Network Forensics for Dummies

CS Lee[1] Meling Mudin[2]

[1]Founder, Defcraft Sdn. Bhd.
Founder and Lead Developer, Rawpacket Project and HeX LiveCD

[2]Freelance Security Consultant

HITBSecConf2009, 5th - 8th October 2009
Kuala Lumpur, Malaysia

Outline

Outline (cont.)

Outline (cont.)

Outline (cont.)

Network Forensics

Network Forensics

The use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyze, and document digital evidence from multiple, actively processing and transmitting digital sources for the purpose of uncovering facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recovery from these activities

- Gary Palmer, A Road Map for Digital Forensic Research, Report from DFRWS 2001, First Digital Forensic Research Workshop, Utica, New York, August 7-8, 2001, Page(s) 27-30

Network Forensics

Network Forensics

The capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents

- The term, attributed to firewall expert Marcus Ranum, is borrowed from the legal and criminology fields where forensics pertains to the investigation of crimes

Network Forensics

- According to Simson Garfinkel, author of several books on security, network forensics systems can be one of two kinds:

Cat-it-as-you-can System

"Catch-it-as-you-can" systems, in which all packets passing through a certain traffic point are captured and written to storage with analysis being done subsequently in batch mode. This approach requires large amounts of storage, usually involving a RAID system.

Stop, look and listen

"Stop, look and listen" systems, in which each packet is analyzed in a rudimentary way in memory and only certain information saved for future analysis. This approach requires less storage but may require a faster processor to keep up with incoming traffic.

Network Forensics

- According to Simson Garfinkel, author of several books on security, network forensics systems can be one of two kinds:

Cat-it-as-you-can System

"Catch-it-as-you-can" systems, in which all packets passing through a certain traffic point are captured and written to storage with analysis being done subsequently in batch mode. This approach requires large amounts of storage, usually involving a RAID system.

Stop, look and listen

"Stop, look and listen" systems, in which each packet is analyzed in a rudimentary way in memory and only certain information saved for future analysis. This approach requires less storage but may require a faster processor to keep up with incoming traffic.

Network Forensics

- According to Simson Garfinkel, author of several books on security, network forensics systems can be one of two kinds:

Cat-it-as-you-can System

"Catch-it-as-you-can" systems, in which all packets passing through a certain traffic point are captured and written to storage with analysis being done subsequently in batch mode. This approach requires large amounts of storage, usually involving a RAID system.

Stop, look and listen

"Stop, look and listen" systems, in which each packet is analyzed in a rudimentary way in memory and only certain information saved for future analysis. This approach requires less storage but may require a faster processor to keep up with incoming traffic.

Network Forensics

Network Forensics

Network forensics is basically about network monitoring, determining if there's any anomaly (or malicious activities), and determining the nature of attacks if any

- Important aspects include traffic capture, preservation, and analysis
- Analysis involved many activities depending on the nature of the investigation and the evidence presented. This may include
  - Reassembling packets
  - Extracting traffic contents
  - Examining network flow
  - Inspecting packet headers
  - ... and so on

Network-based forensics can answer the following

- When the particular IP has successfully compromised the system?
- What's the duration of this http session?
- Which network protocol is used in the attack?
- What's the main attribute of a successful data transfer in tcp connection?
- What's the main attribute of a successful file download/upload in ftp connection?
- What's the main attribute of a successful file download/upload in smb connection?
- What's the main attribute of a successful file download/upload in http connection?
- What's the important data when dealing with encrypted network connection?

Network-based forensics can answer the following

- When the particular IP has successfully compromised the system?
- What's the duration of this http session?
- Which network protocol is used in the attack?
- What's the main attribute of a successful data transfer in tcp connection?
- What's the main attribute of a successful file download/upload in ftp connection?
- What's the main attribute of a successful file download/upload in smb connection?
- What's the main attribute of a successful file download/upload in http connection?
- What's the important data when dealing with encrypted network connection?

