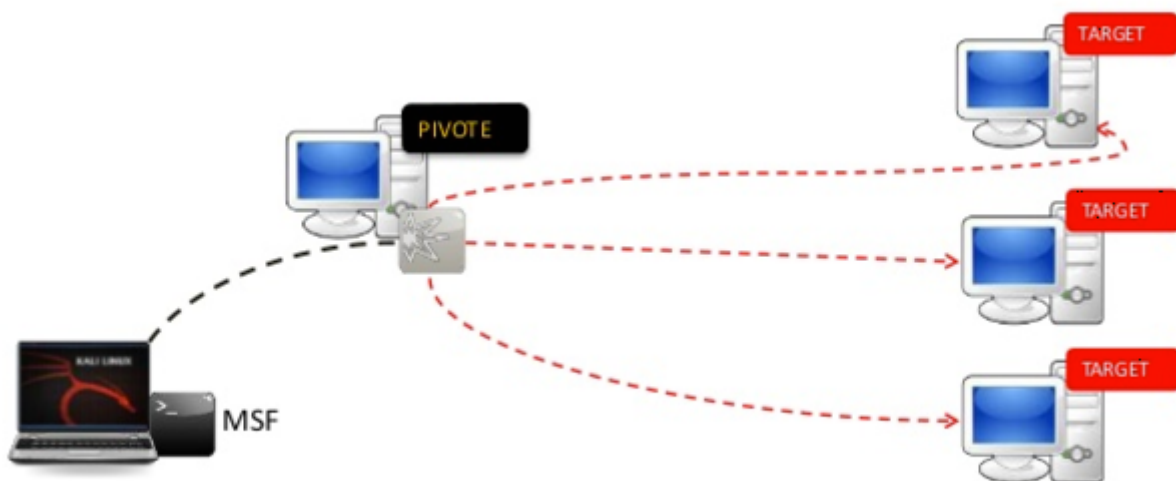


# [Pivoting] Port forwarding

## Introduction

Dans le cas d'une compromission d'une infrastructure où une machine cible n'est pas directement accessible, il va falloir établir un **tunnel** entre vous et la machine cible en passant par une machine intermédiaire, qui a elle, accès à la machine cible.

On va devoir mettre en place un **reverse proxy** sur la machine intermédiaire pour pouvoir router les paquets sur la ou les machines cibles.

