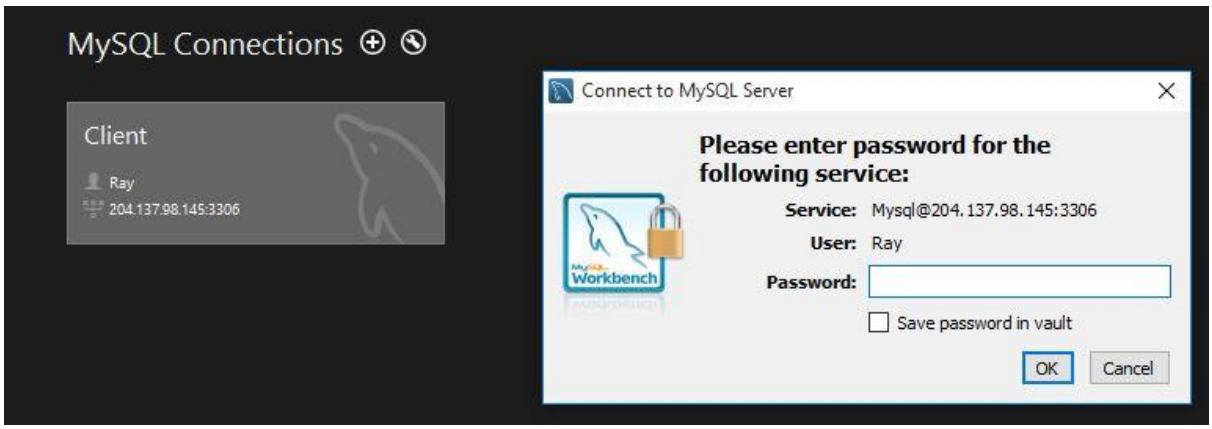
Check that MySQL Notifier and mysqld services are running on the server side from viewing Processes in Task Manager accessible from the Task Bar.



At a DOS prompt *netstat –a* should show port 3306 as listening. At this stage the server should be running without further action but one can double click on MySQL icon on the Task bar and bring up as follows:

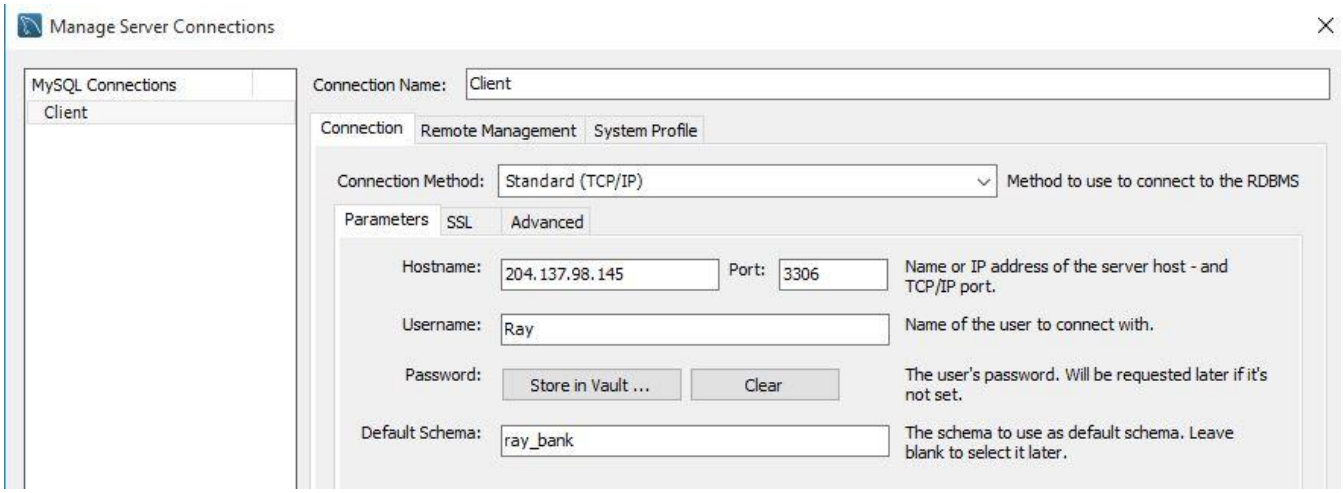


Now from Pen Testing machine (204.137.98.190) start client from task bar at bottom of screen then click on the Client gray box in the diagram below.



