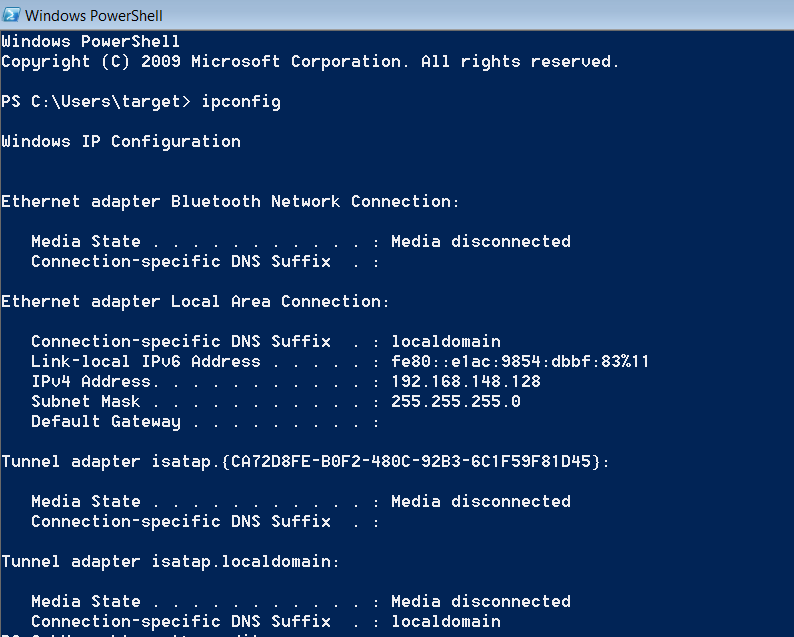
LLMNR / NBT-NS Responder -> Hashcat -> CME -> Empire

This is an example of how I got domain admin on my last engagement – with all sensitive information removed – for educational purposes only. I did not include in this document the admin hash I got from a box and logging onto the DC to dump Ntds.dit, but that should be self-explanatory after considering these methods. Please let me know if you have any questions.

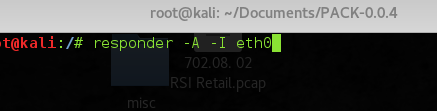
* First we have a vulnerable Win7 Box:



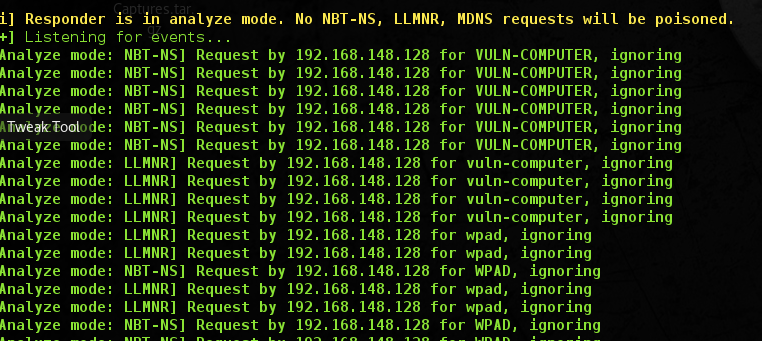
[Responder.py / Multirelay](https://github.com/lgandx/Responder-Windows)

This script works by spoofing common default protocols found within an AD environment. The protocols LLMNR and NBT-NS are used to find other shares and computers on the network, and additionally we are going to spoof a web proxy to take advantage of anyone using WPAD – a proxy auto-discovery protocol that is enabled by default in Internet Explorer.

