



LECTURE 7:

INTRUSION DETECTION TECHNIQUES AND NETWORK FORENSICS

*Part 2: Investigating
Network Traffics*

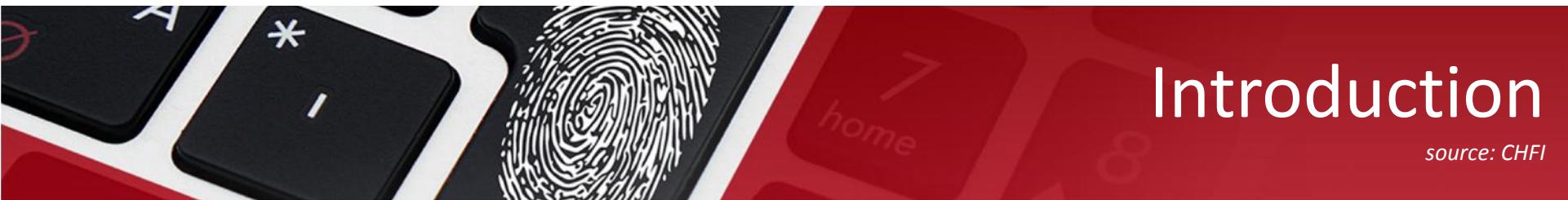
References:

Guide to Computer Forensics and Investigations
Certified Ethical Hacker and Forensic Investigations



Objectives

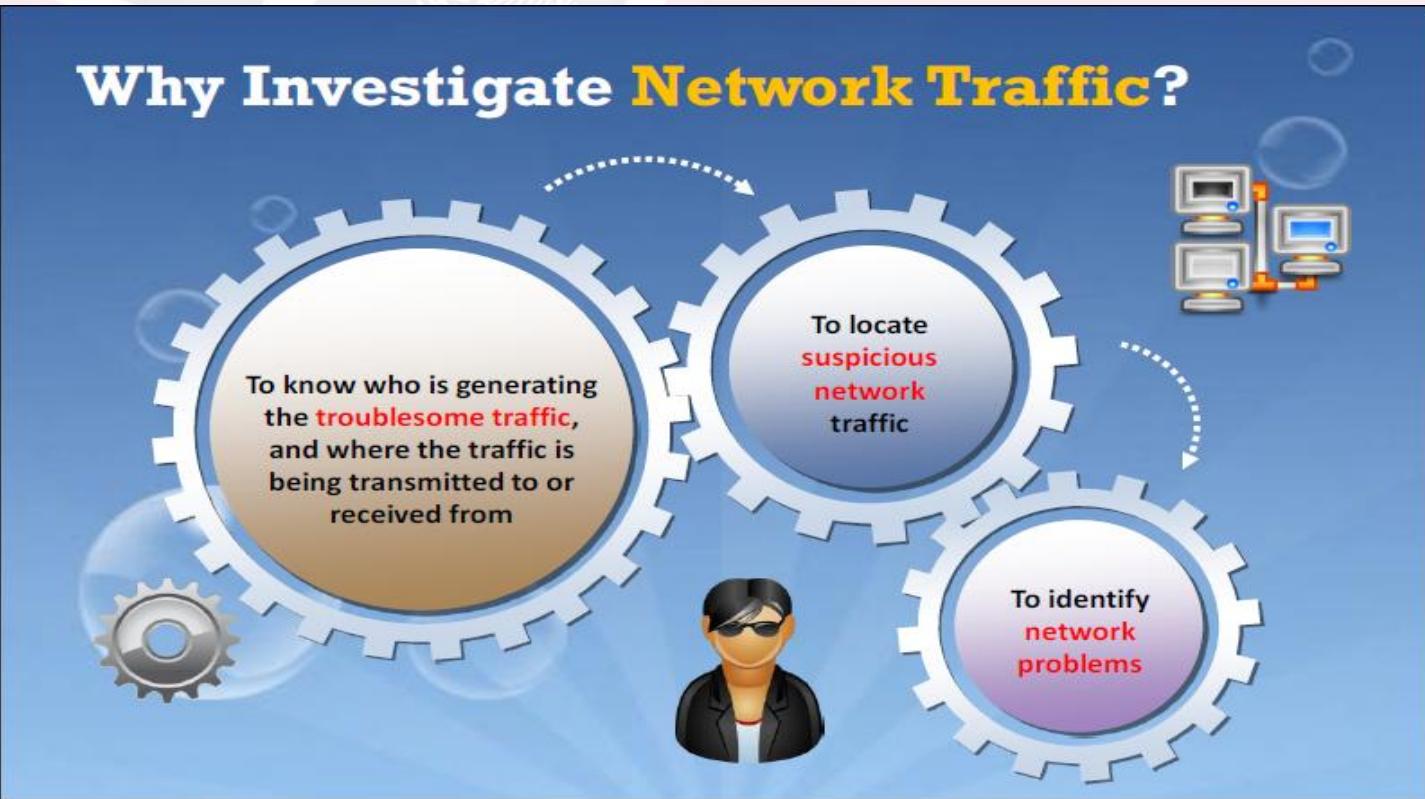
- Understanding the network protocols
- Identifying the sources of evidence on a network
- Understanding the Physical and data-link layer of the OSI model
- Gathering evidence at the physical and data link layer
- Understanding Network and transport layer of the OSI model
- Gathering evidence at the network and transport layer
- Collecting evidence and documenting on a network
- Gathering evidence on a network
- Documenting the gathered evidence on a network
- Evidence reconstruction for investigation
- Identifying network forensic tools



Introduction

source: CHFI

Why Investigate Network Traffic?



To know who is generating the **troublesome traffic**, and where the traffic is being transmitted to or received from

To locate
suspicious network traffic

To identify
network problems





Network Addressing Schemes

source: CHFI

LAN Addressing

Each node in LAN has a MAC address that is factory-programmed into its NIC.

Data packets are addressed to either one of the nodes or all of the nodes.

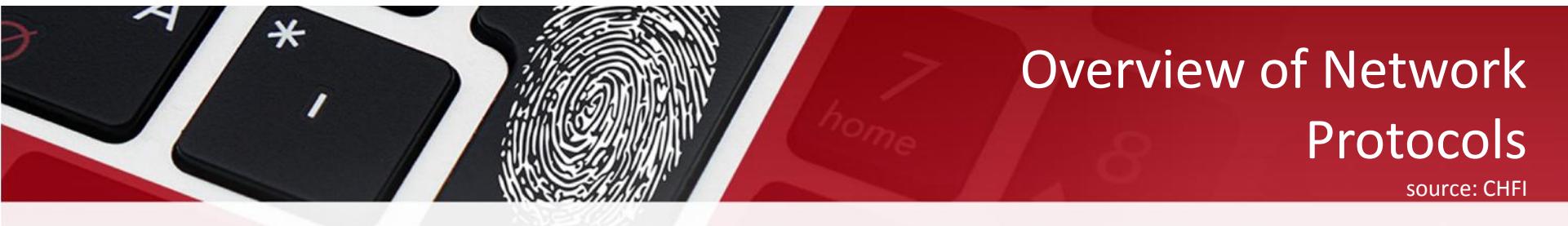
There are **two types** of network addressing schemes:

Internetwork Addressing

Internetwork is a collection of LANs and/or other networks that are connected with routers.

Each network has a unique address and each node on the network has a unique address, so an address is network address/node address combination.

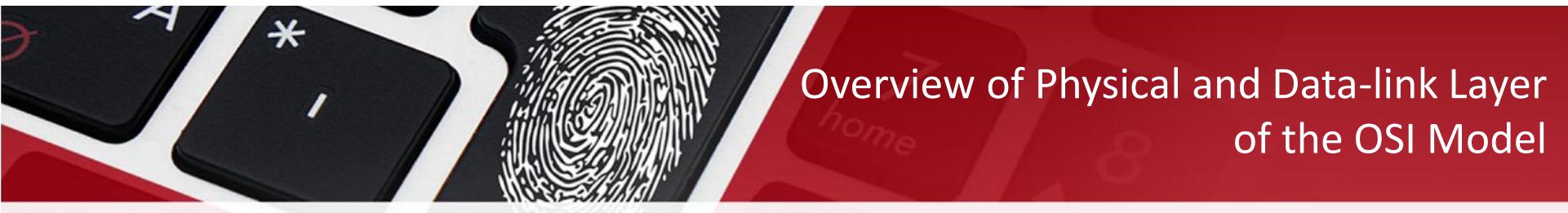
IP is responsible for network layer addressing in the TCP/IP protocol.



Overview of Network Protocols

source: CHFI

	Data Unit	Layer	Function	Protocols
Host Layer	Data	Application	Network process to application	HTTP, SMTP, NNTP, TELNET, FTP, NMP, TFTP
		Presentation	Data representation and encryption	
		Session	Interhost communication	
	Segments	Transport	End-to-end connections and reliability	UDP, TCP
Media Layer	Packets	Network	Path determination and logical addressing (IP)	ARP, RARP, ICMP, IGMP, IP
	Frames	Data Link	Physical addressing (MAC & LLC)	PPP, SLIP
	Bits	Physical	Media, signal and binary transmission	



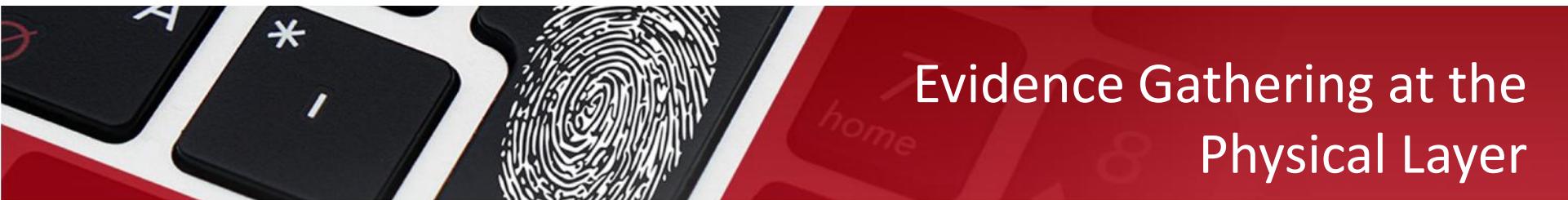
Overview of Physical and Data-link Layer of the OSI Model

Physical layer:

- It helps in transmitting data bits over a physical channel.
- It has a set of predefined rules that physical devices and interfaces on a network have to follow for data transmission to take place.

Data-link layer:

- It controls error in transmission by adding a trailer to the end of the data frame.



Evidence Gathering at the Physical Layer



Sniffer is computer software or hardware that can intercept and log traffic passing over a digital network or part of a network.



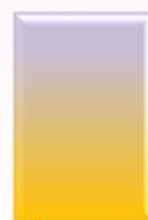
Sniffers, which put NICs in promiscuous mode, are used to collect digital evidence at the physical layer.



SPANned ports, hardware taps help sniffing in a switched network.



Sniffers collect traffic from the network and transport layers other than the physical and data-link layer.



Investigators should configure sniffers for the size of frames to be captured.

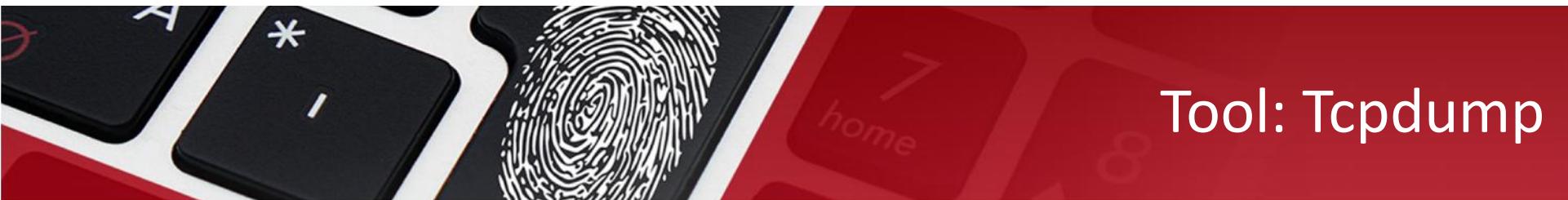


Evidence Gathering: Sniffing

source: CHFI

Evidence Gathering via Sniffing





Tool: Tcpdump

Tcpdump allows to sniff network packets and make statistical analysis of these dumps.

It operates by putting the network card into promiscuous mode.

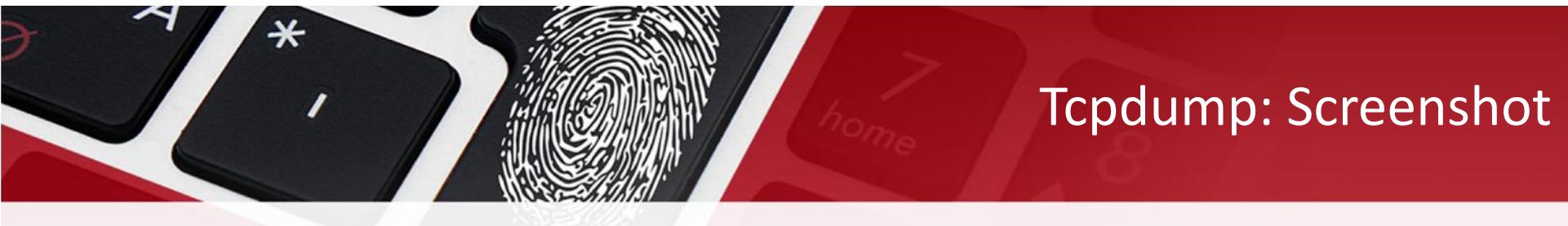
It may be used to measure the response time, packet loss percentages, and view TCP/UDP connection establishment and termination.

Tcpdump report consists of:

- Captured packet count.
- Received packet count.
- “dropped by kernel” packets count.

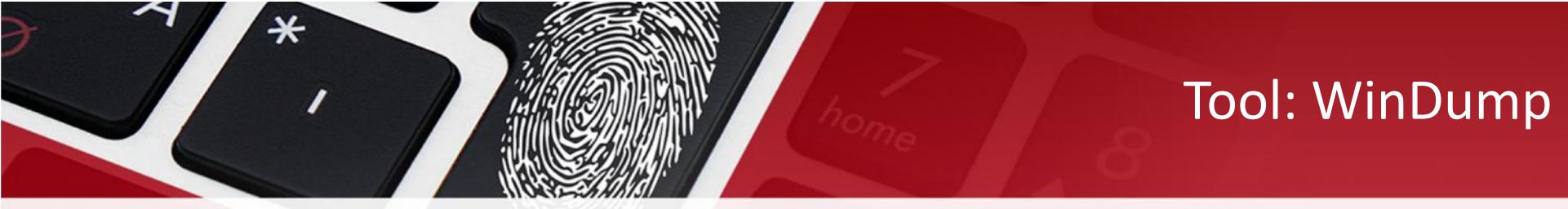
It supports following platforms:

- Sun OS 3.x or 4.x Solaris HP-UX IRIX Linux Ultrix and Digital UNIX BSD



Tcpdump: Screenshot

```
[root@ronin jgs]# tcpdump -q -c 20 -i eth0 arp
tcpdump: listening on eth0
00:05:24.654601 arp who-has ip68-110-147-15.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:24.658557 arp who-has ip68-110-147-16.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:24.672092 arp who-has ip68-110-147-18.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:24.680933 arp who-has ip68-110-145-55.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:24.681426 arp who-has ip68-110-147-20.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:24.791531 arp who-has ip68-110-145-63.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:24.928085 arp who-has ip68-110-147-232.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.064636 arp who-has ip68-110-147-26.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.082924 arp who-has ip68-110-145-67.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.151736 arp who-has ip68-110-147-28.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.170069 arp who-has ip68-110-147-29.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.173116 arp who-has ip68-110-147-30.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.178898 arp who-has ip68-110-145-72.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.181540 arp who-has ip68-110-147-32.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.234346 arp who-has ip68-110-145-77.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.255511 arp who-has ip68-110-147-33.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.590827 arp who-has ip68-110-145-83.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.601338 arp who-has ip68-110-145-84.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.634303 arp who-has ip68-110-147-39.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
00:05:25.640085 arp who-has ip68-110-147-40.hr.hr.cox.net tell ip68-110-144-1.hr.hr.cox.net
[root@ronin jgs]#
```



Tool: WinDump

WinDump is a version of tcpdump for Windows platform.