Network-based forensics can answer the following

- When the particular IP has successfully compromised the system?
- What's the duration of this http session?
- Which network protocol is used in the attack?
- What's the main attribute of a successful data transfer in tcp connection?
- What's the main attribute of a successful file download/upload in ftp connection?
- What's the main attribute of a successful file download/upload in smb connection?
- What's the main attribute of a successful file download/upload in http connection?
- What's the important data when dealing with encrypted network connection?

Network-based forensics can answer the following

- When the particular IP has successfully compromised the system?
- What's the duration of this http session?
- Which network protocol is used in the attack?
- What's the main attribute of a successful data transfer in tcp connection?
- What's the main attribute of a successful file download/upload in ftp connection?
- What's the main attribute of a successful file download/upload in smb connection?
- What's the main attribute of a successful file download/upload in http connection?
- What's the important data when dealing with encrypted network connection?

Network-based forensics can answer the following

- When the particular IP has successfully compromised the system?
- What's the duration of this http session?
- Which network protocol is used in the attack?
- What's the main attribute of a successful data transfer in tcp connection?
- What's the main attribute of a successful file download/upload in ftp connection?
- What's the main attribute of a successful file download/upload in smb connection?
- What's the main attribute of a successful file download/upload in http connection?
- What's the important data when dealing with encrypted network connection?

Network-based forensics can answer the following

- When the particular IP has successfully compromised the system?
- What's the duration of this http session?
- Which network protocol is used in the attack?
- What's the main attribute of a successful data transfer in tcp connection?
- What's the main attribute of a successful file download/upload in ftp connection?
- What's the main attribute of a successful file download/upload in smb connection?
- What's the main attribute of a successful file download/upload in http connection?
- What's the important data when dealing with encrypted network connection?

Network-based forensics can answer the following

- When the particular IP has successfully compromised the system?
- What's the duration of this http session?
- Which network protocol is used in the attack?
- What's the main attribute of a successful data transfer in tcp connection?
- What's the main attribute of a successful file download/upload in ftp connection?
- What's the main attribute of a successful file download/upload in smb connection?
- What's the main attribute of a successful file download/upload in http connection?
- What's the important data when dealing with encrypted network connection?

Network-based forensics can answer the following

- When the particular IP has successfully compromised the system?
- What's the duration of this http session?
- Which network protocol is used in the attack?
- What's the main attribute of a successful data transfer in tcp connection?
- What's the main attribute of a successful file download/upload in ftp connection?
- What's the main attribute of a successful file download/upload in smb connection?
- What's the main attribute of a successful file download/upload in http connection?
- What's the important data when dealing with encrypted network connection?

Why Network Forensics?

- Tamper resistant especially if deployed in bridge/tap mode
- No performance impact on the end point
- No management impact on platforms
- Works across operating systems
- Able to derive information that host based might not provide

Why Network Forensics?

- Tamper resistant especially if deployed in bridge/tap mode
- No performance impact on the end point
- No management impact on platforms
- Works across operating systems
- Able to derive information that host based might not provide

Why Network Forensics?

- Tamper resistant especially if deployed in bridge/tap mode
- No performance impact on the end point
- No management impact on platforms
- Works across operating systems
- Able to derive information that host based might not provide

Why Network Forensics?

- Tamper resistant especially if deployed in bridge/tap mode
- No performance impact on the end point
- No management impact on platforms
- Works across operating systems
- Able to derive information that host based might not provide

Why Network Forensics?

- Tamper resistant especially if deployed in bridge/tap mode
- No performance impact on the end point
- No management impact on platforms
- Works across operating systems
- Able to derive information that host based might not provide

Advantages of Network Forensics

- Evidence recovery
  - An attacker uploaded a backdoor to a compromised server
  - She deleted the backdoor
  - ☺ We are able to retrieve the tool from our packet capture!
  - A disgruntled employee ftp'ed sensitive document from his notebook to a remote file server
  - ☺ We are able to retrieve the document, and hence have strong evidence for prosecution
- We can know exactly what is transferred and whom the attacker communicates with even though he may have deleted the evidence from the compromised server