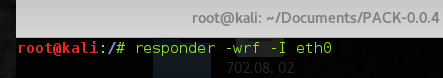
* We are going to start Responder in “Analyze” mode on interface ethernet0:



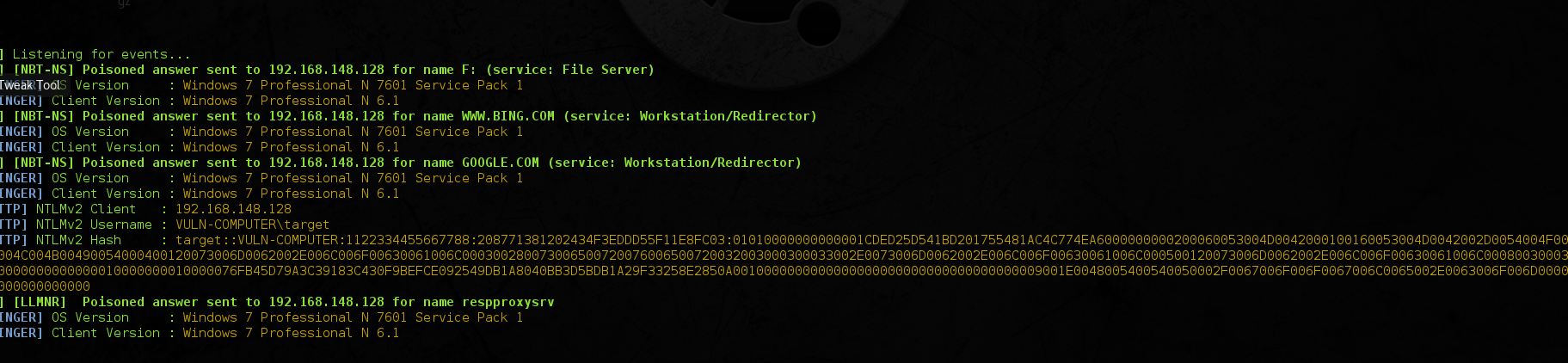
* Here we notice that this network is susceptible to this vector:



* We are going to put Responder.py in spoofing mode. LLMNR and NBT-NS spoofing are automatically enabled. We are also going to enable WPAD (-w), fingerprinting (-f) and redirecting (-r). Be cautious, on sensitive networks redirecting could cause issues.



* Looks like we got a hash:

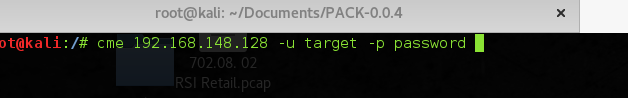


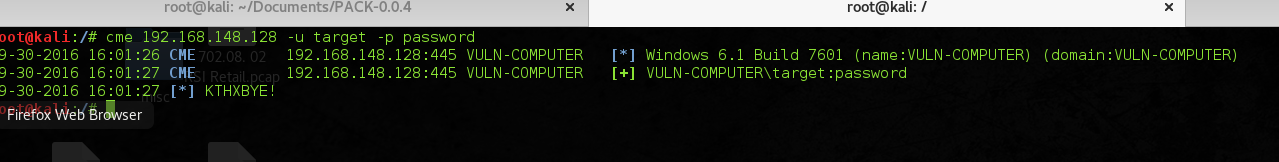
* When a computer searches across the domain for specific shares, other computers, or proxies, this is what we’ll see. Occasionally we can get plain text credentials from proxy authentication, but we didn’t get anything this time.
* The next step is to bring the hash over to the password cracking server and attempt to crack it. Please see Hashcat document for those tips.
* It turns out that our user’s credentials were target::password. That’s…not great.

[CrackMapExec](https://github.com/byt3bl33d3r/CrackMapExec)