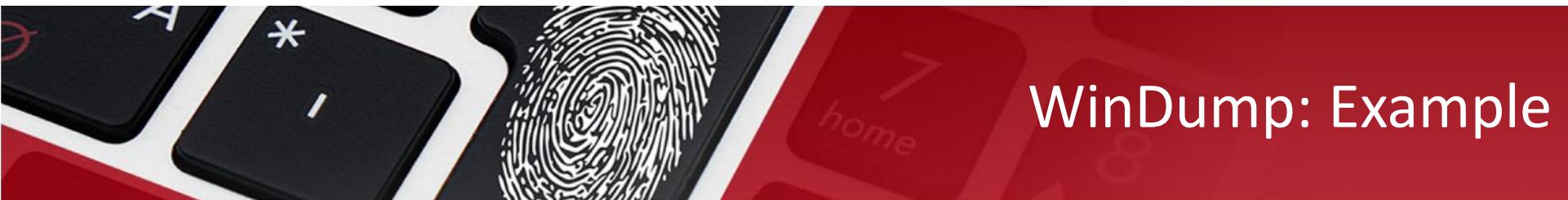
Command for saving the captured data packets using Windump as a sniffer:

- *C:\Windump -w filename.dmp*

The packets are stored in the C drive with the filename. The packets can be analyzed by using a notepad.

- *C:\Windump -w filename.dmp -s 65535*

The above command can be used to specify the size of the Ethernet packet to be captured.



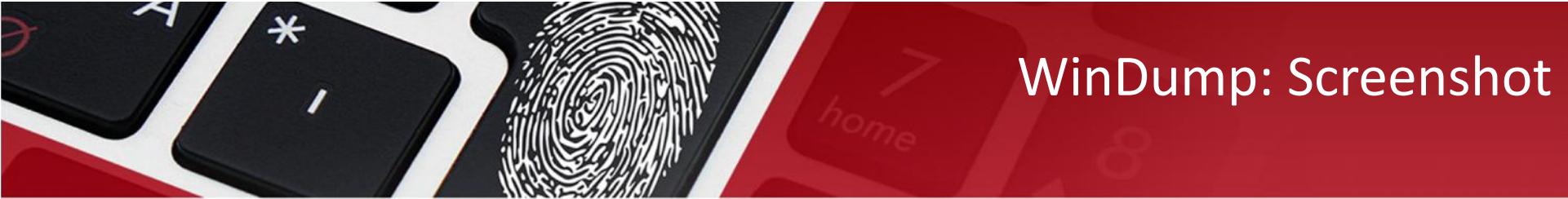
WinDump: Example

- Given below is the output obtained from WinDump:

```
20:50:00.037087 IP (tos 0x0, ttl 128, id 2572, len 46) 192.168.2.24.1036 >
64.12.24.42.5190: P [tcp sum ok] 157351:157357(6) ack 2475757024 win 8767 (DF)
```

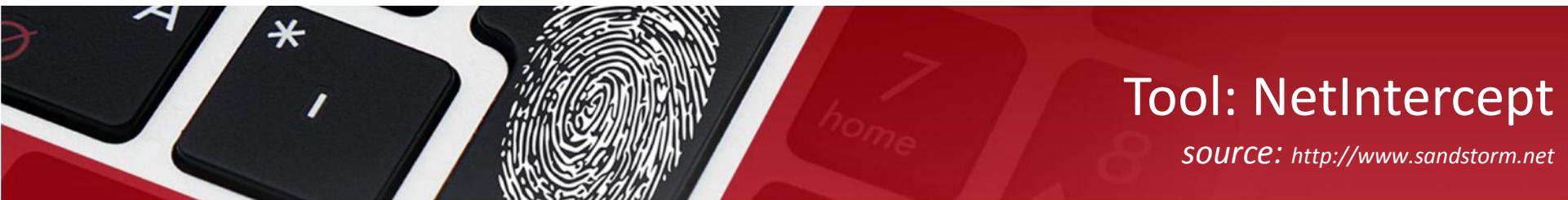
The above entry can be deciphered as:

Timestamp → 20:50:00.037087
IP [protocol header] → tos 0x0, ttl 128, id 2572, len 46
source IP:port → 192.168.2.24.1036
destination IP:port → 64.12.24.42.5190:
P [push flag] [tcp sum ok] → 157351:157357
[sequence numbers] (6) [bytes of data]
acknowledgement and sequence number → ack 2475757024
window size (DF) [don't fragment set] → win 8767



WinDump: Screenshot

```
C:\>windump -n -S -vv
windump: listening on \Device\NPF_{F036ABE8-53D7-4C7B-B2E4-082BEF4D72D8}
19:56:53.427131 IP <tos 0x88, ttl 106, id 58655, len 108> 68.193.110.230.5000 >
192.168.2.162.5000: udp 80
19:56:53.493683 IP <tos 0x88, ttl 106, id 58656, len 108> 68.193.110.230.5000 >
192.168.2.162.5000: udp 80
19:56:53.506094 IP <tos 0x88, ttl 43, id 46880, len 40> 64.4.26.250.80 > 192.168
.2.69.2446: . [tcp sum ok] 894239202:894239202<0> ack 4229117801 win 17520
19:56:53.506528 IP <tos 0x88, ttl 43, id 46881, len 510> 64.4.26.250.80 > 192.16
8.2.69.2446: P 894239202:894239672<470> ack 4229117801 win 17520
19:56:53.508241 IP <tos 0x88, ttl 43, id 46882, len 576> 64.4.26.250.80 > 192.16
8.2.69.2446: . 894239672:894240208<536> ack 4229117801 win 17520
19:56:53.508465 IP <tos 0x0, ttl 128, id 19205, len 40> 192.168.2.69.2446 > 64.4
.26.250.80: . [tcp sum ok] 4229117801:4229117801<0> ack 894240208 win 16514 <DF>
19:56:53.508602 IP <tos 0x88, ttl 43, id 46883, len 106> 64.4.26.250.80 > 192.16
8.2.69.2446: . 894240208:894240274<66> ack 4229117801 win 17520
19:56:53.527161 IP <tos 0x88, ttl 107, id 30218, len 1500> 68.58.11.235.2824 > 1
92.168.2.69.2443: . 47592813:47594273<1460> ack 4228398193 win 8359 <DF>
19:56:53.538245 IP <tos 0x88, ttl 106, id 58657, len 108> 68.193.110.230.5000 >
192.168.2.162.5000: udp 80
19:56:53.580115 IP <tos 0x88, ttl 243, id 39962, len 40> 202.87.41.115.80 > 192.
168.2.129.2549: F [tcp sum ok] 3461109112:3461109112<0> ack 6724698 win 8760 <DF>
```



Tool: NetIntercept

source: <http://www.sandstorm.net>



```
grep rsyslog
-6.617.x86_64
4
-v
-WITH:
support;
bug build, slow code;
actions supported;
actions supported;
station (slow code);

n for more information
```

NetIntercept captures and archives network traffic, so you can analyze problems as soon as they're detected.



```
netintercept -a
[...]
[netintercept] netintercept is a framework
[netintercept] for intercepting network traffic
[netintercept] and analyzing it.
[netintercept] It can be used to
[netintercept] monitor network traffic,
[netintercept] detect anomalies, and
[netintercept] reconstruct files transmitted or received over
[netintercept] the network.
[netintercept] It can also be used to
[netintercept] correlate user sessions and
[netintercept] reconstruct files transmitted or received over
[netintercept] the network, giving you immediate evidence of
[netintercept] misbehavior.
```

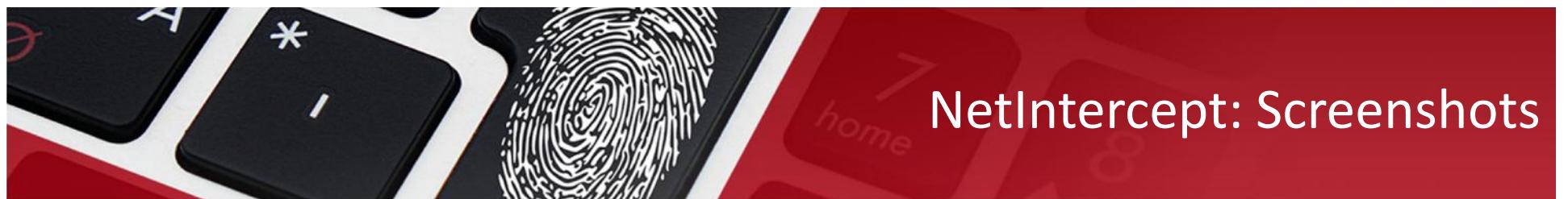
NetIntercept correlates user sessions and reconstructs files transmitted or received over the network, giving you immediate evidence of misbehavior.



```
[netintercept] netintercept -a > log.txt
[netintercept] netintercept is a framework
[netintercept] for intercepting network traffic
[netintercept] and analyzing it.
[netintercept] It can be used to
[netintercept] monitor network traffic,
[netintercept] detect anomalies, and
[netintercept] reconstruct files transmitted or received over
[netintercept] the network.
[netintercept] It can also be used to
[netintercept] correlate user sessions and
[netintercept] reconstruct files transmitted or received over
[netintercept] the network, giving you immediate evidence of
[netintercept] misbehavior.
```

Using NetIntercept, you can discover the security breaches, the points of regulatory non-compliance, and the network problems, and shift your focus from finding problems to fixing them.

NetIntercept: Screenshots



NetIntercept 3.2 - dbthursday30 - default_profile - cap1.sandstorm.net

File Forensics View Window Help

Open Rename Analyze Delete Print Import Export Find # Mask Time Save Revert Bookmark Schedule What's this?

Traffic Summary Forensics Alerts Views Reports Configuration

Search And Search And Search And Search And Search And Search And

SRC IP Address	DNS Name
4.0.30.19	
4.2.35.17	
4.2.143.75	
4.2.143.76	
4.17.99.95	
4.17.136.36	
4.17.158.10	
4.17.160.219	
4.17.247.11	
4.17.250.5	
4.21.3.3	
4.21.116.35	
4.22.130.11	
4.24.20.78	
4.24.20.158	
4.24.20.220	

DST IP Address	DNS Name
4.2.49.2	dnsauth1.sys.gtei
4.2.49.3	dnsauth2.sys.gtei
4.2.49.4	dnsauth3.sys.gtei
12.26.159.122	oak.pwccglobal.co
12.33.56.16	ns1.cdnow.com
12.33.56.17	ns2.cdnow.com
12.33.56.129	www.cdnow.com
12.33.56.130	gs.cdnow.com
12.33.56.131	ads.cdnow.com
12.43.230.67	co.dawwn.com
12.46.120.8	dns1.hasbro.com
12.46.120.10	www.hasbro.com
12.46.120.11	dns2.hasbro.com
12.46.120.19	
12.47.48.230	mail3.frk.ccm

Xfer Method
FTF Retrieve
FTF Store
FTF Store Unique
FTF Append
FTF List
FTF Name List
HTTP Get
HTTP Put
HTTP Post
HTTP Head
HTTP Delete
HTTP Trace
HTTP Connect
HTTP Option
TFTP Get
TFTP Put

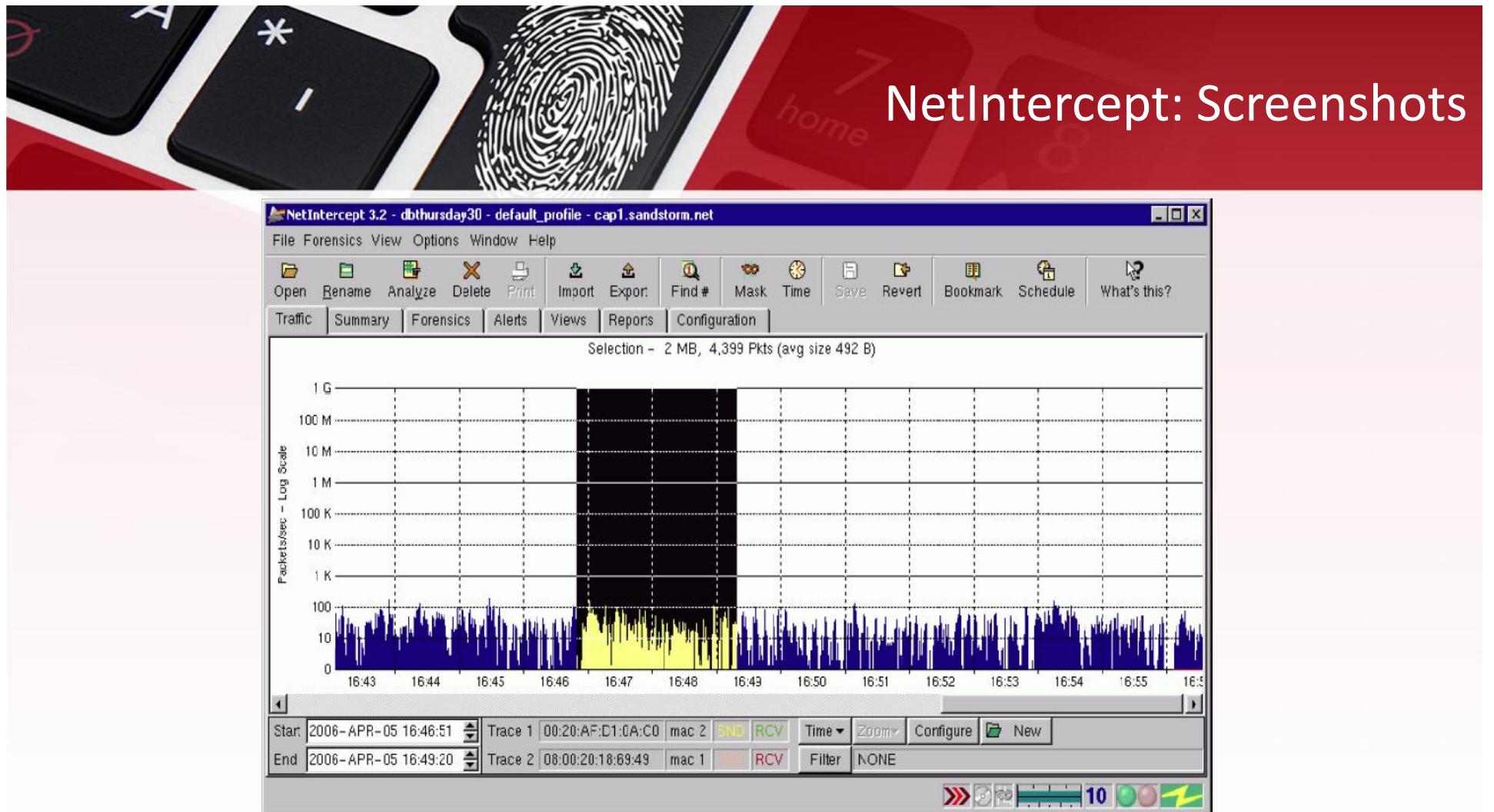
Content Type	Times
FINDPHRASE	0
FINDWORD	5
Finger	0
Flash	12
FTP	1
FTPdata	13
GIF	2,997
Gnutella	0
GZip	56
HTML	928
HTTP	12,158
ICMP	0
Ident	1,412
IMAP	0
IP	14,080