Advantages of Network Forensics

- Evidence recovery
  - An attacker uploaded a backdoor to a compromised server
  - She deleted the backdoor
  - ☺ We are able to retrieve the tool from our packet capture!
  - A disgruntled employee ftp'ed sensitive document from his notebook to a remote file server
  - ☺ We are able to retrieve the document, and hence have strong evidence for prosecution
- We can know exactly what is transferred and whom the attacker communicates with even though he may have deleted the evidence from the compromised server

Advantages of Network Forensics

- Evidence recovery
  - An attacker uploaded a backdoor to a compromised server
  - She deleted the backdoor
  - ☺ We are able to retrieve the tool from our packet capture!
  - A disgruntled employee ftp'ed sensitive document from his notebook to a remote file server
  - ☺ We are able to retrieve the document, and hence have strong evidence for prosecution
- We can know exactly what is transferred and whom the attacker communicates with even though he may have deleted the evidence from the compromised server

Advantages of Network Forensics

- Evidence recovery
  - An attacker uploaded a backdoor to a compromised server
  - She deleted the backdoor
  - ☺ We are able to retrieve the tool from our packet capture!
  - A disgruntled employee ftp'ed sensitive document from his notebook to a remote file server
  - ☺ We are able to retrieve the document, and hence have strong evidence for prosecution
- We can know exactly what is transferred and whom the attacker communicates with even though he may have deleted the evidence from the compromised server

Advantages of Network Forensics

- Evidence recovery
  - An attacker uploaded a backdoor to a compromised server
  - She deleted the backdoor
  - ☺ We are able to retrieve the tool from our packet capture!
  - A disgruntled employee ftp'ed sensitive document from his notebook to a remote file server
  - ☺ We are able to retrieve the document, and hence have strong evidence for prosecution
- We can know exactly what is transferred and whom the attacker communicates with even though he may have deleted the evidence from the compromised server

Advantages of Network Forensics

- Evidence recovery
  - An attacker uploaded a backdoor to a compromised server
  - She deleted the backdoor
  - ☺ We are able to retrieve the tool from our packet capture!
  - A disgruntled employee ftp'ed sensitive document from his notebook to a remote file server
  - ☺ We are able to retrieve the document, and hence have strong evidence for prosecution
- We can know exactly what is transferred and whom the attacker communicates with even though he may have deleted the evidence from the compromised server

Advantages of Network Forensics

- Evidence recovery
  - An attacker uploaded a backdoor to a compromised server
  - She deleted the backdoor
  - ☺ We are able to retrieve the tool from our packet capture!
  - A disgruntled employee ftp'ed sensitive document from his notebook to a remote file server
  - ☺ We are able to retrieve the document, and hence have strong evidence for prosecution
- We can know exactly what is transferred and whom the attacker communicates with even though he may have deleted the evidence from the compromised server

Network Forensics Process

- Capture - the process of capturing packets that travels on the wire
- Record - the process of writing captured packets to storage devices
- Analysis - the process of analyzing packets

Goals

- Discover the nature of intrusion
- Complement host-based forensics

Network Forensics Process

- Capture - the process of capturing packets that travels on the wire
- Record - the process of writing captured packets to storage devices
- Analysis - the process of analyzing packets

Goals

- Discover the nature of intrusion
- Complement host-based forensics

Traditional Network Forensics Analysis Techniques

Common steps are

1. IDS alert message
2. Analyze and examine alert event
3. Packet examination (protocol header analysis)
4. Determine event (intrusion, virus, scanning, etc)
5. Escalate event

Traditional Network Forensics Analysis Techniques

Common steps are

1. IDS alert message
2. Analyze and examine alert event
3. Packet examination (protocol header analysis)
4. Determine event (intrusion, virus, scanning, etc)
5. Escalate event

Traditional Network Forensics Analysis Techniques

Common steps are

1. IDS alert message
2. Analyze and examine alert event
3. Packet examination (protocol header analysis)
4. Determine event (intrusion, virus, scanning, etc)
5. Escalate event

Traditional Network Forensics Analysis Techniques

Common steps are

1. IDS alert message
2. Analyze and examine alert event
3. Packet examination (protocol header analysis)
4. Determine event (intrusion, virus, scanning, etc)
5. Escalate event