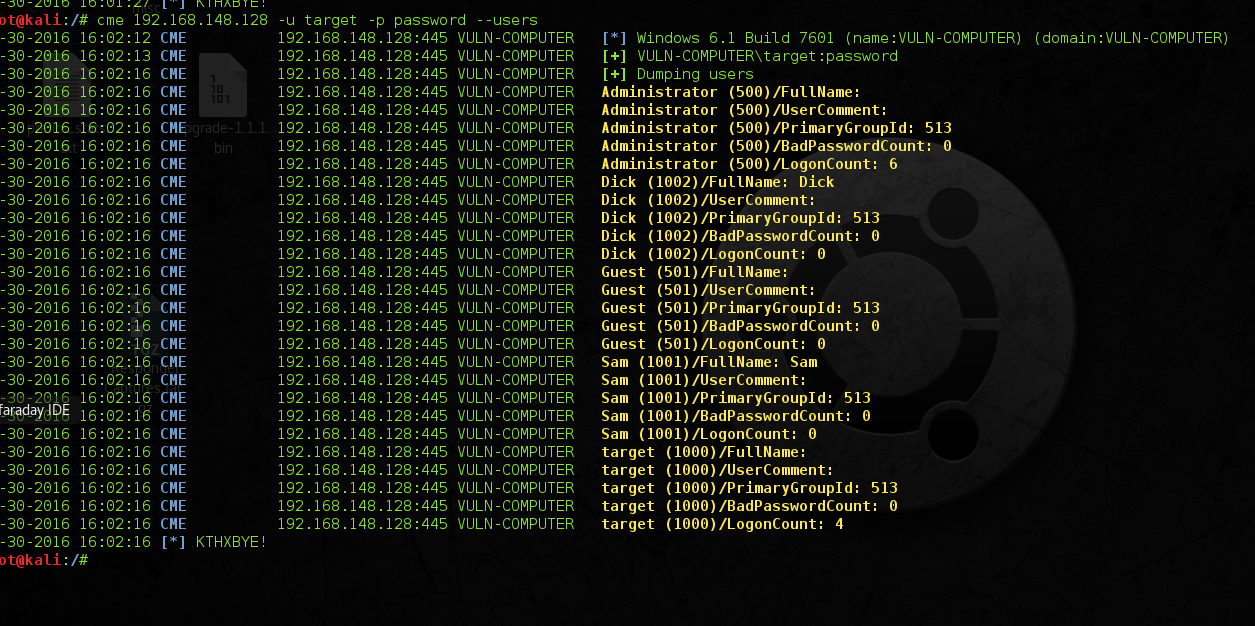
In order to start getting shells with these credentials and move laterally, we’re going to use CrackMapExec. This tool is purely post exploitation. You must have either a NTLM hash for PTH or regular credentials.

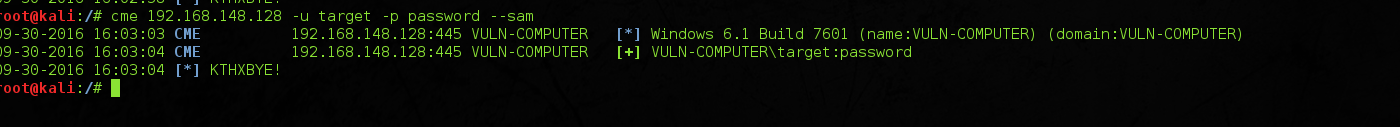
* Let’s see if this user has privileges:





* Looks like our credentials worked. Let’s enumerate users and hopefully dump the SAM database:

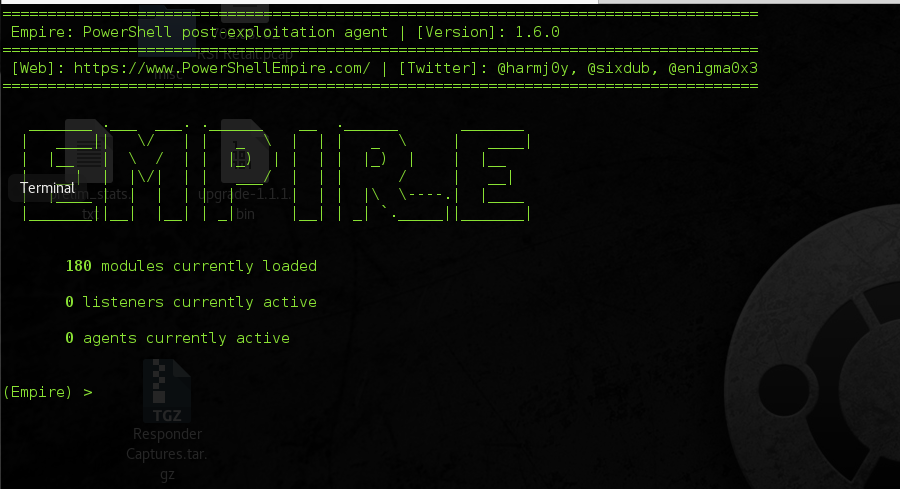




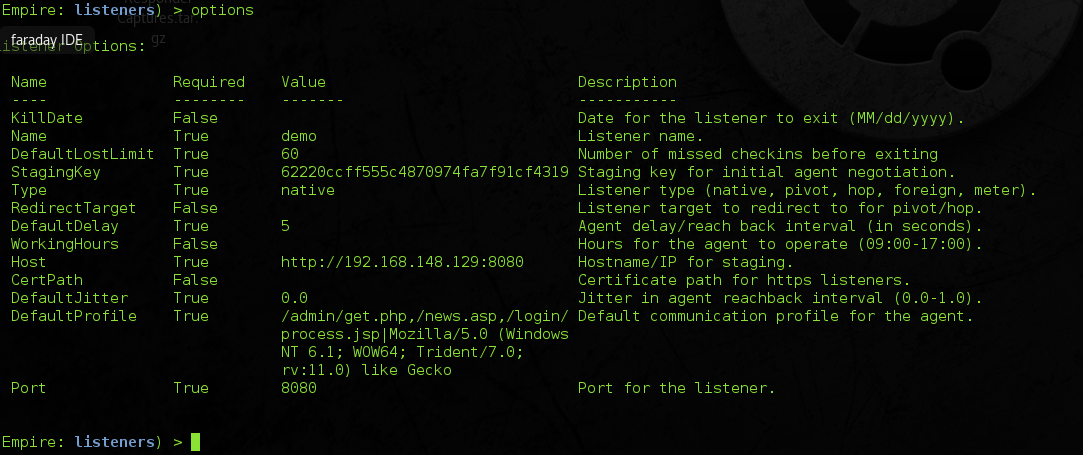
* No luck, no SAM database for NTLMhashes.

[Empire Framework](https://github.com/EmpireProject/Empire)

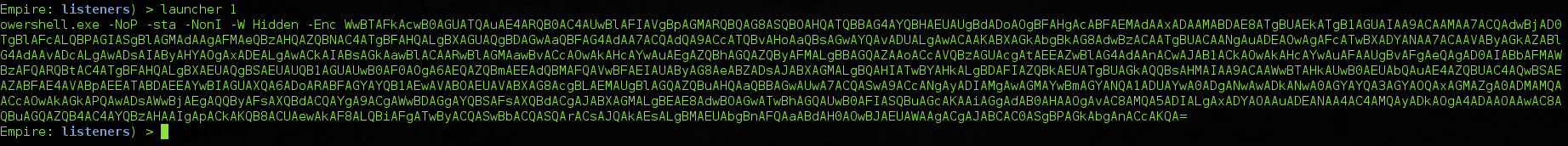
We are going to use a combination between Empire and CrackMapExec, Empire is a full Powershell post exploitation framework. Much like Metasploit, it has built in modules:



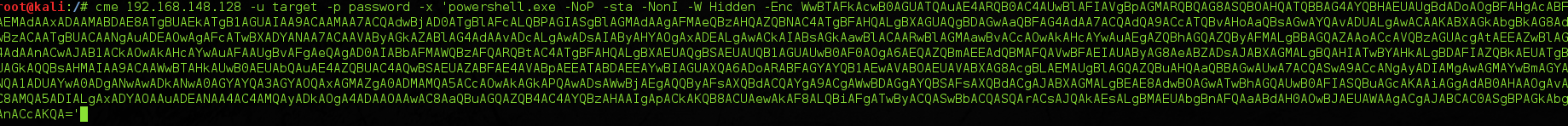
* It works by using Powershell Remote Access Trojans (RATs) – the advantage of this is it will not be flagged by AV because it uses windows native API calls.
* Here are our options for out Trojan. After verifying them, type “run” and hit enter:



* Much like Metasploit, after we created our listener, we can create a PowerShell launcher code that injects the Trojan into a process on the RHOST.



* We can use CrackMapExec to inject this code by copying it. The -X flag means PowerShell command. Make sure to quote this w/ ‘ ‘



* Hit enter and wait for your framework to receive a response, and you should have your active agent.