**Sending Files to Remote using SCP**

* Copying "file1.txt" from the local client to a remote client.
* Executed on the Client

SCP "file\_to\_send.txt" user@dst.ip:directory/location

**Receiving Files from Remote using SCP**

* Copy "file2.txt" from a remote client to the local client
* Executed on the Client
* Scp user@dst.ip:file\_to\_get.txt /local/directory
* Use '-P' and @localhost when a previously established ssh tunnel was setup
  + Ex. scp -P 1234 user@localhost:/file/to/rcv.txt /dst/location

**Covert Channels**

* Protocol Headers: modulation, manipulation, options
* Data Field
  + ICMP allows you to write hex to the data portions with (-p) switch
  + Demo: ping #.#.#.# -p <char>
  + \*Standard packet size 64 bytes
  + tcpdump -tttt -vvv -XX "icmp"
    - 'X' shows hex and ASCII
    - 'x' shows only hex

BOTNET (Another Covert Channel)

* Master/Zombies
* Used for: Spam, DDOS, Spyware, Virus, Clickfraud
* Notorious Examples: Conflicker, Zeus, TDL-4 (Alureon)
* Commonly used protocol for C2: HTTP, IRC, UDP, DNS over TCP

Describe protocol swapping

* DNSCAT2 is just one of many
* Most often, common protocols are used to hide devious actions
* Common Botnet C2 protocols are also popular choices for programs that provide protocol swapping

**Data Exfiltration**

* Covertness = function of Capacity of the Medium/Transmission Rate
* Most common methods of data exfiltration: HTTP/HTTPS, DNS, FTP, RDP, SMTP/email, SMB, TOR
* Techniques to help obfuscate the exfiltration: Compression, Encoding, Encryption

**File Transfer Basics**

Stdin (channel 0) / Stdout (channel 1)

Pipes

* Fifo in nature
* Can live longer than the process it facilitates
* Can use it for a pass-through in data transfers

Netcat

* Listener (receive file): nc -l -p 9001 > newfile.txt
  + Creates listener for other hosts to connect to
* Client(sends file): nc 10.0.0.2 9001 < file.txt
* Must ALWAYS setup listener first
* By default cannot encrypt data/information
* **Netcat Relays**
  + On listener Relay:
    - Mkfifo pipe OR mknod mypipe p
    - nc -lp 9001 0< mypipe | nc -lp 9002 1> mypipe
  + On Client 1 (sends info):
    - nc X.X.X.X 9002 < infile.txt
  + On Client 2 (receives info):
    - nc X.X.X.X 9001 > outfile.txt
    - Writes the output only to listener1 and listener2 through the named pipe

**SSH Tunnelling / Local Port Forwarding**

Flushing iptables (wiping all iptable rules/configurations)