Traditional Network Forensics Analysis Techniques

Common steps are

1. IDS alert message
2. Analyze and examine alert event
3. Packet examination (protocol header analysis)
4. Determine event (intrusion, virus, scanning, etc)
5. Escalate event

Traditional Network Forensics Analysis Techniques

Common steps are

1. IDS alert message
2. Analyze and examine alert event
3. Packet examination (protocol header analysis)
4. Determine event (intrusion, virus, scanning, etc)
5. Escalate event

Emerging Network Forensics Analysis Techniques

Different approach to network forensics analysis

1. Alert Reporting from other sources
2. Determine event occurrence
3. Session Reconstruction
4. Packet examination (protocol header and payload analysis)
5. Escalate event

Payload plays main role in malicious traffic mining!

Emerging Network Forensics Analysis Techniques

Different approach to network forensics analysis

1. Alert Reporting from other sources
2. Determine event occurrence
3. Session Reconstruction
4. Packet examination (protocol header and payload analysis)
5. Escalate event

Payload plays main role in malicious traffic mining!

Emerging Network Forensics Analysis Techniques

Different approach to network forensics analysis

1. Alert Reporting from other sources
2. Determine event occurrence
3. Session Reconstruction
4. Packet examination (protocol header and payload analysis)
5. Escalate event

Payload plays main role in malicious traffic mining!

Emerging Network Forensics Analysis Techniques

Different approach to network forensics analysis

1. Alert Reporting from other sources
2. Determine event occurrence
3. Session Reconstruction
4. Packet examination (protocol header and payload analysis)
5. Escalate event

Payload plays main role in malicious traffic mining!

Emerging Network Forensics Analysis Techniques

Different approach to network forensics analysis

1. Alert Reporting from other sources
2. Determine event occurrence
3. Session Reconstruction
4. Packet examination (protocol header and payload analysis)
5. Escalate event

Payload plays main role in malicious traffic mining!

Emerging Network Forensics Analysis Techniques

Different approach to network forensics analysis

1. Alert Reporting from other sources
2. Determine event occurrence
3. Session Reconstruction
4. Packet examination (protocol header and payload analysis)
5. Escalate event

Payload plays main role in malicious traffic mining!

Network Forensics Purpose

The major purposes are

1. Uncover malicious activities
2. Reconstruction of session data
3. Reconstruct past network event
4. Extracting evidence
5. Analyzing encrypted or hidden traffic

Network Forensics Purpose

The major purposes are

1. Uncover malicious activities
2. Reconstruction of session data
3. Reconstruct past network event
4. Extracting evidence
5. Analyzing encrypted or hidden traffic

Network Forensics Purpose

The major purposes are

1. Uncover malicious activities
2. Reconstruction of session data
3. Reconstruct past network event
4. Extracting evidence
5. Analyzing encrypted or hidden traffic

Network Forensics Purpose

The major purposes are

1. Uncover malicious activities
2. Reconstruction of session data
3. Reconstruct past network event
4. Extracting evidence
5. Analyzing encrypted or hidden traffic