All Inv All Inv All Inv All Inv

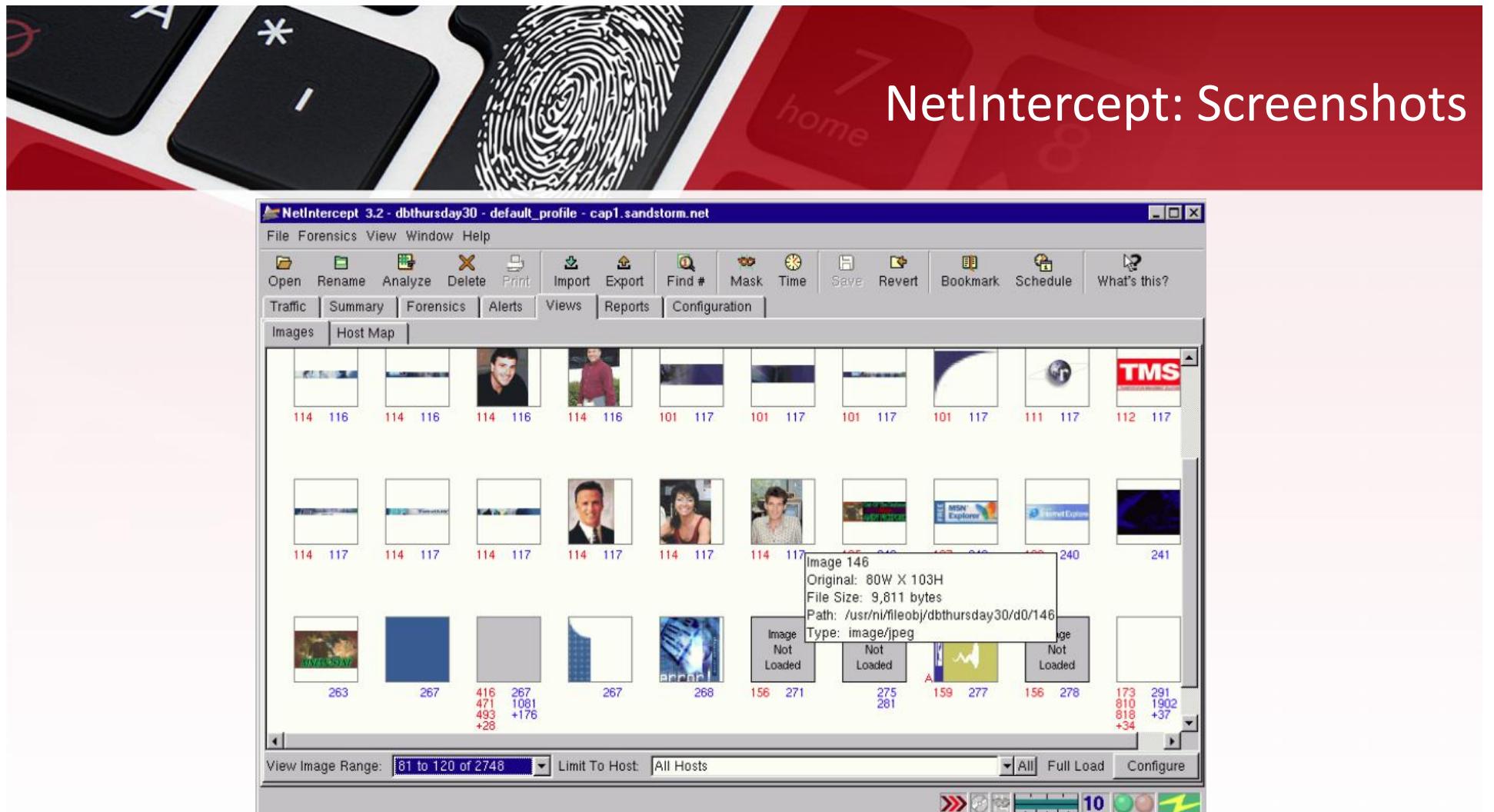
Query: Save Load Delete Columns: Clear All Balance Columns Criteria Min Surety: 0 Find Connections

10

NetIntercept: Screenshots



NetIntercept: Screenshots



NetIntercept: Screenshots

NetIntercept - Connection 5876 - dbthursday2_1

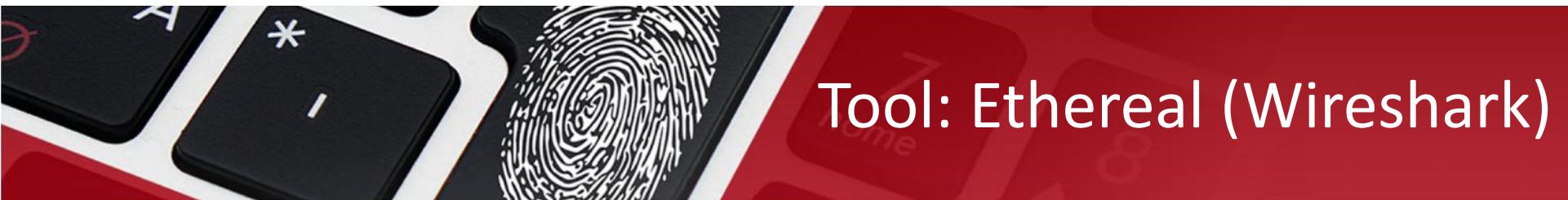
File Traffic Format Help

AS M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 | Search Repeat Image Hex Go To Packets ?

Traffic Session Summary Conclusions Criteria

Type	Surety	Xfer	Full Name	CONC ID	File Path
Ethernet	100		DIX Ethernet (RFC 894)	36645	
IP	100		Internet Protocol (RFC 791) traffic	36646	
TCP	100		Transmission Control Protocol (RFC 793)	36647	
HTTP	100	HTTP Get 200	HyperText Transfer Protocol	36648	
HTTP	100	HTTP Get 200	HyperText Transfer Protocol	36649	
MEDIA	100	HTTP Get 200	Central point for content with declared media...	36650	
HTML	100	HTTP Get 200	Hyper-Text Markup Language	36651	/usr/n/fileobj/dbthursday2_1/d2/2721
ASCII Type 100			Central point for ASCII (text) types.	36652	
PlainTxt	100	HTTP Get 200	Plain Text messages	36653	/usr/n/fileobj/dbthursday2_1/d2/2720
HTTP	100	HTTP Get 200	HyperText Transfer Protocol	36654	
MEDIA	100	HTTP Get 200	Central point for content with declared media...	36655	
JPEG	100	HTTP Get 200	JPEG File Interchange Format	36656	/usr/n/fileobj/dbthursday2_1/d2/2722
HTTP	100	HTTP Get 200	HyperText Transfer Protocol	36657	

00000000 FF D8 FF E0 00 10 4A 46 - 49 45 00 01 01 01 00 48 JFIF.
00000015 00 48 00 00 FF DB 00 43 - 00 05 03 04 04 04 03 05 . H. . . C. . .
00000032 04 04 04 05 05 06 07 - 06 08 07 07 07 07 0F 0B
00000048 08 09 0C 11 0F 12 12 11 - 0F 11 11 13 16 1C 17 13
00000064 14 1A 15 11 11 18 21 18 - 1A 1D 1D 1F 1F 13 17 !
00000080 22 24 22 1E 24 1C 1E 1F - 1E FF DB 00 43 01 05 05 '\$' \$. . . . C. . .
00000095 09 07 06 07 0E 08 08 0E - 1E 14 11 14 1E 1E 1E 1E
00000112 1E 1E 1E 1E 1E 1E 1E 1E - 1E 1E 1E 1E 1E 1E 1E 1E
00000128 1E 1E 1E 1E 1E 1E 1E 1E - 1E 1E 1E 1E 1E 1E 1E 1E
00000144 1E 1E 1E 1E 1E 1E 1E 1E - 1E 1E 1E 1E 1E 1E FF C0
00000150 00 11 00 00 00 00 01 70 01 01 00 00 00 11 01 00 11



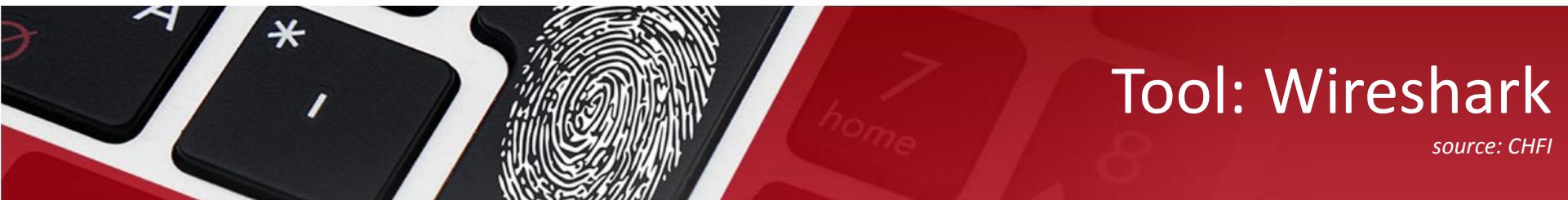
Tool: Ethereal (Wireshark)

Wireshark

Ethereal (Wireshark) is a network protocol analyzer for UNIX and Windows.

It allows the user to examine data from a live network or from a capture file on a disk.

The user can interactively browse the captured data, viewing the summary and detailed information of each packet captured.



Tool: Wireshark

source: CHFI

Capturing Live Data Packets Using Wireshark

1

Wireshark is a traffic capturing and sniffing tool
Wireshark uses Winpcap to capture packets, so it can only capture the packets on the networks supported by Winpcap

2

Captures live network traffic from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI networks

3

Captured files can be programmatically edited via command-line
A set of filters for customized data display can be refined using a display filter



Investigator

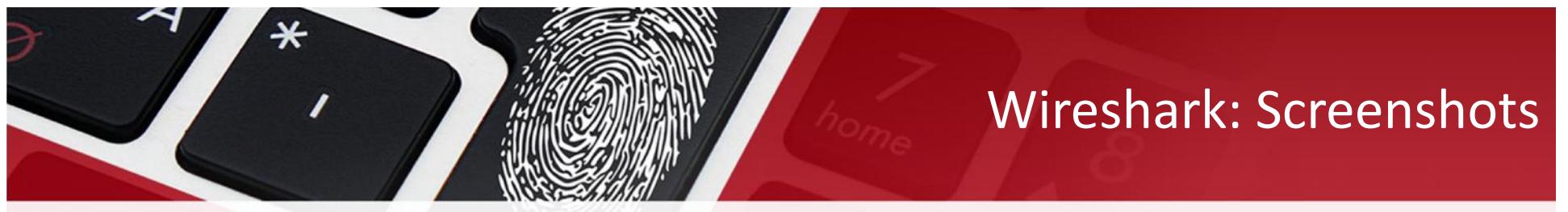


Wireshark Tool



Network

Wireshark: Screenshots



The screenshot shows the Wireshark interface displaying network traffic. The packet list pane shows 12 captured frames, mostly ICMP and LDAP requests. The details pane provides a breakdown of Frame 1, and the bytes pane shows the raw hex and ASCII data for the selected frame.

Packets List:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.0.0.5	10.0.0.1	LDAP	MsgId=14857 Search Request, Base DN=CN=Configur
2	0.000113	10.0.0.5	10.0.0.1	ICMP	Echo (ping) request
3	0.000176	10.0.0.1	10.0.0.5	ICMP	Echo (ping) reply
4	0.000632	10.0.0.1	10.0.0.5	LDAP	MsgId=14857 Search Entry, 1 result
5	0.202407	10.0.0.5	10.0.0.1	TCP	22862 > 3268 [ACK] Seq=188 Ack=169 Win=63564 Le
6	0.921485	10.0.0.5	10.0.0.1	LDAP	MsgId=62548 Search Request, Base DN=CN=Configur
7	0.921993	10.0.0.1	10.0.0.5	LDAP	MsgId=62548 search Entry, 1 result
8	1.076817	10.0.0.5	10.0.0.1	TCP	22863 > 3268 [ACK] Seq=189 Ack=171 win=63214 Le
9	2.154733	10.0.0.5	10.0.0.1	ICMP	Echo (ping) request
10	2.155209	10.0.0.1	10.0.0.5	ICMP	Echo (ping) reply
11	6.813562	10.0.0.5	10.0.0.1	LDAP	Invalid LDAP message (Can't parse sequence head
12	6.813658	10.0.0.5	10.0.0.1	LDAP	Invalid LDAP message (Can't parse sequence head