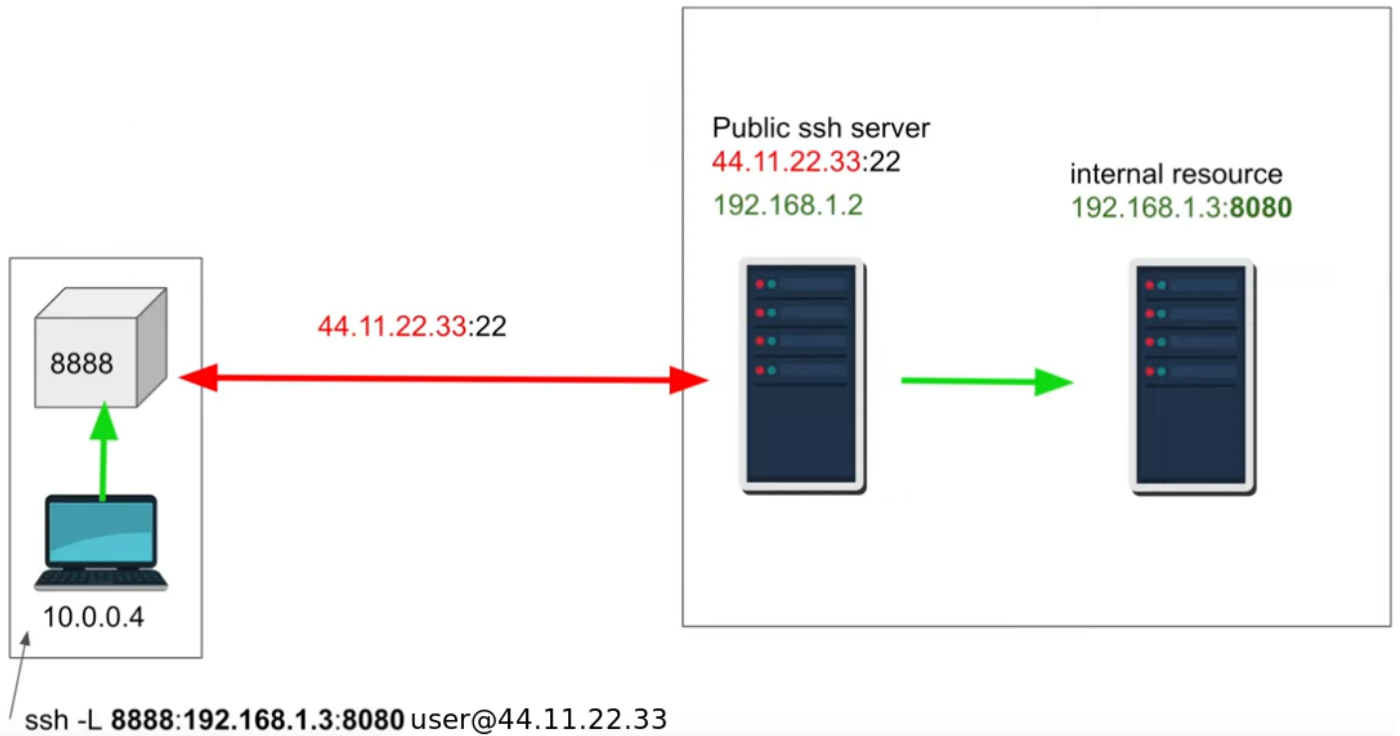


## SSH

Il est possible de faire du port forwarding avec le service SSH.

La machine intermédiaire doit être munie d'un serveur SSH et vous devez avoir des accès SSH pour vous connecter dessus.

## Local port forwarding



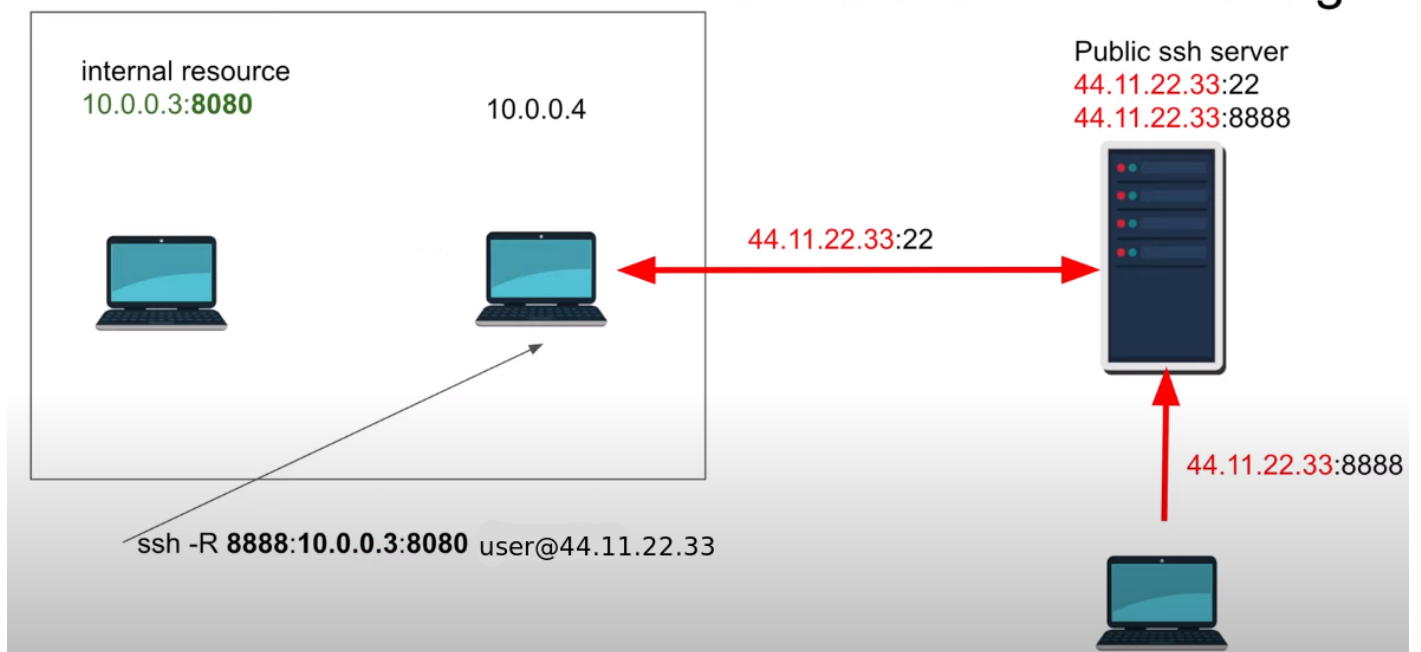
```
ssh -L <LOCAL_PORT>:<DESTINATION_IP>:<DST_PORT> <REMOTE_USER>@<REMOTE_IP>
```

Vous devriez pouvoir accéder à votre service via **localhost:<LOCAL\_PORT>**

**Le tunnel restera ouvert tant que la session SSH est active.**

## Remote port forwarding

### Remote Port Forwarding



```
ssh -R <LISTENER_PORT>:<DST_IP>:<DST_PORT> <LISTENER_USER>@<LISTENER_IP>
```

Cette commande exposera le service du **DST** sur le **LISTENER**.