Network Forensics Purpose

The major purposes are

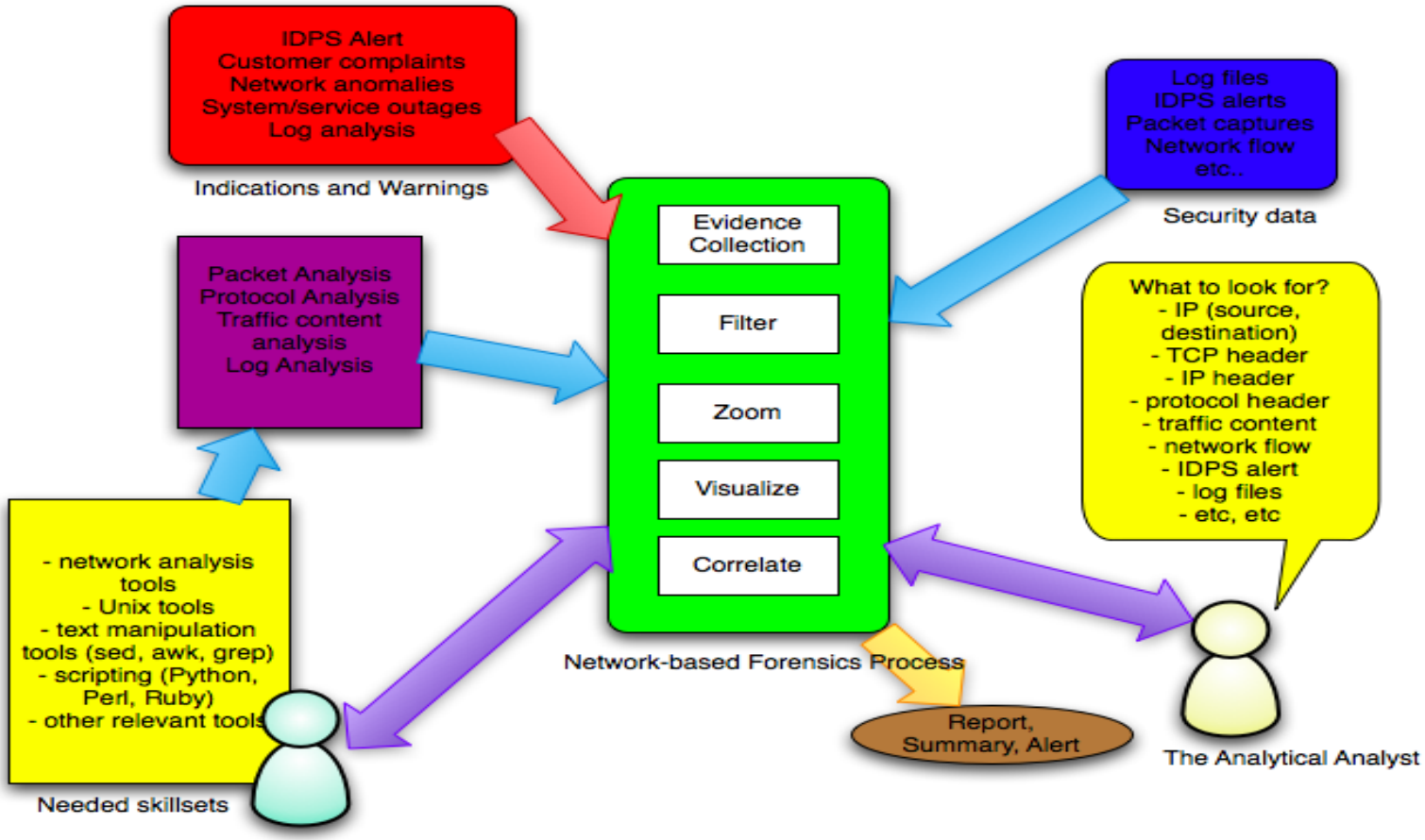
1. Uncover malicious activities
2. Reconstruction of session data
3. Reconstruct past network event
4. Extracting evidence
5. Analyzing encrypted or hidden traffic

Network Forensics Purpose

The major purposes are

1. Uncover malicious activities
2. Reconstruction of session data
3. Reconstruct past network event
4. Extracting evidence
5. Analyzing encrypted or hidden traffic

Network Forensics - The Big Picture



Network Forensics Toolkit

- Ensure data integrity
- Offer high quality protocol dissectors
- Emphasis on tools that can automatically reconstruct session data
- Derive high level information about network event from packet data
- Able to extract files or useful contents from packet capture
- Able to replay sessions such as telnet, ftp, IRC
- Free and Open Source!
- Usage of tools depends on your needs

Network Forensics Toolkit

Tools

- Xplico
- NetworkMiner
- Pyflags(Network Forensics Module)
- Bro-IDS
- Tcpxtract
- Tcpflow
- Chaosreader
- Wireshark
- Foremost
- ClamAV - anti virus

Xplico

- It's not Network Protocol Analyzer but Open Source Network Forensic Tool
- Offers Port Independent Protocol Identification(PIPI) capability
- Solid tcp re-assembler to extract application data/content from pcap file
- Output logging to database(sqlite/mysql)
- Geo Location Mapping
- Many more