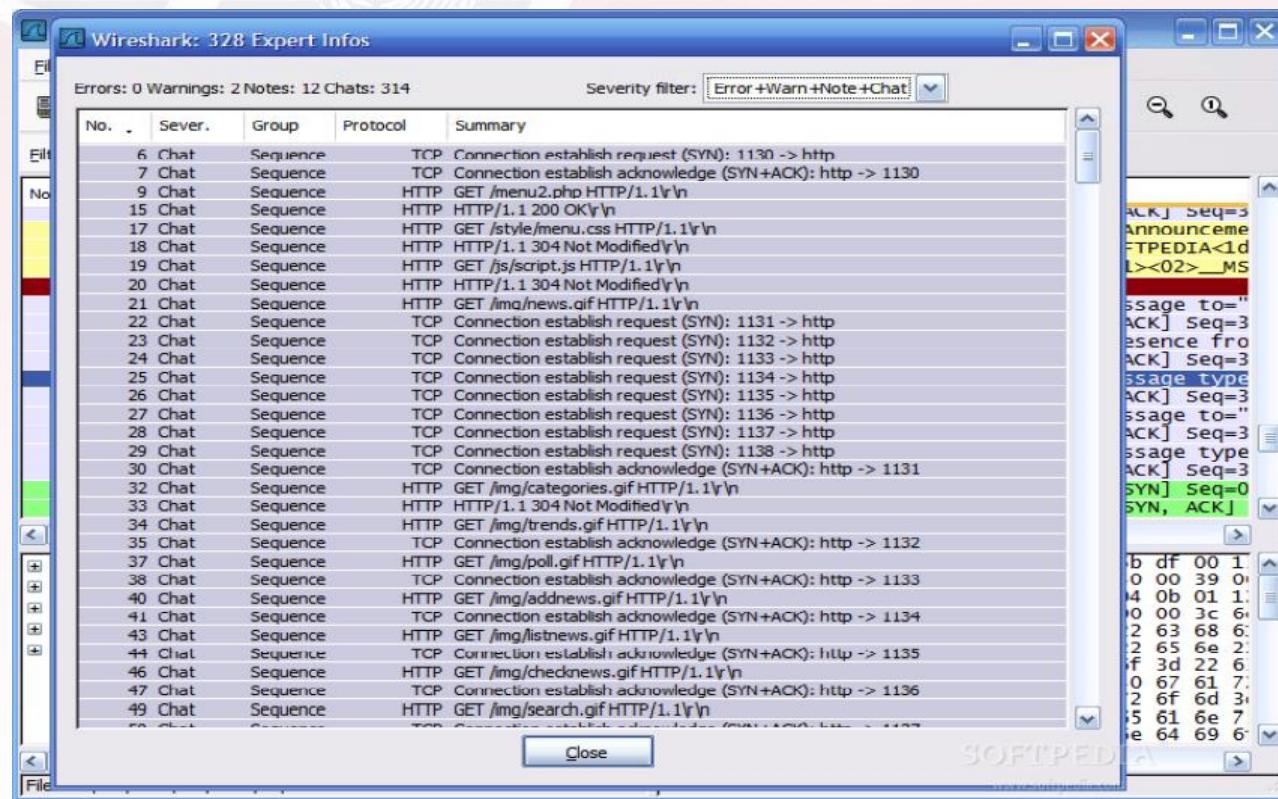
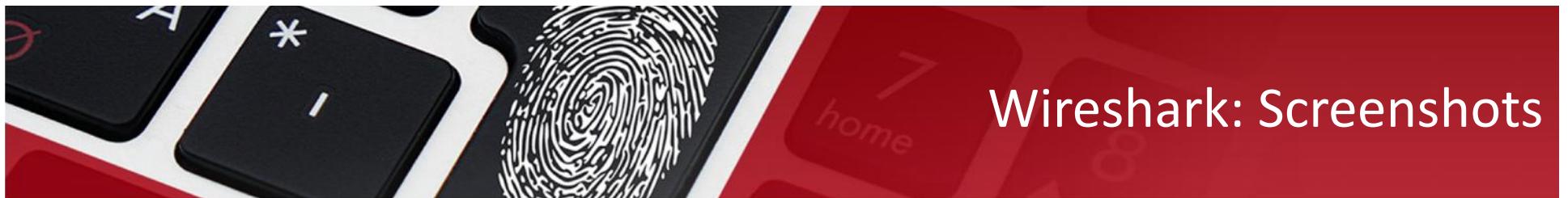
Frame 1 Details:

- Frame 1 (242 bytes on wire, 242 bytes captured)
- Ethernet II, Src: VMware_e6:45:e6 (00:0c:29:e6:45:e6), Dst: VMware_32:1a:5f (00:0c:29:32:1a:5f)
- Internet Protocol Version 4, Src: 10.0.0.5 (10.0.0.5), Dst: 10.0.0.1 (10.0.0.1)
- Transmission Control Protocol, Src Port: 22862 (22862), Dst Port: 3268 (3268), seq: 0, Ack: 169, Len: 242
- Lightweight Directory Access Protocol

Frame 1 Bytes:

Hex	Dec	ASCII
0000	00 0c 29 32 1a 5f 00 0c	29 e6 45 e6 08 00 45 00
0010	00 e4 c4 fa 40 00 80 06	..)2. ...) .E...E.
0020	00 01 59 4e 0c c4 8e 97@... !.....
0030	f8 f5 63 96 00 00 00 00	..YN....P.
0040	00 b8 60 81 b5 06 09 2a	..C....*
0050	00 0e e8 78 73 a7 10 00	.H.....
0060	00 10 21 f2 12 01 02 02	73.~... !.....x
0070	00 0e e8 78 73 a7 10 00	^

Wireshark: Screenshots



Tool: Wireshark

source: CHFI

Display Filters in Wireshark

Display filters are used to **change the view of packets** in the captured files

Example: Type the protocol in the filter box; arp, http, tcp, udp, dns

Display Filtering by Protocol



```
tcp.port==23  
ip.addr==192.168.1.100  
machine  
ip.addr==192.168.1.100 &&  
tcp.port=23
```

Monitoring the Specific Ports

Other Filters

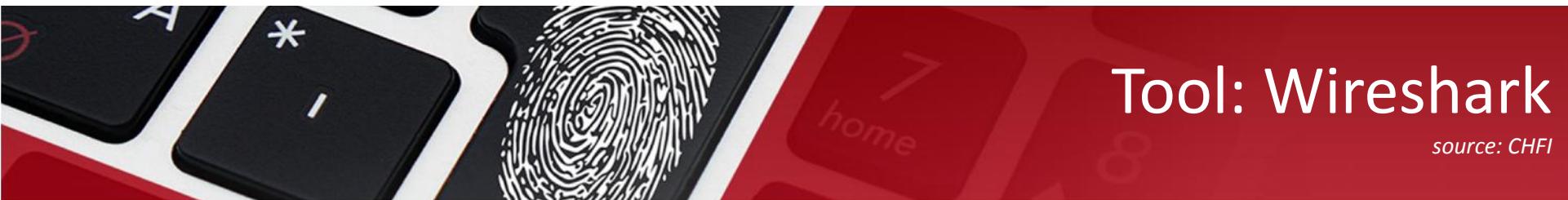
```
ip.dst == 10.0.1.50 && frame.pkt_len >  
400  
ip.addr == 10.0.1.12 && icmp &&  
frame.number > 15 && frame.number < 30  
ip.src==205.153.63.30 or  
ip.dst==205.153.63.30
```

Filtering by IP Address

```
ip.addr == 10.0.0.4
```

Filtering by Multiple IP Addresses

```
ip.addr == 10.0.0.4 or  
ip.addr == 10.0.0.5
```



Tool: Wireshark

source: CHFI

Additional Wireshark Filters

1

Displays all TCP resets

`tcp.flags.reset==1`

2

Displays all HTTP GET requests

`http.request`

3

Displays all TCP packets that contain the word "traffic"

`tcp contains traffic`

4

Sets a filter for the HEX values of 0x33 0x27 0x58 at any offset

`udp contains 33:27:58`

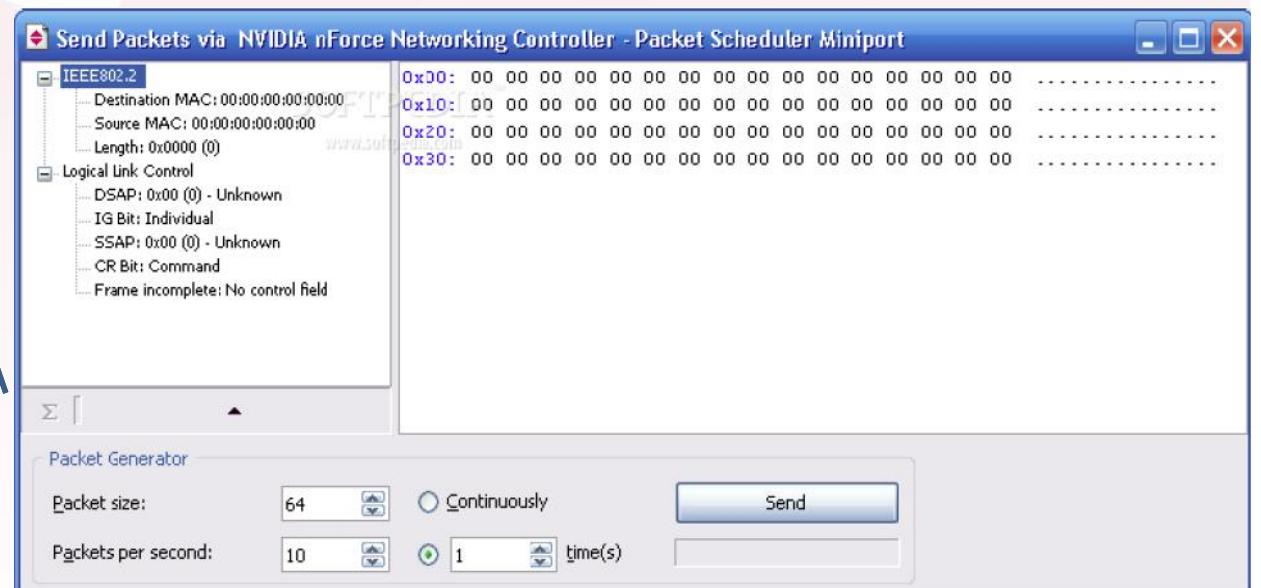
5

Displays all retransmissions in the trace

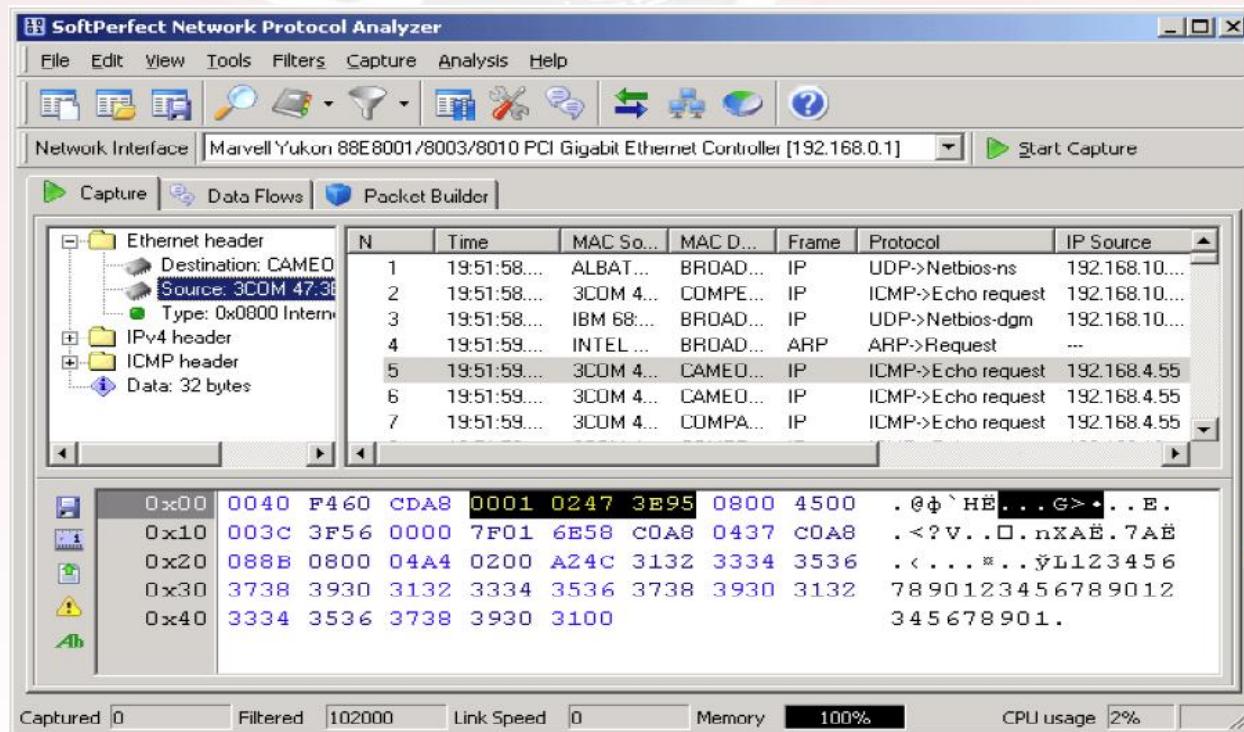
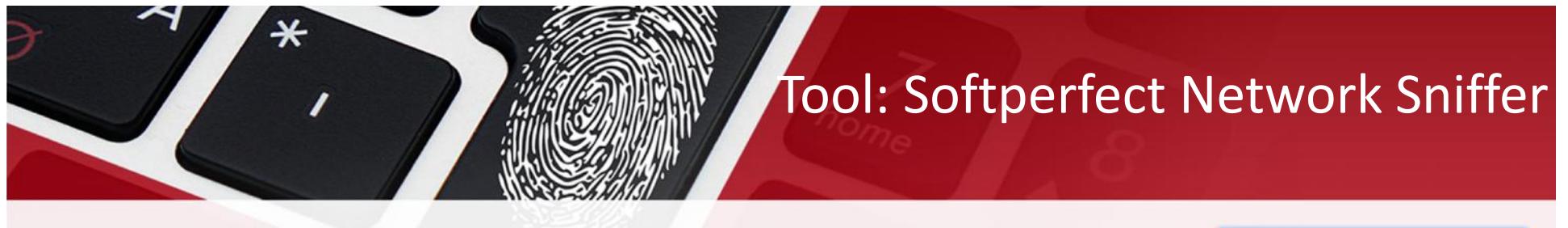
`tcp.analysis.retransmission`

Tool: CommView

CommView - monitor network activity capable of capturing and analyzing packets on any Ethernet network.



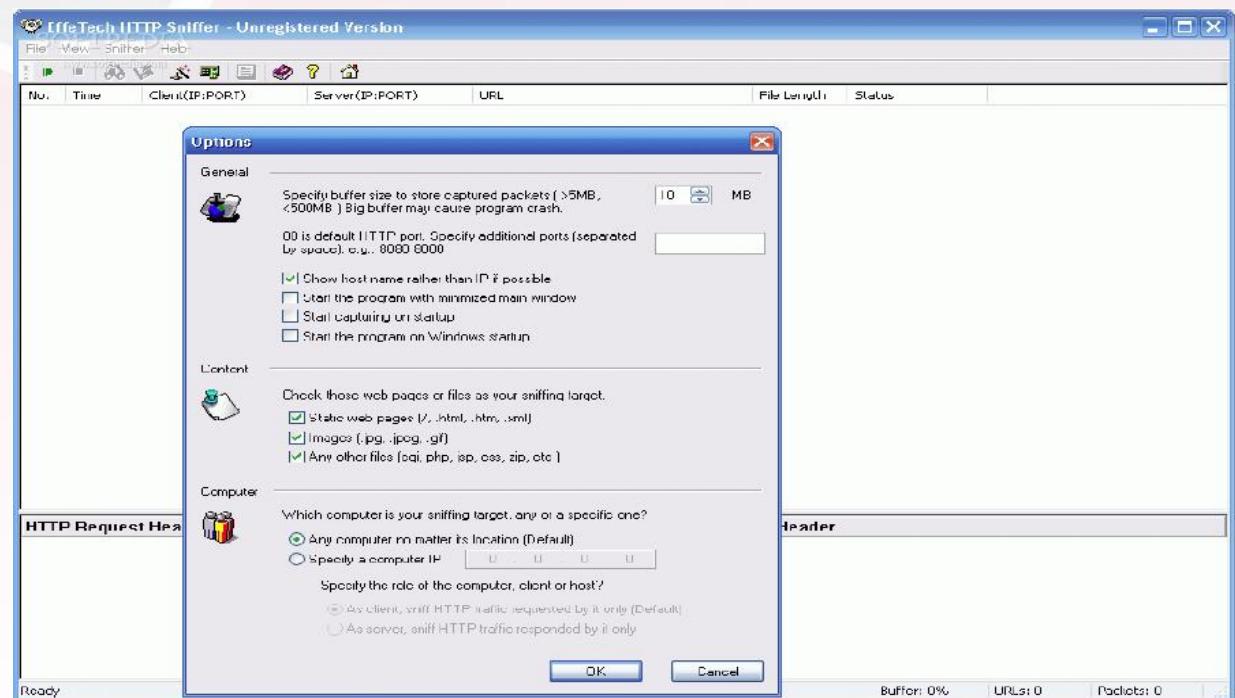
Tool: Softperfect Network Sniffer



Softperfect
Network Sniffer -
A network
protocol
analyzer or
sniffer.