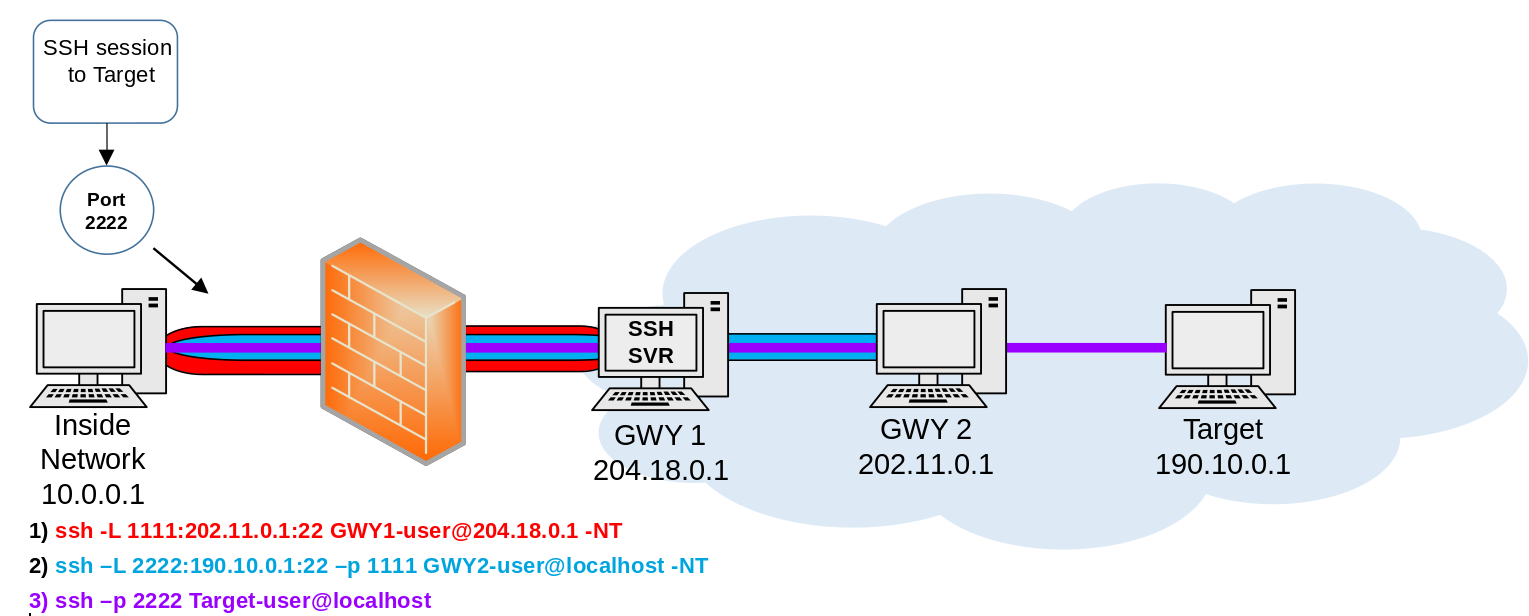
sudo iptables -F

\*\*\*\*\*\*\*\*\*\*\*THIS IS A MUST KNOW TOPIC

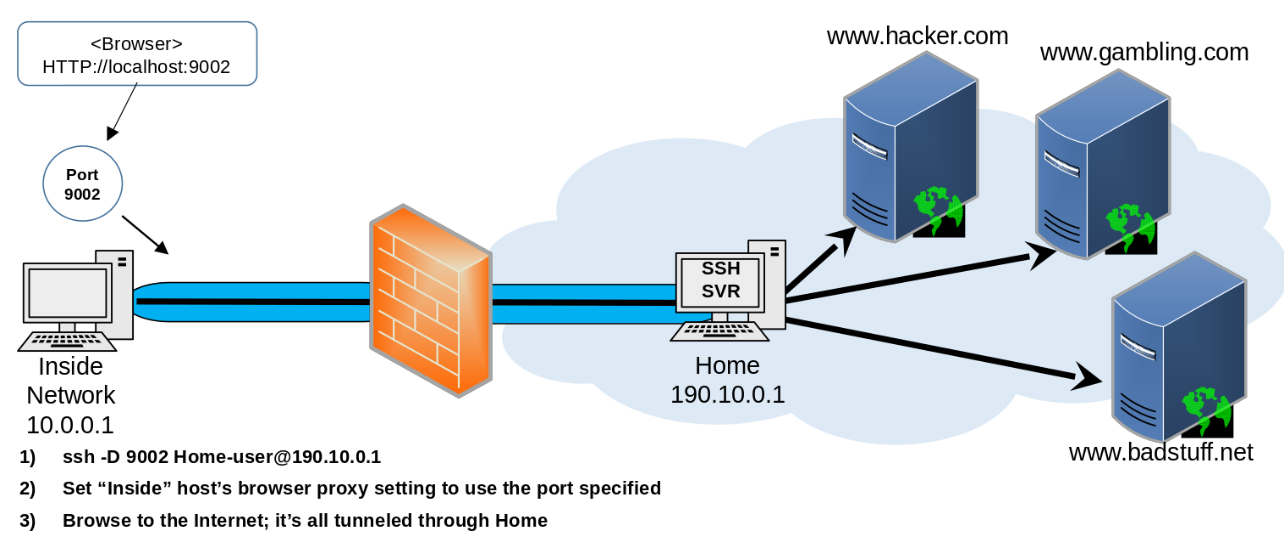
* <https://www.ssh.com/ssh/tunneling/>
* Ssh @<ip host>
  + Ssh user2@box2
* -L<port>:<ip/dst>:<port dst>
  + -L 1111:Box3:22
    - Ssh user@rmt.ip -L 1234:<tgt.ip>:80 -NT #Example

Tunneling within SSH tunnel

* ssh -L 1234:190.10.0.1:22 -NT inside-user@204.18.0.1
* ssh -p 1234 home-user@localhost # -p option allows to specify desired port



Dynamic Port Fowarding



# Don't forget to add -NT at the end of your statement

* Ssh -D 9002 home-user@190.10.0.1 -NT

Proxy Chains

1. ssh -D 9050 user@200.0.0.1 -p22 -D 9050 -NT
2. Proxychains nc -nvz 200.0.0.2 9001 # connecting to remote netcat listener

Remote Port Fowarding

Inside host initiates the tunnel to tell the remote host where to forward its port to (through the tunnel)

* ssh -R 9001:10.0.0.57:80 home@190.10.0.1 -NT