Xplico CLI

- `xplico -m rltm -i eth0`

→ run in real time and listen to the network interface

- `xplico -m pcap -f capture.pcap`

→ run xplico to decode packet capture file

- `xplico -m pcap -d /tmp/pcap-directory`

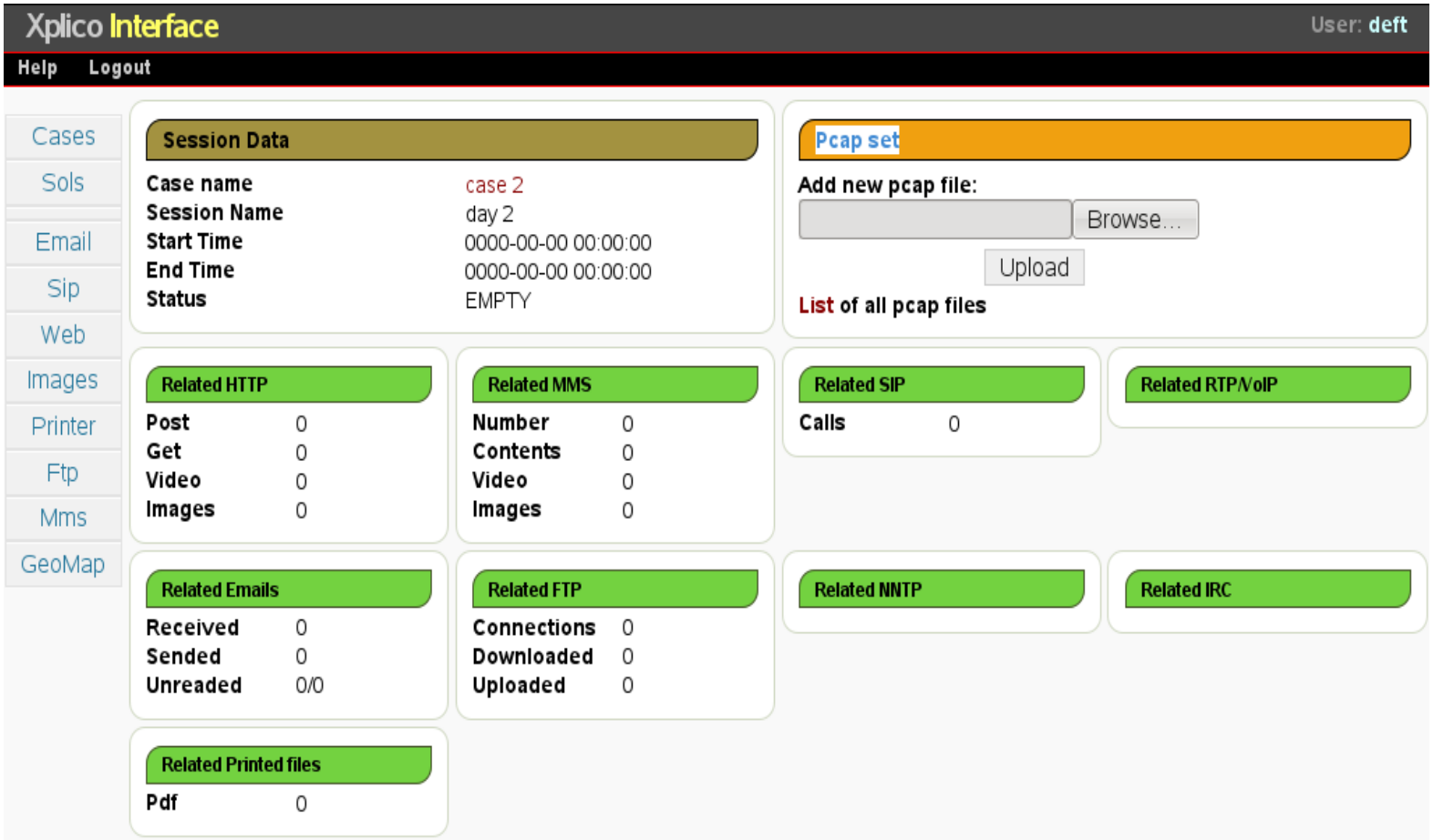
→ run xplico to decode packet capture file in the directory