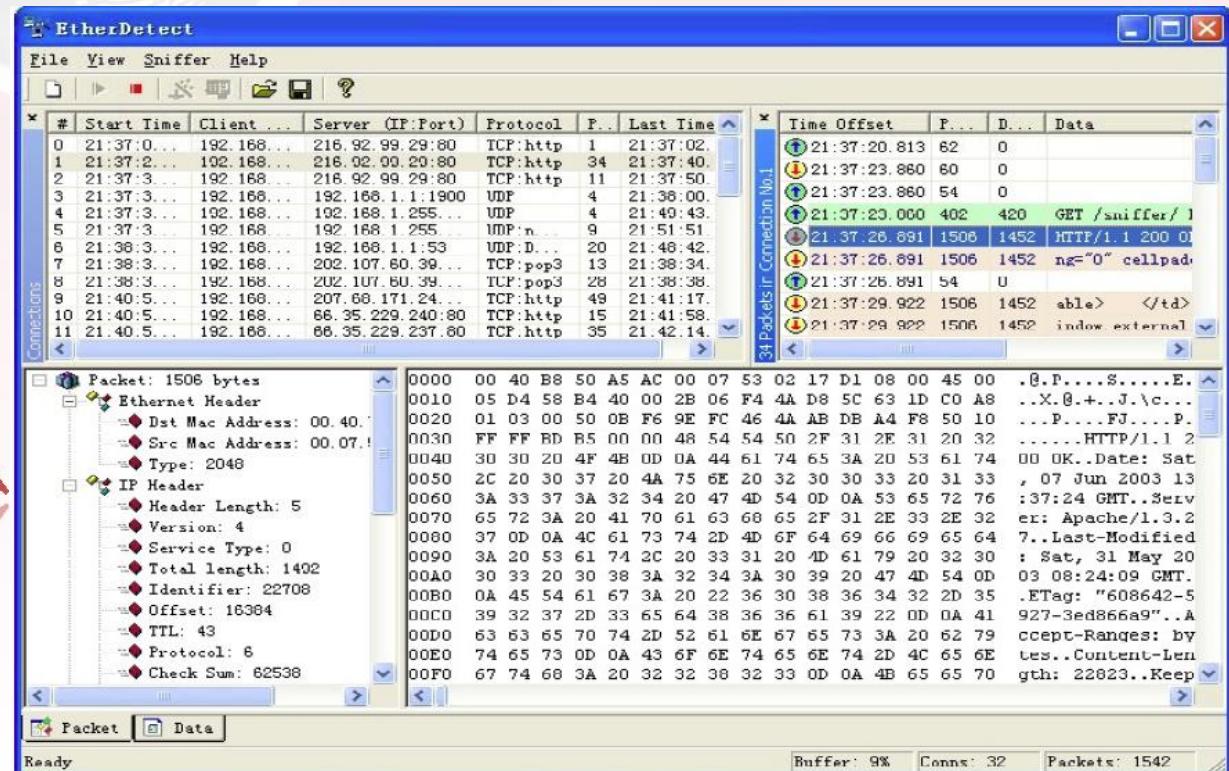
Tool: HTTP Sniffer

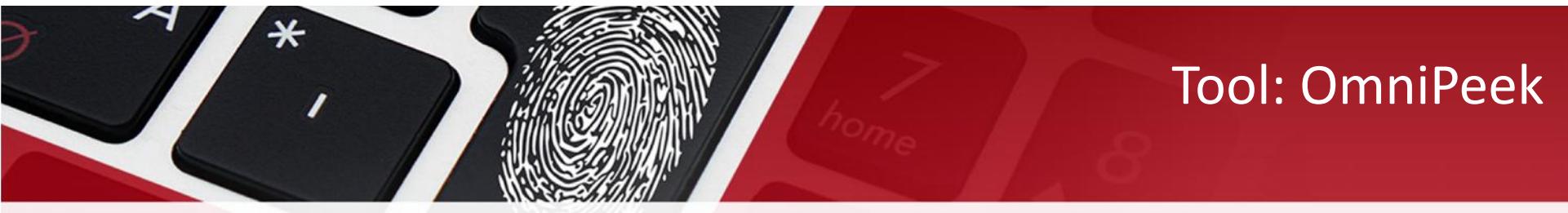
HttpDetect (EffeTech HTTP Sniffer) - HTTP sniffer, packet analyzer, content rebuilder, and http traffic monitor.



Tool: EtherDetect Packet Sniffer

EtherDetect Packet Sniffer
- Connection oriented
packet sniffer and protocol
analyzer.

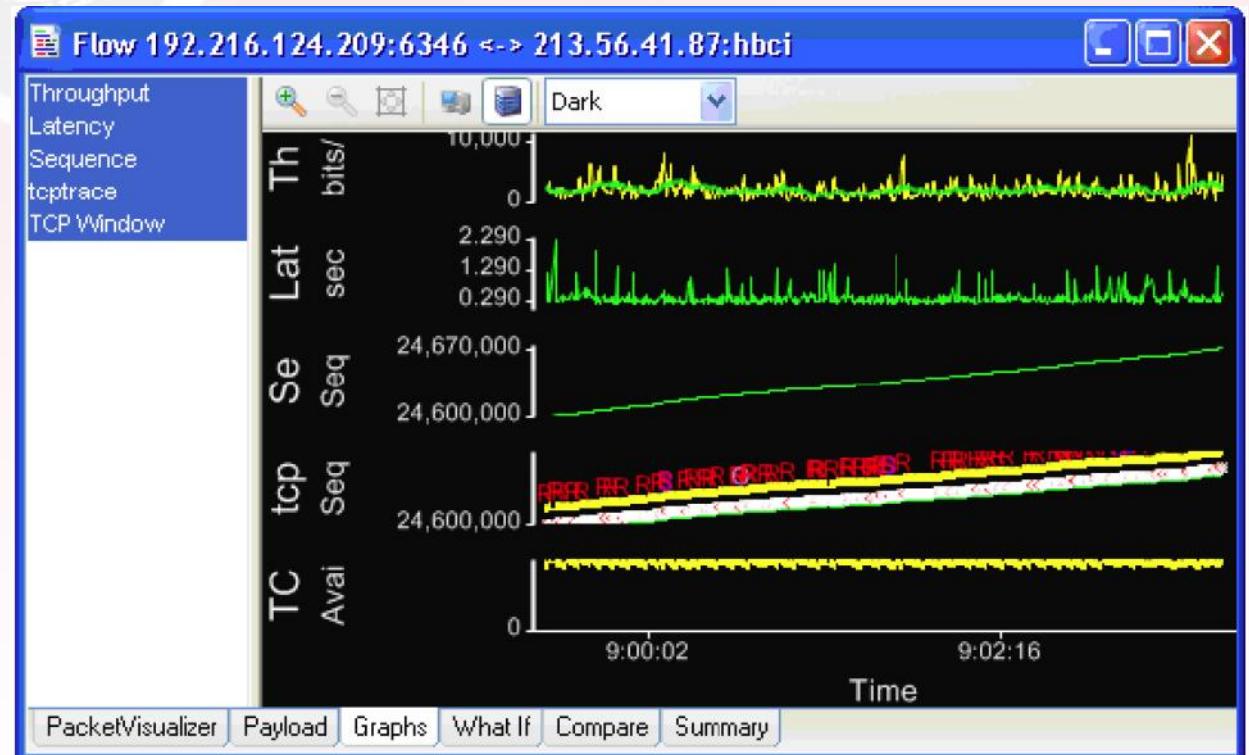




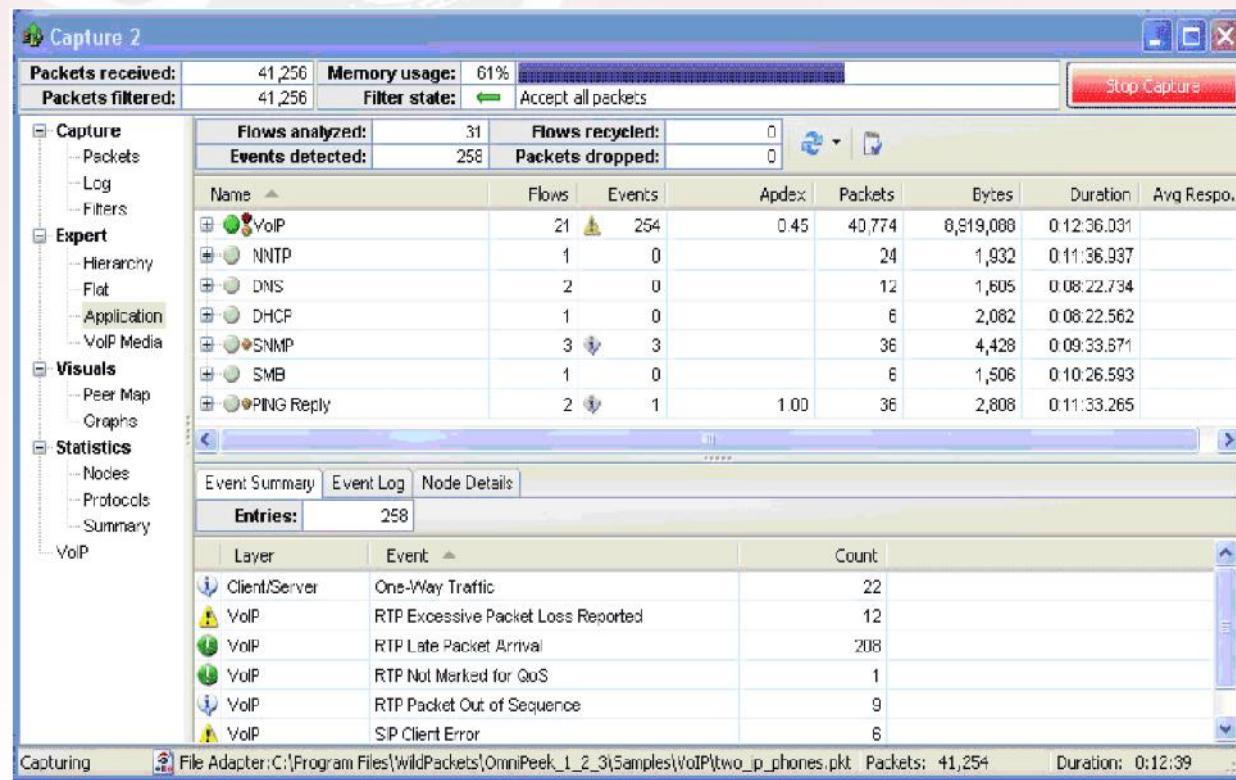
Tool: OmniPeek

OmniPeek Workgroup
is a full-featured,
stand-alone network
forensic analysis tool.

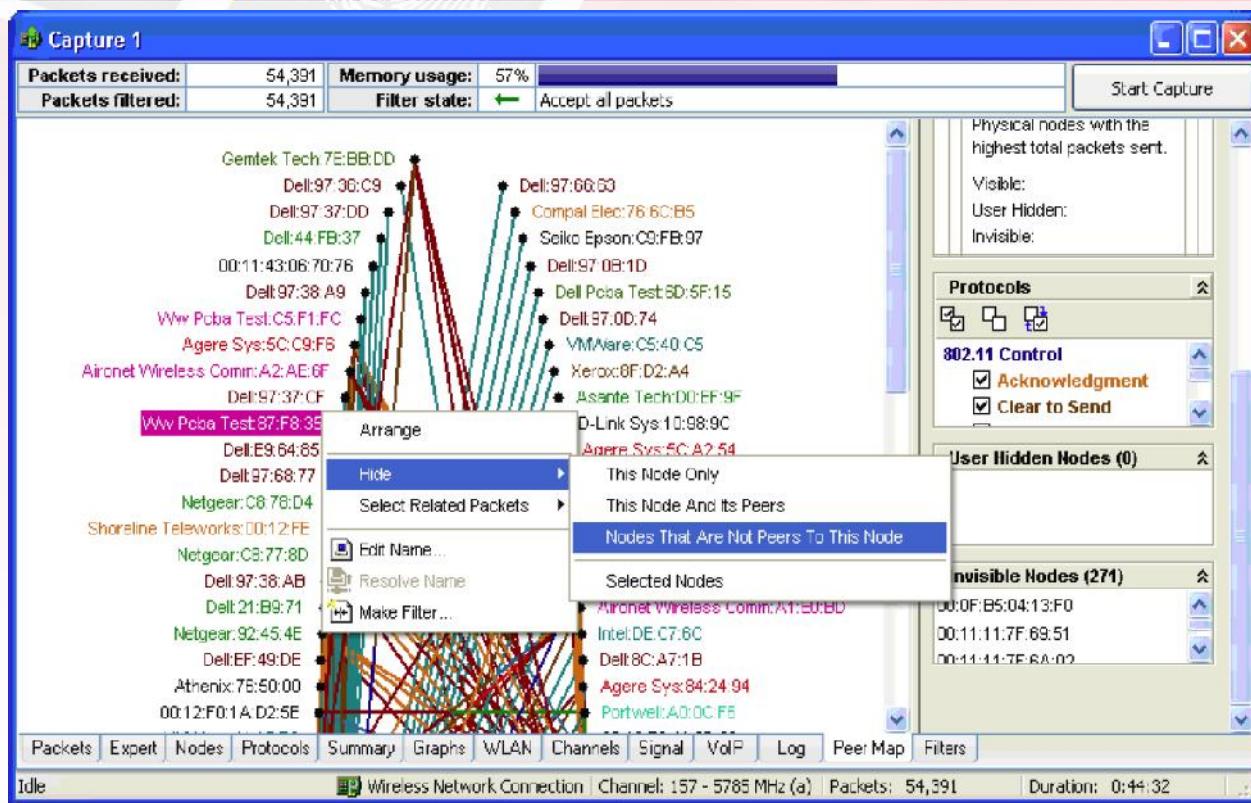
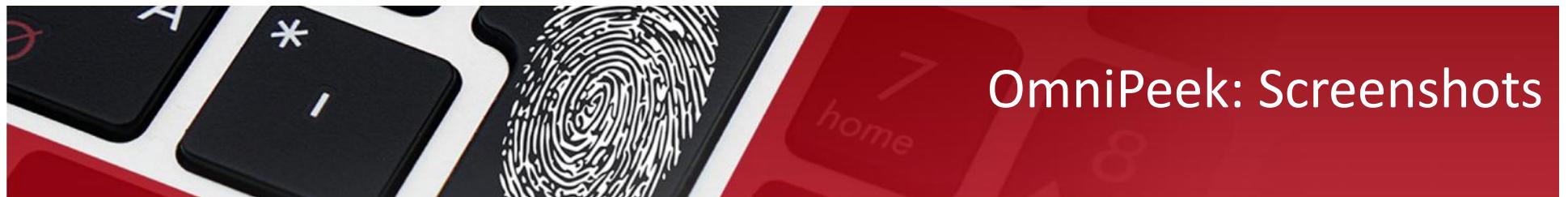
(<http://www.wildpackets.com>)

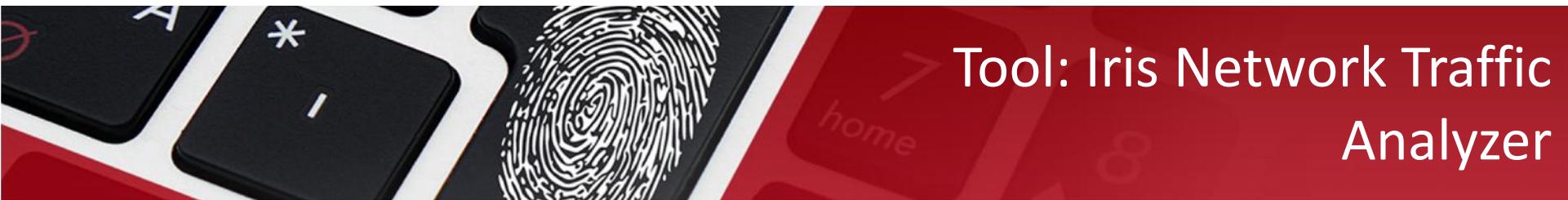


OmniPeek: Screenshots



OmniPeek: Screenshots





Tool: Iris Network Traffic Analyzer



The Iris Network Traffic Analyzer is a vulnerability forensics solution used for network traffic analysis and reporting.