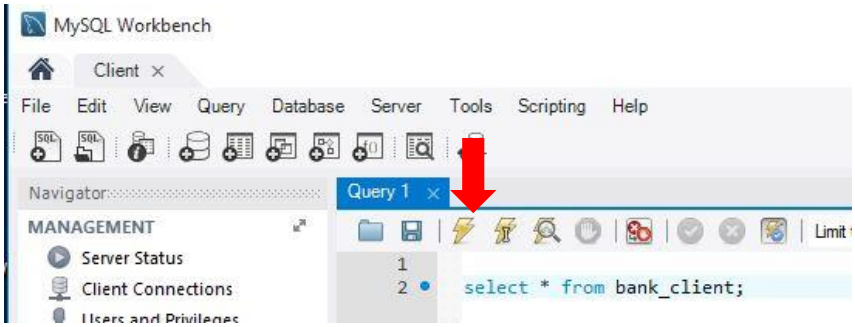
.. and enter P#ssw0rd. This password is the only security control between Ray's bank and the outside world! If connection fails try stopping and restarting the server on the bank (145).

To check the details of this client connection right click and edit connection on grey box above:

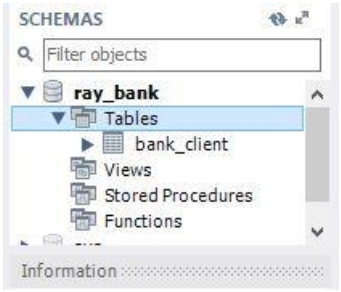


Notice the IP address of the bank server, port 3306, User name = Ray, Default Schema *ray_bank*.

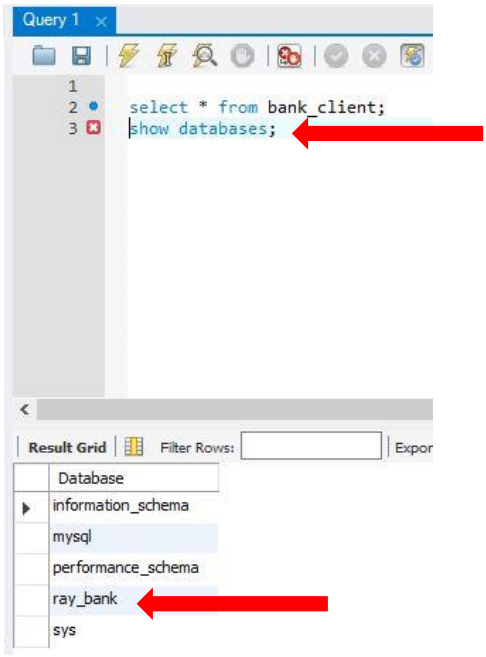
Click on connection to Rays Bank (gray box above) and this bring up a window on the SQL server. From here show the databases first (highlight second line (*show databases;*) and use execute arrow – red pointer) and then the clients in the database.



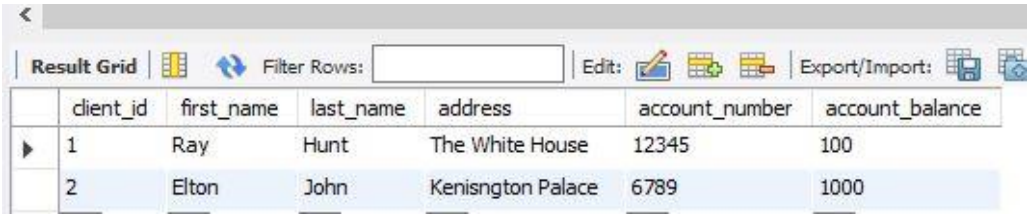
The *schema* bottom left is *ray_bank* but the *table* which holds the client's records is *bank_client*:



One can see the list of schema databases by selecting the whole line: (*show databases;)*



Highlight the line *select * from bank_client*; (notice ; at end of line) and click on the icon pointed to by the vertical red arrow shown in the earlier figure. At the lower part of this screen the details of two bank clients can be seen.



client_id	first_name	last_name	address	account_number	account_balance
1	Ray	Hunt	The White House	12345	100
2	Elton	John	Kenisngton Palace	6789	1000