- `/opt/xplico/scripts/sqlite_demo.sh`

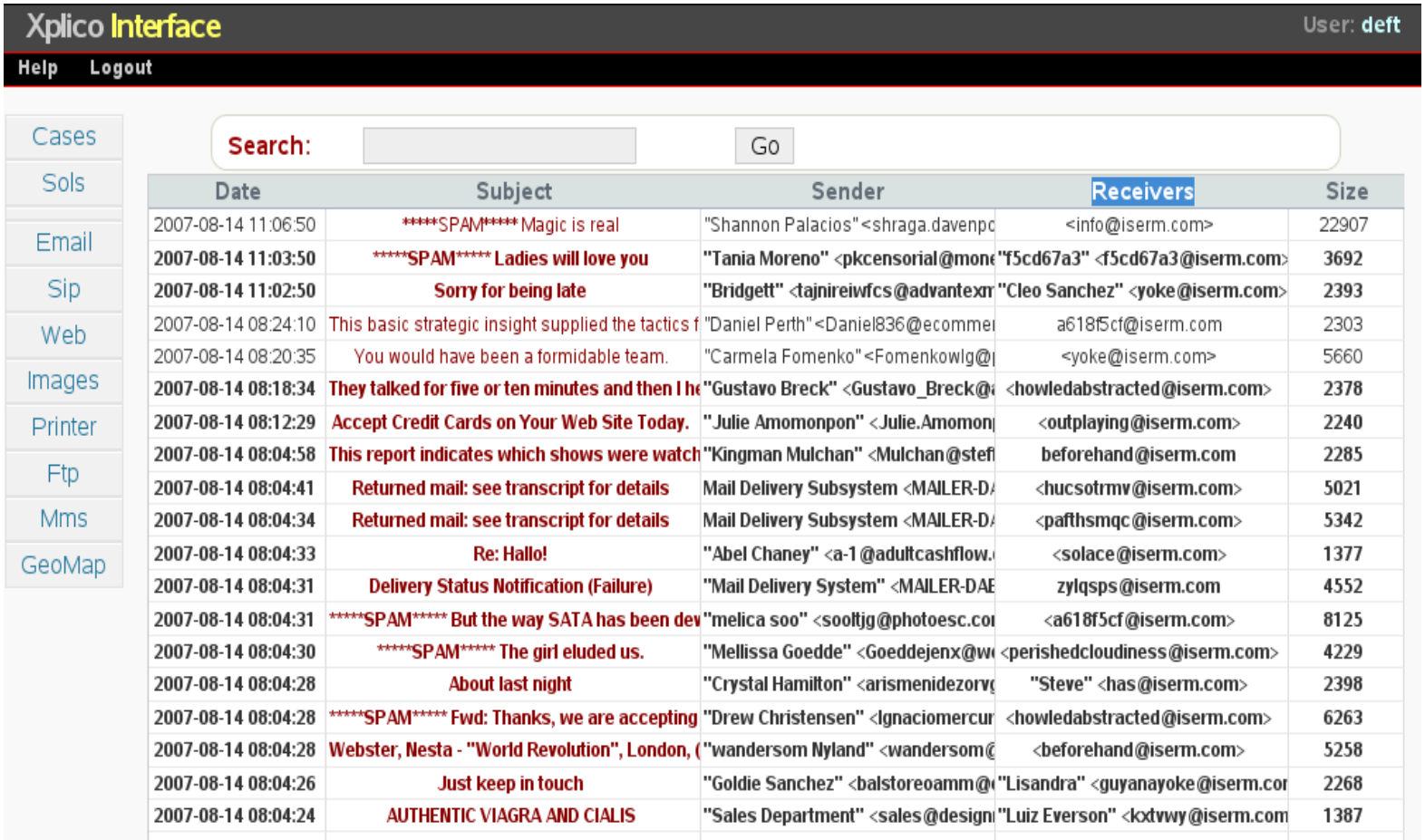
→ start xplico decoding manager