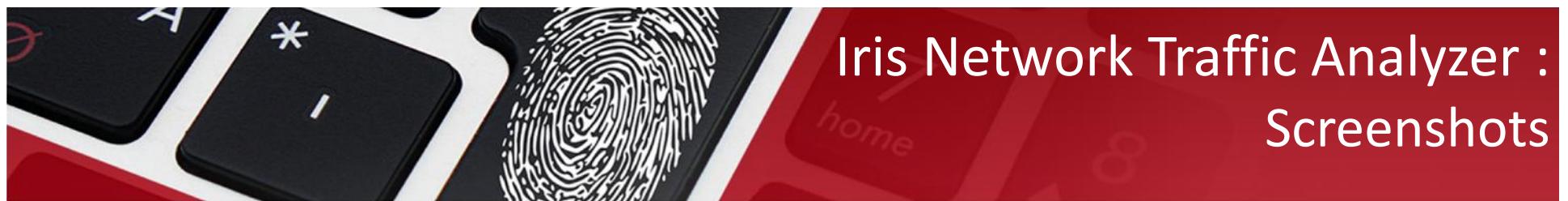
Iris captures network traffic and can automatically reassemble it to its native format.



Features of Iris Network Traffic Analyzer:

- Protocol Decoding
- Continuous Traffic Capture
- Create Custom Filters
- Complete Packet Reconstruction
- Powerful Sniffing and Spoofing Engine
- Screen Traffic by Key Criteria or Time Frame
- Alerting Capabilities
- Reconstruct TCP Sessions
- Packet Manipulation/Forging
- Log Foreign Connection Attempts
- Comprehensive Reporting
- Monitor Web-Based Email and Instant Messenger Services

Iris Network Traffic Analyzer : Screenshots



Iris Professional - Evaluation Version - 15 trial days remaining

File View Capture Decode Filters Tools Help

Decode Statistics Protocol Distribution Top Hosts Size Distribution Bandwidth Help Topics eEye Web Site Technical Support About Iris

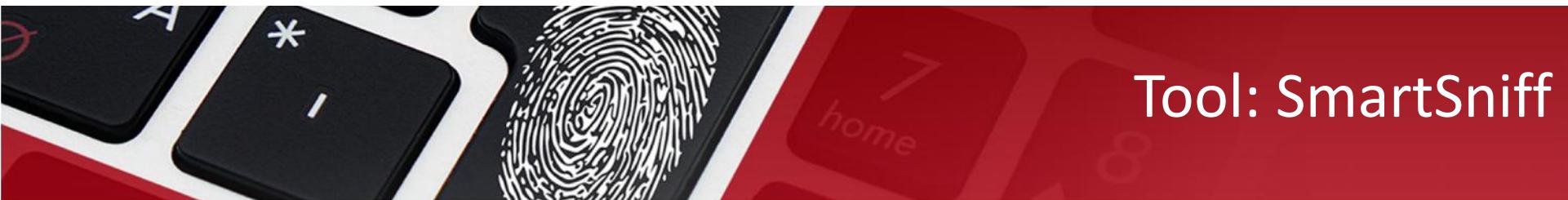
Capture

Packet Decoder

No.	Time (h:m:s:ms)	MAC source addr	MAC dest addr
3	11:15:52:163	00:F4:4E:5E:9...	00:0
4	11:15:52:183	00:0C:E5:0E:...	00:F
✓ 5	11:15:52:223	00:F4:4E:5E:9...	00:0
✓ 6	11:15:52:243	00:0C:E5:0E:...	00:F
✓ 7	11:15:52:243	00:F4:4E:5E:9...	00:0
✓ 8	11:15:52:313	00:F4:4E:5E:9...	00:0
✓ 9	11:15:52:323	00:0C:E5:0E:...	00:F
✓ 10	11:15:52:964	00:0C:E5:0E:...	00:F
✓ 11	11:15:52:964	00:0C:E5:0E:...	00:F
✓ 12	11:15:52:964	00:F4:4E:5E:9...	00:0
✓ 13	11:15:52:964	00:0C:E5:0E:...	00:F
✓ 14	11:15:52:964	00:0C:E5:0E:...	00:F
✓ 15	11:15:52:964	00:F4:4E:5E:9...	00:0
✓ 16	11:15:53:415	00:F4:4E:5E:9...	00:0
✓ 17	11:15:53:425	00:0C:E5:0E:...	00:F
✓ 18	11:15:54:046	00:0C:E5:0E:...	00:F
✓ 19	11:15:54:206	00:F4:4E:5E:9...	00:0
20	11:15:54:366	00:13:A9:14:...	01:0
21	11:15:58:863	00:13:A9:14:...	01:0

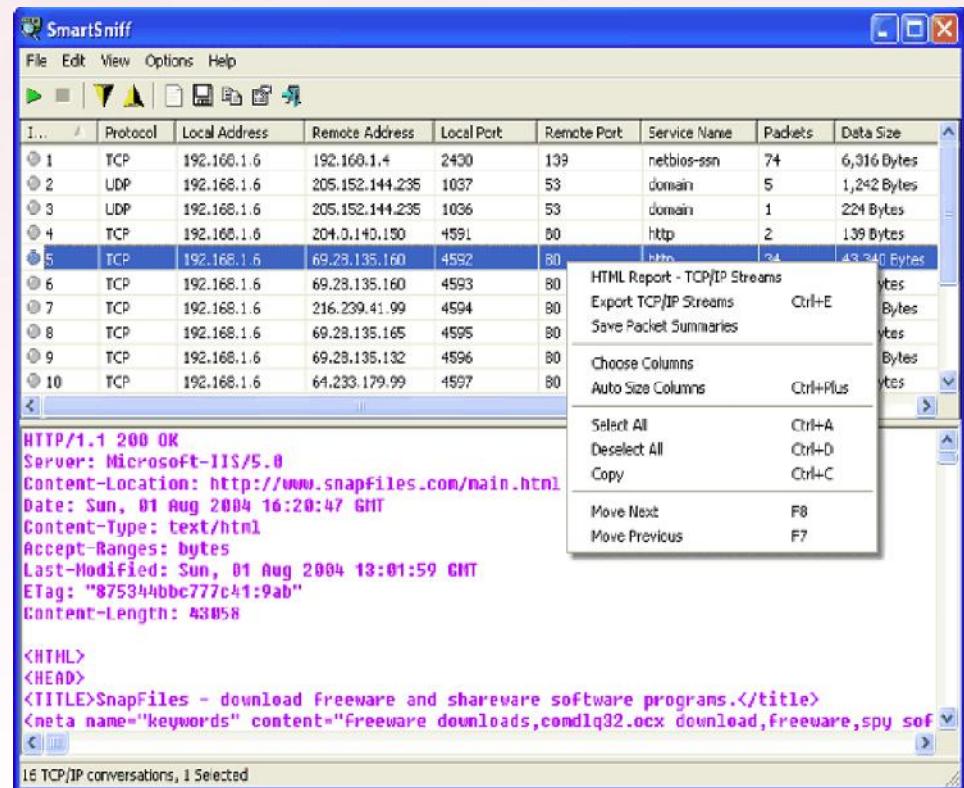
Did you know...
To start capturing in a new buffer, press Shift key while clicking Start Capture. Shortcut: Shift-A

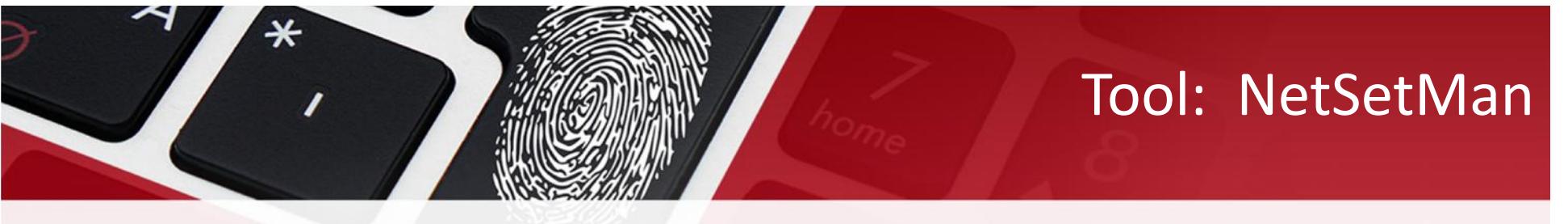
Filter: email.htm 21/2000



Tool: SmartSniff

- SmartSniff is a TCP/IP packet capture program that allows you to inspect network traffic that passes through your network adapter.
- It is a valuable tool to check what packets your computer is sending to the outside world.

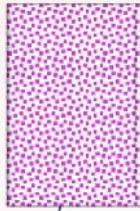




Tool: NetSetMan



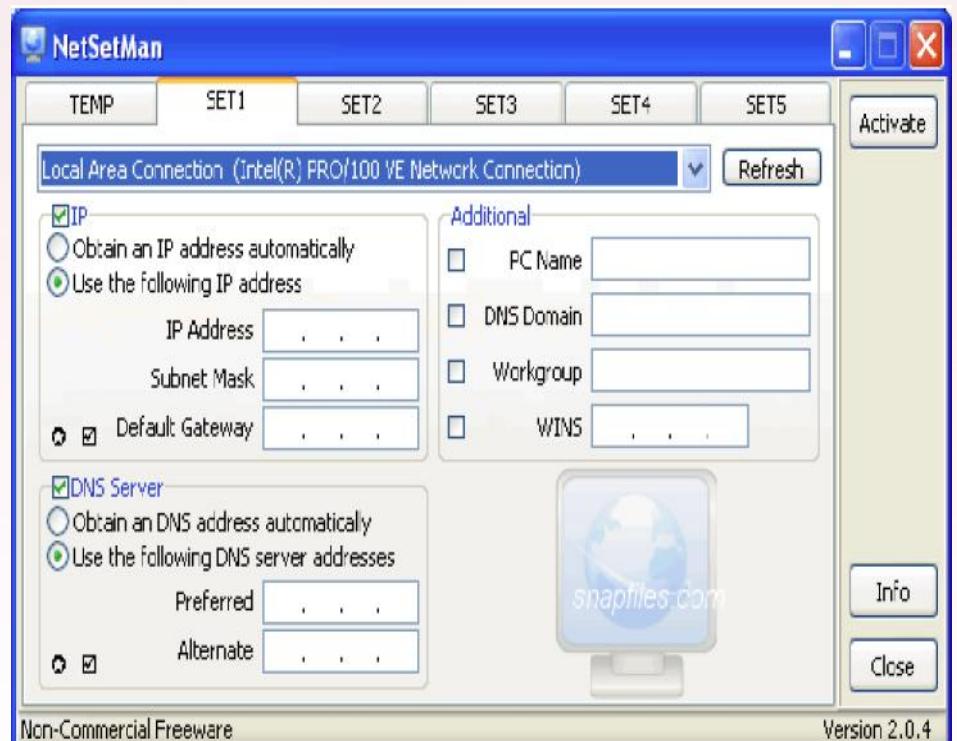
NetSetMan allows you to quickly switch between pre-configured network settings.

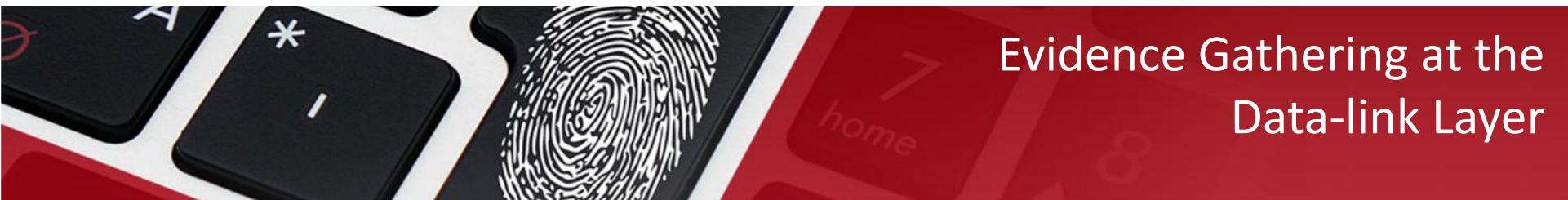


It is ideal for ethical hackers that have to connect to different networks all the time and need to update their network settings each time.

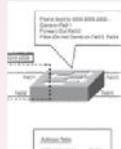


It allows you to create 6 profiles including IP address settings, Subnet Mask, Default Gateway, and DNS servers.





Evidence Gathering at the Data-link Layer



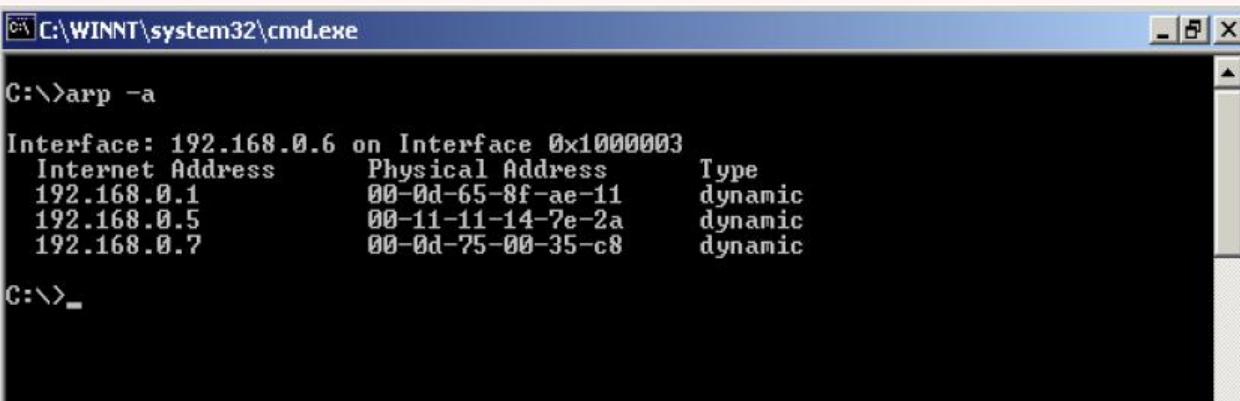
The MAC address , a part of the data-link layer, is associated with the hardware of a computer.