# Chisel

## Prérequis

- Trouvez un port non utilisé (absent de cette liste) :

```
(netstat -pnta || ss -n -t -p -u)
```

## Installation

Voici le lien github de l'outil :

- <https://github.com/jpillora/chisel/>

Vous pouvez installer l'outil sur le poste intermédiaire si vous avez les droits **root** grâce à la commande suivante :

```
curl https://i.jpillora.com/chisel! | bash
```

Téléchargez le binaire (version 1.9.1) :

```
curl -O -L https://github.com/jpillora/chisel/releases/download/v1.9.1/chisel_1.9.1_linux_amd64.gz && gunzip  
chisel_1.9.1_linux_amd64.gz && mv chisel_1.9.1_linux_amd64 chisel && chmod +x chisel
```

## Port forwarding

- Sur le pc intermédiaire, lancez la commande suivante :

```
./chisel server -p <LISTENER_PORT>
```

- Puis sur le pc de l'attaquant, exécutez la commande suivante :

```
./chisel client <LISTENER_IP>:<LISTENER_PORT> <LOCAL_PORT>:<TARGET_IP>:<TARGET_PORT>
```

Sur la machine de l'attaquant vous pourrez accéder au service via  
**localhost:<LOCAL\_PORT>**

## Reverse port forwarding

- Sur le poste de l'attaquant :

```
./chisel server <INTERMEDIATE_IP> -p <INTERMEDIATE_PORT> --reverse
```

- Puis sur le pc intermédiaire, exécutez la commande suivante :

```
./chisel client <LISTENER_IP>:<LISTENER_PORT> R:<LOCAL_PORT>:<TARGET_IP>:<TARGET_PORT>
```

Sur la machine de l'attaquant vous pourrez accéder au service via  
**localhost:<LOCAL\_PORT>**

# Socat

## Installation

Voici le Github du projet :

- <https://github.com/3ndG4me/socat>

Pour télécharger le binaire :

```
curl -o socat -L https://github.com/3ndG4me/socat/releases/download/v1.7.3.3/socatx64.bin && chmod +x socat
```

Sinon, installez le depuis les dépôts :

```
apt install -y socat
```

## Port forwarding

```
socat TCP-LISTEN:<LOCAL_PORT>,fork TCP:<REMOTE_IP>:<REMOTE_PORT>
```

Le service de la machine **REMOTE** sera exposé sur le **LOCAL\_PORT** de la machine qui exécute la commande.

# Sshuttle

Ce logiciel permet d'établir une connexion VPN à travers un tunnel SSH sans nécessiter de privilège root sur la machine distante :

- <https://github.com/sshuttle/sshuttle>

```
sshuttle -vr <REMOTE_BRIDGE> <REMOTE_NETWORK>
```

Tous le trafic sera routé par le réseau distant !

Voici un exemple :

```
sshuttle -vr username@target-ip 10.1.1.0/24
```

# Netcat

Il est possible de faire du port forwarding avec Netcat grâce à certaines options.

Depuis la machine attaquante :

```
nc -lv --broker --max-conns 2
```

Et sur la machine intermédiaire :

```
nc -nv <ATTACKER_IP> 31337 -c 'nc -nv <TARGET_IP> <TARGET_PORT>'
```

# Proxychains / SSH Forwarding

Cette technique va vous permettre d'effectuer des scans nmap en passant par la commande proxychains qui va faire passer les paquets dans le tunnel SOCKS SSH, ce qui n'est pas possible avec les autres techniques car elles permettent d'exposer qu'un seul port.

Tout d'abord, créer le tunnel SSH :

```
ssh -D <LOCAL_PORT> -f -N <BRIDGE_USER>@<BRIDGE_IP>
```

Ensuite, créez votre configuration proxychains dans le fichier **/etc/proxychains4.conf** :

```
tcp_read_time_out 800
tcp_connect_time_out 800

[ProxyList]
socks4 127.0.0.1 <LOCAL_PORT>
```

Désormais, vous pouvez lancer votre scan nmap en passant par la commande proxychains :

```
proxychains nmap -Pn -p- --top-ports 10 -sT <TARGET_IP>
```

---

Revision #18

Created 31 October 2023 15:38:29 by Elieroc

Updated 7 May 2024 09:37:09 by Elieroc