The ARP table of a router comes handy for investigating network attacks, as the table contains IP addresses associated with the respective MAC addresses.



ARP table can be seen using the given command in Windows OS
c:\arp -a



```
C:\>arp -a

Interface: 192.168.0.6 on Interface 0x1000003
  Internet Address      Physical Address      Type
  192.168.0.1            00-0d-65-8f-ae-11    dynamic
  192.168.0.5            00-11-11-14-7e-2a    dynamic
  192.168.0.7            00-0d-75-00-35-c8    dynamic

C:\>
```



Evidence Gathering at the Data-link Layer: ARP Table

source: CHFI

Evidence Gathering from ARP Table



```
C:\Windows\system32\cmd.exe
C:\>arp -a
Interface: 192.168.168.9 --- 0xb
Internet Address      Physical Address          Type
192.168.168.1          00-21-67-d0-d0-01    dynamic
192.168.168.2          00-21-67-d0-d0-02    dynamic
192.168.168.3          00-21-67-d0-d0-03    dynamic
192.168.168.4          00-0c-95-00-00-04    dynamic
224.0.0.1               ff-ff-ff-ff-ff-ff    static
224.0.0.2               ff-ff-ff-ff-ff-ff    static
239.0.0.1               ff-ff-ff-ff-ff-ff    static
255.255.255.255         ff-ff-ff-ff-ff-ff    static
C:\>
```

ARP table can be accessed using the
c:\arp -a command in Windows OS

MAC address:
A part of the data-link layer is associated with the system hardware



The ARP table of a router comes in handy for investigating network attacks, as the table contains IP addresses associated with the respective MAC addresses





Evidence Gathering at the Data-link Layer: DHCP Database

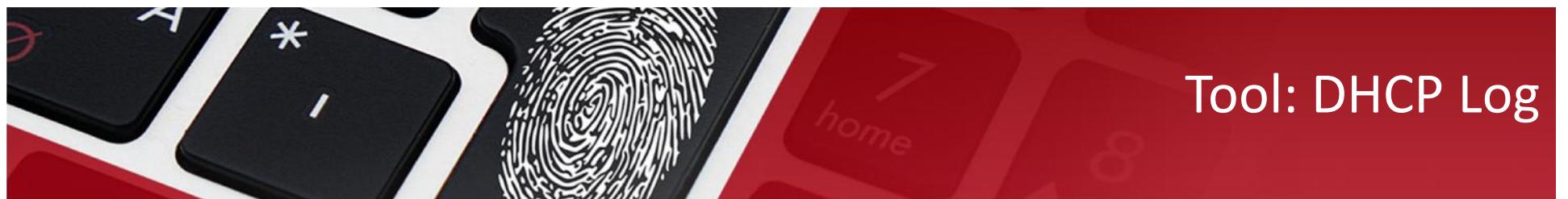
The DHCP database also provides as a means for determining the MAC addresses associated with the computer in custody.

The DHCP server maintains a list of recent queries, along with the MAC address and IP address.

Documentation of ARP table is done by the following:

- Photographing the computer screen.
- Screenshot of the table is taken and saved on the disk.
- The HyperTerminal logging facility is used.

Tool: DHCP Log



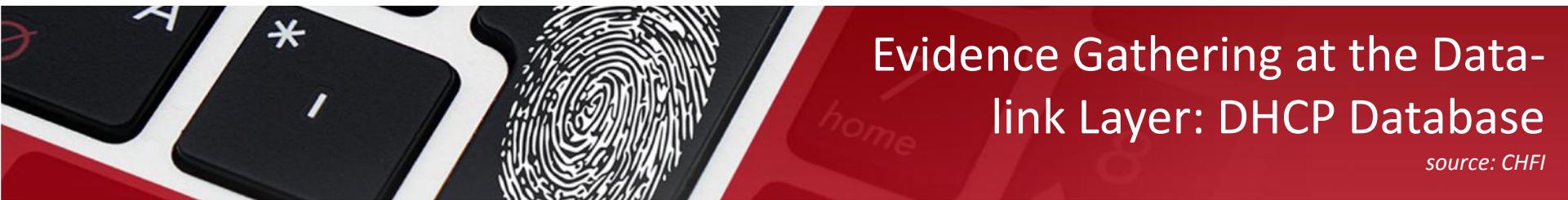
The image shows a user interface for a network device configuration tool. On the left is a vertical menu bar with blue buttons, each containing a white label. The buttons are: Home, Configure, Statistics, ADSL, ATM, Ethernet, IP, LAN (which is highlighted with a red border), Logs, Diagnostics, Remote Access, Update Modem, Reset Modem, Modem Self Test, and Basic Mode.

The main content area has a light blue header bar labeled "LAN Statistics". Below it is a table with the following data:

Modem IP Address	192.168.1.254
DHCP NetMask	255.255.255.0
DHCP Start Address	192.168.1.1
DHCP End Address	192.168.1.253
DHCP Server Status	ON
DHCP Server	Private

Below this is another header bar labeled "Devices on LAN". Underneath it is a table with the following data:

IP Address	MAC Address	Name	Status
192.168.1.97	00:XX:0f:XX:95:6e	*	Active



Evidence Gathering at the Data-link Layer: DHCP Database

source: CHFI

Evidence Gathering at the Data-Link Layer: DHCP Database

- The DHCP database determines the **MAC addresses** associated with the computer in custody
- The DHCP server **maintains a list of recent queries** along with the MAC address and IP address

Documentation of the ARP table is done by:

- **Photographing** the computer screen
- Taking the **screenshot** of the table and saving it on a disk
- Using the **HyperTerminal** logging facility



New leases: 192.168.0.9 00-21-00-02-89-9A				
COMMAND	IP	MAC	Entered	Expir.
lease	192.168.1.5	00-15-AP-7B-C9-99	07/03/2009 7:46:00 AM	06/14/2009 7:46:00 PM
lease-00999999	192.168.0.5	00-15-BE-60-51-E1	07/03/2009 7:46:00 AM	06/13/2009 12:00:00 AM
lease	192.168.0.3	00-15-00-00-00-9A	07/03/2009 7:47:57 AM	06/14/2009 7:47:57 AM
lease-PC	192.168.7.39	00-15-P3-PD-89-1B	07/03/2009 12:29:57 AM	06/13/2009 12:29:57 AM
lease-PC	192.168.1.2	00-13-CE-7E-A3-6F	07/03/2009 3:01:51 AM	06/14/2009 2:01:51 AM
lease	192.168.0.75	00-15-0E-94-64-82	07/03/2009 1:52:00 PM	06/14/2009 1:52:00 PM
lease	19.200.3.111	00-1P-3A-1A-09-5A	07/03/2009 7:44:00 PM	06/13/2009 7:44:00 PM
lease	192.168.0.29	00-15-00-00-00-9A	07/03/2009 7:44:00 PM	06/13/2009 12:00:00 AM
lease	192.168.0.1	00-15-00-00-00-9A	07/03/2009 7:45:10 AM	06/13/2009 7:45:10 AM
lease-192.168.0.1	192.168.0.1	00-15-00-00-00-9A	07/03/2009 7:45:10 AM	06/13/2009 7:45:10 AM
lease-0326e90	192.168.0.7	00-21-00-00-00-9A	07/03/2009 11:55:00 AM	06/13/2009 11:55:00 AM
lease	192.168.0.2	00-16-45-C2-B3-4C	07/03/2009 3:29:00 AM	06/14/2009 3:29:00 AM
lease-0326e90	192.168.0.31	00-14-45-27-70-0F	07/03/2009 1:21:07 PM	06/13/2009 1:21:07 PM
lease	192.168.0.33	00-15-44-54-04-66	07/03/2009 1:55:05 PM	06/13/2009 1:55:15 PM



Overview of Network and Transport Layer of the OSI Model

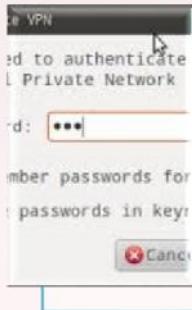
Network layer:

- It is responsible for sending information from the source to a destined address across various links.
- It adds logical addresses of the sender and receiver to the header of the data packet.

Transport layer:

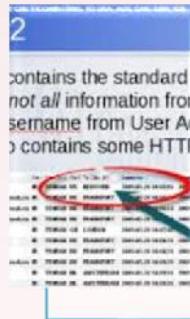
- The transport layer ensures that the whole message sent by the source has reached its destination and is in order.
- It oversees the error control and flow control in between the transmission.

Evidence Gathering at the Network and Transport Layer



Authentication logs:

- Shows accounts related to a particular event.
- The IP address of the authenticated user gets stored in this log file.



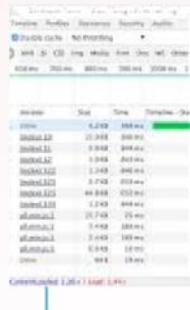
Operating System logs:

- It maintains log of events such as errors, system reboot, shutdown, security policy changes, user, and group management.
- But before enabling logging bear in mind: what to log otherwise, it can result in over-collection of data, making it difficult to trace the critical event.



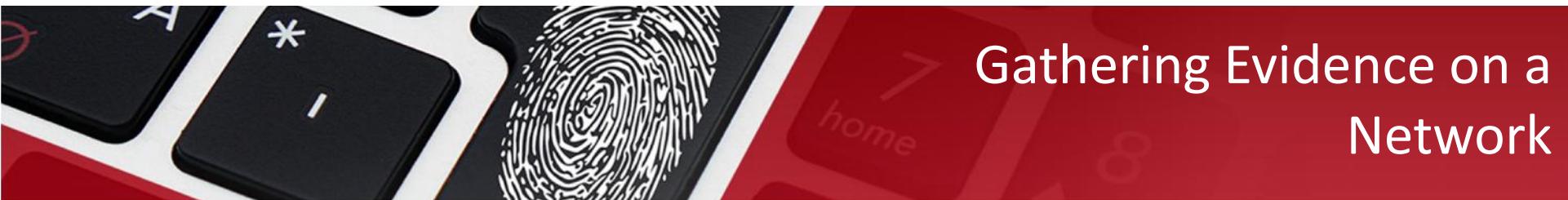
Application logs:

- Application logging is meant for the storage of auditing information, which includes information produced by application activity.
- Web server logs help identify the system which was used as a means to commit the crime.
- Only administrator has the privilege to access these log files.



Network device logs:

- Network devices such as router and firewalls are configured to send a copy of their logs to remote server, as the memory for these devices is low.
- The logs from network devices can be used as evidence for particular investigation on that network.



Gathering Evidence on a Network

Evidence on a Network

IDS can be configured to capture network traffic when an alert is generated.

Examination results of networking devices such as routers and firewalls, can be recorded through a serial cable using a Windows HyperTerminal program or by Script on UNIX.

If the amount of information to be captured is huge, then record the onscreen event using a video camera or a relative software program.

Gathering Evidence on a Network: Snort Intrusion Detection System



Snort is a versatile, lightweight, and very useful intrusion detection system.



Snort logs packets in either tcpdump binary format or in Snort's decoded ASCII format to logging directories that are named based on the IP address of the foreign host.

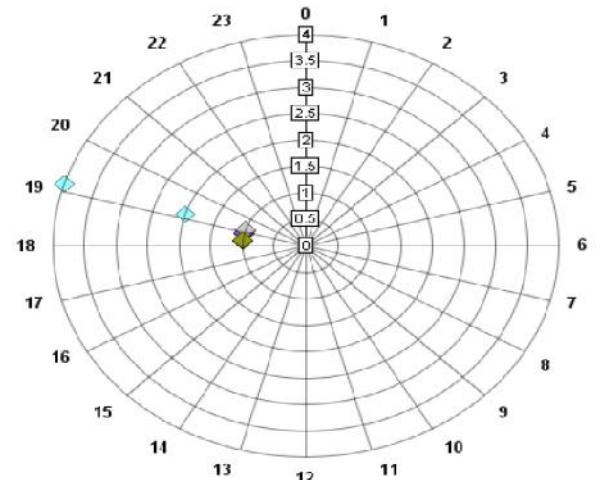


Plug-ins allow the detection and reporting subsystems to be extended.



Available plug-ins include database logging, small fragment detection, portscan detection, and HTTP URI normalization.

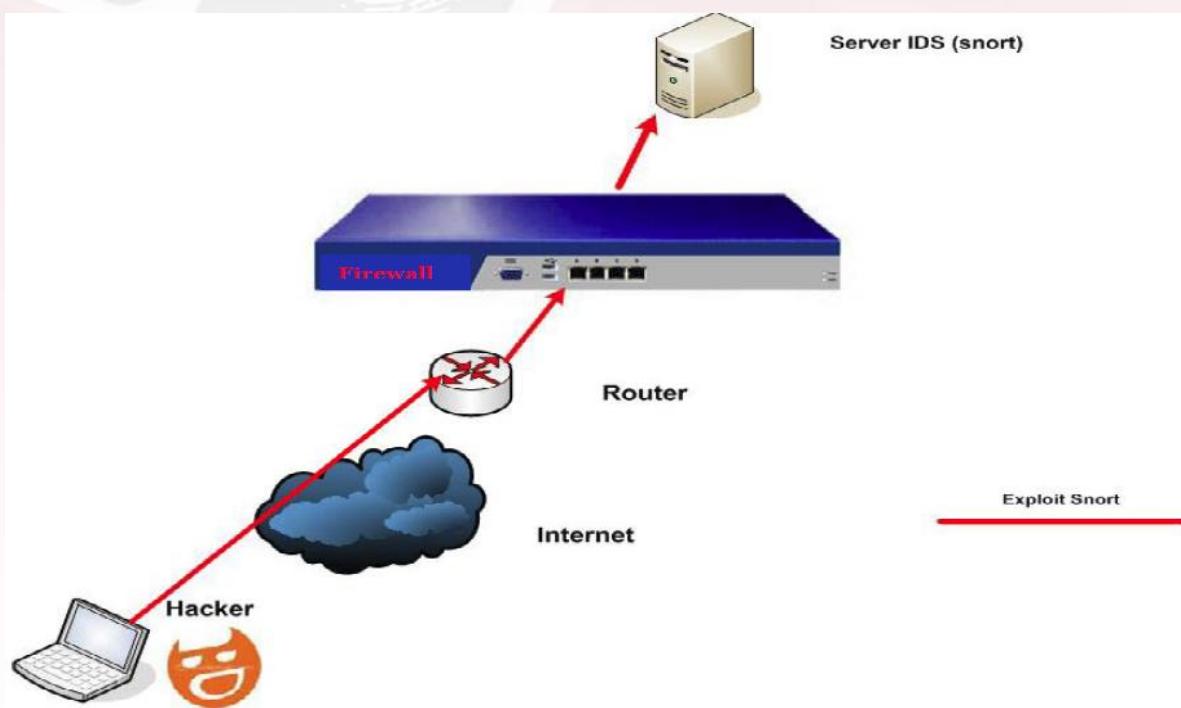
Attacks, Scans, Probes etc Daily from 216.254.95.40

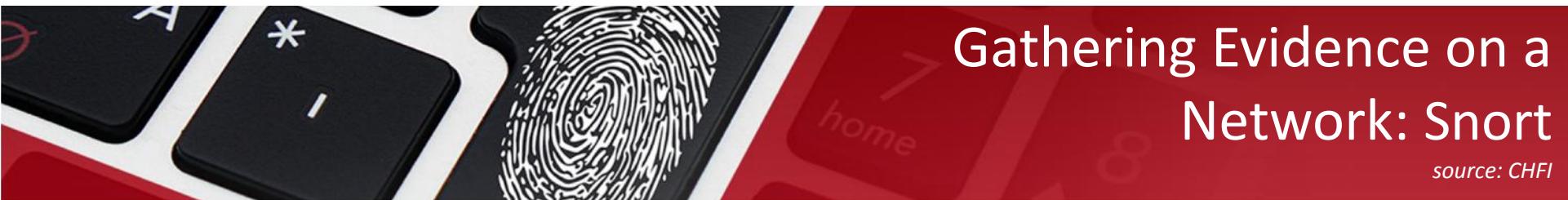


- ◆ ICMP PING NMAP
- ◆ BAD-TRAFFIC top port 0 traffic
- ◆ BAD-TRAFFIC udp port 0 traffic
- ◆ (portscan) TCP Portscan
- ◆ (portscan) SNMP trap top
- ◆ (portscan) Open Port
- ◆ SNMP AgentXop request
- ◆ SNMP request top



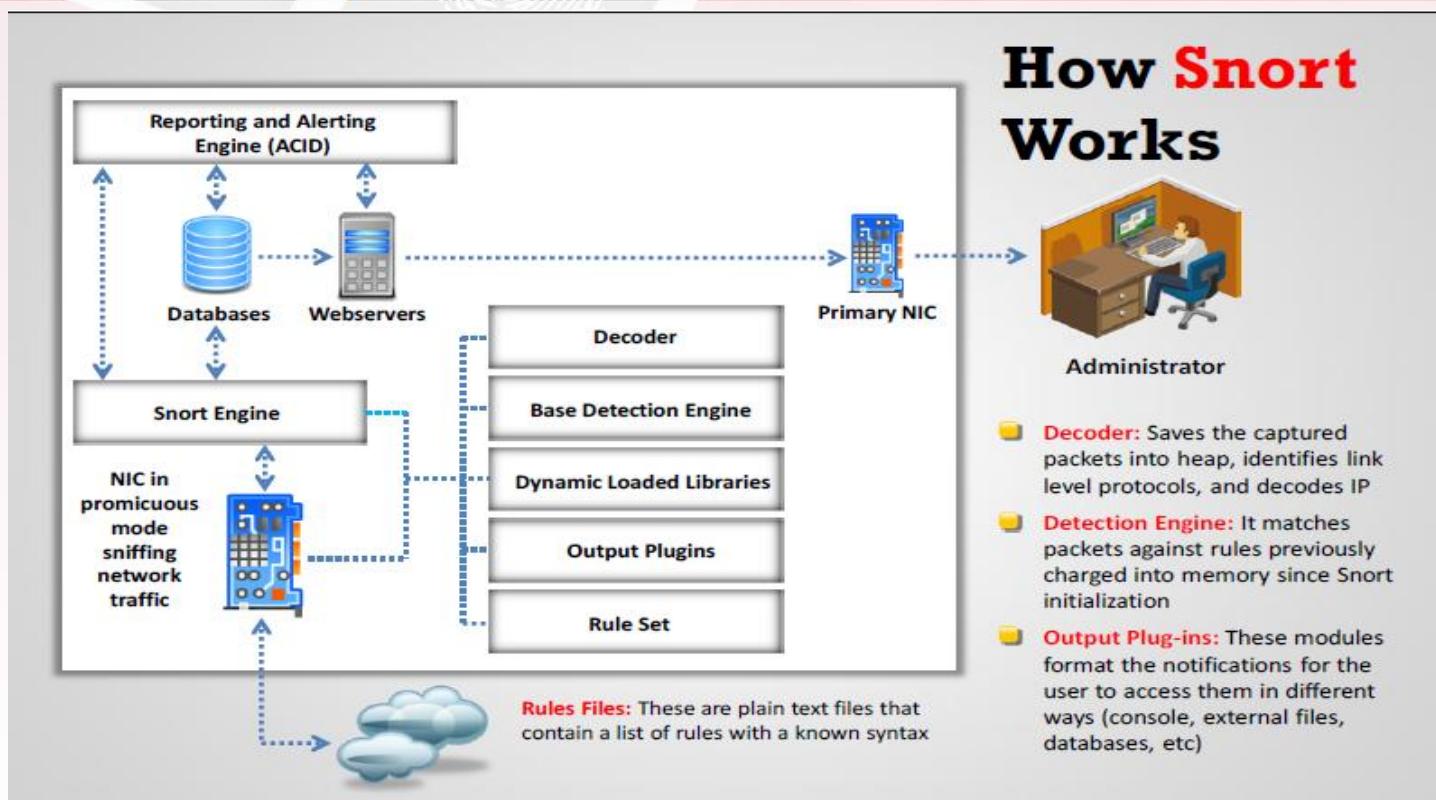
Gathering Evidence on a Network: Snort IDS Placement





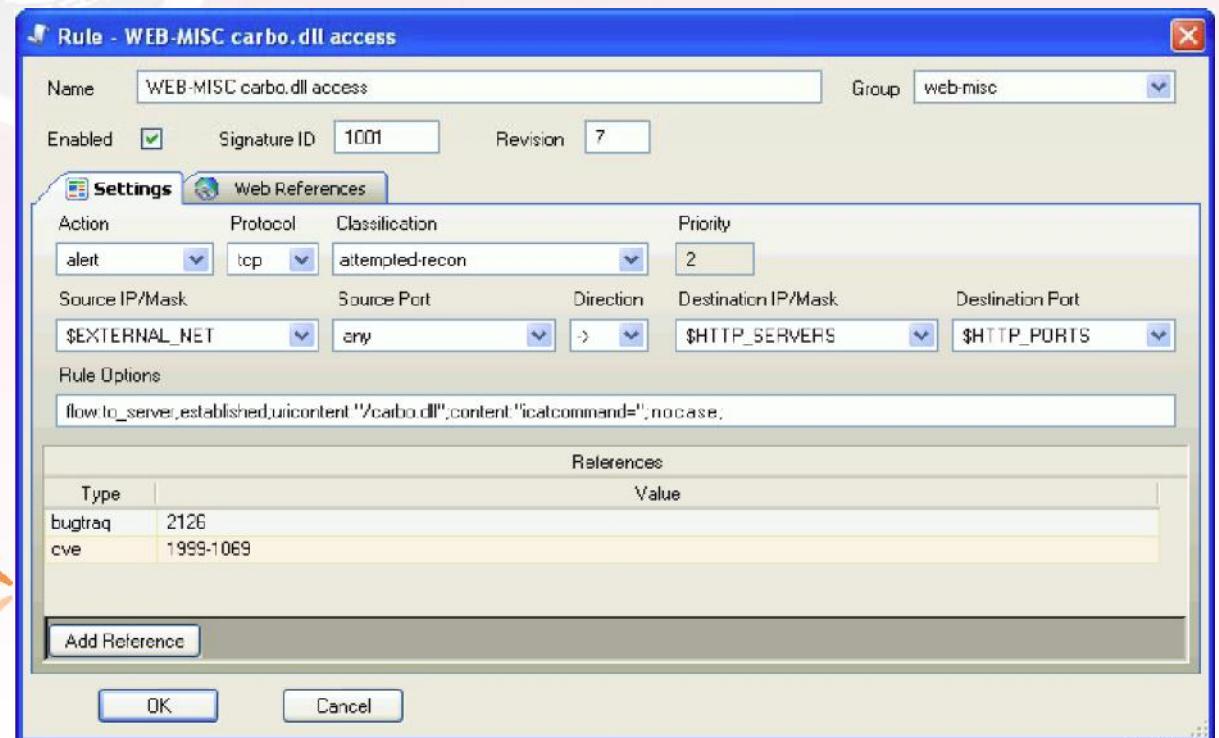
Gathering Evidence on a Network: Snort

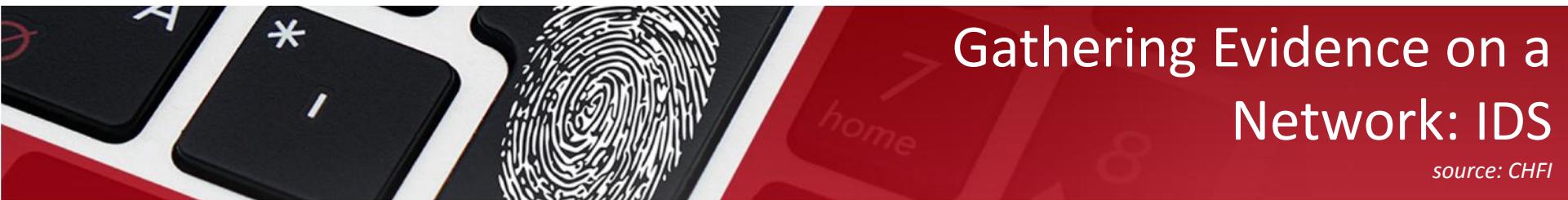
source: CHFI



Gathering Evidence on a Network: IDS Policy Manager

IDS Policy Manager has been the de facto standard for managing Snort rules on Windows. You can create Snort rules graphically.





Gathering Evidence on a Network: IDS

source: CHFI

Gathering Evidence by **IDS**

1

IDS can be configured to **capture the network traffic** and generate alerts



2

Results of networking devices such as routers and firewalls, can be recorded through a serial cable using the **Windows HyperTerminal** program or using a **UNIX script**



3

If the amount of information to be captured is huge, then **record the onscreen event** using a video camera or a relative software program



Tool: NetWitness

NetWitness Investigator

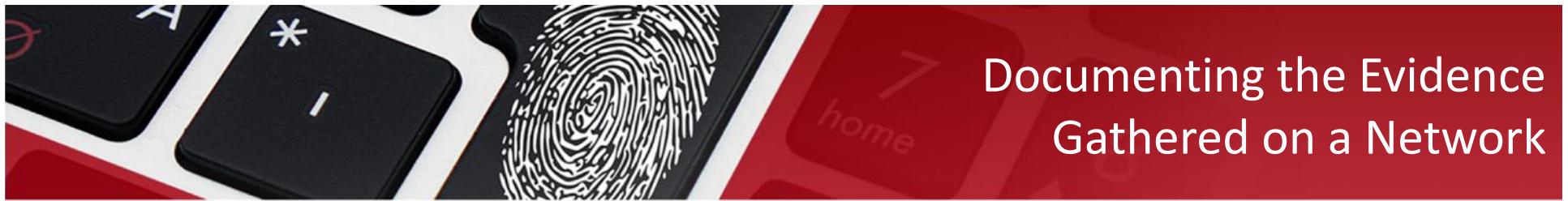
NetWitness Investigator can locally capture live traffic and process packet files from virtually any existing network collection device for quick and easy analysis

- Real-time, Patented Layer 7 Analytics
- Analyze data starting from application layer entities
- Extensive network and application layer filtering

- Integrated GeoIP for resolving IP addresses to city/county
- SSL Decryption (with server certificate)
- Interactive time charts, and summary view



Documenting the Evidence Gathered on a Network



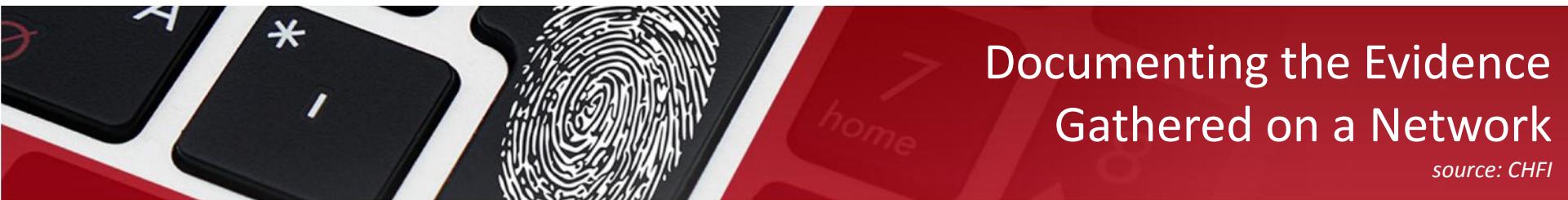
If the network logs are small, you can take a print-out and test.



Document the evidence gathering process by mentioning the name of the person who collected the evidence, from where it was collected.

- The procedure used to collect evidence and the reason for collecting evidence.

The process of documenting digital evidence on a network becomes more complex when the evidence is gathered from systems which are on remote locations.

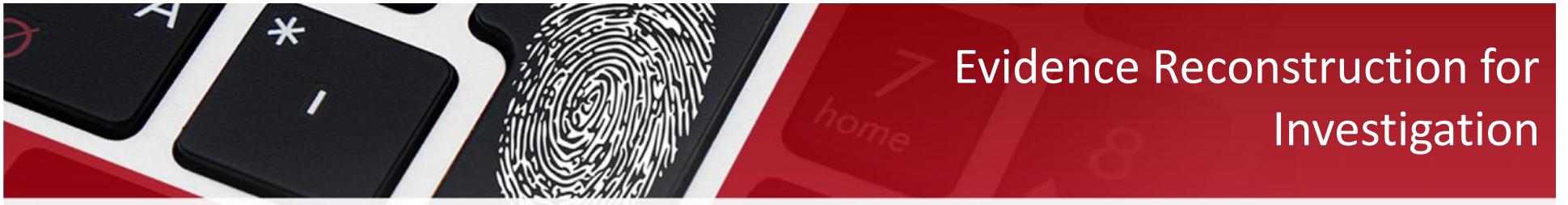


Documenting the Evidence Gathered on a Network

source: CHFI

Documenting the Evidence Gathered on a Network





Evidence Reconstruction for Investigation

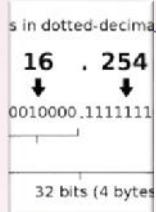
Gathering evidence trails on a network is very cumbersome for the following reasons:

- Evidence is not static and is not concentrated at a single point on the network.
- The variety of hardware and software found on the network makes the evidence gathering process more difficult.

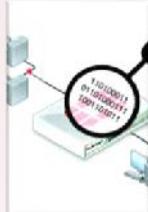
Three fundamentals of reconstruction for investigating crime are:

- Temporal analysis; helps to identify time and sequence of events.
- Relational analysis; helps to identify the link between suspect and the victim with respect to the crime.
- Functional analysis; helps to identify events that triggered the crime.

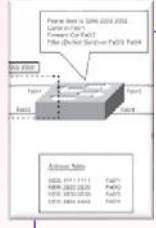
Summary



There are two types of network addressing schemes: LAN Addressing and Internetwork Addressing.



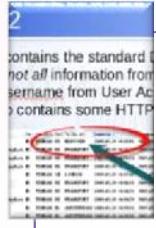
Sniffer is computer software or hardware that can intercept and log traffic passing over a digital network or part of a network.



The ARP table of a router comes handy for investigating network attacks as the table contains IP addresses associated with the respective MAC addresses.



The DHCP server maintains a list of recent queries along with the MAC address and IP address.



Application logging is meant for the storage of auditing information, which includes information produced by application activity.



IDS can be configured to capture network traffic when an alert is generated.