→ open url: http://localhost:9876

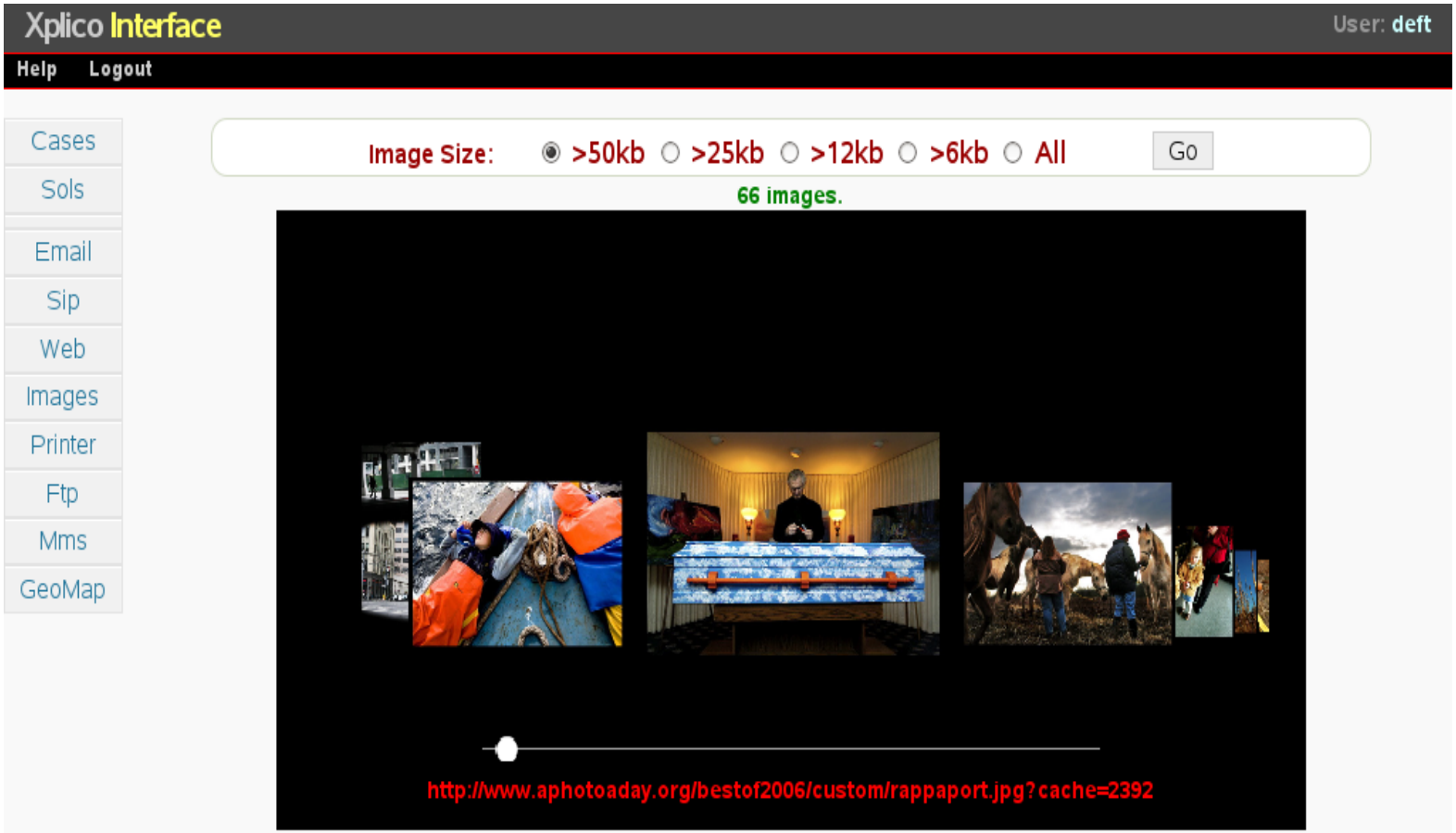
Xplico GUI - Session Page



Xplico GUI - Email Page



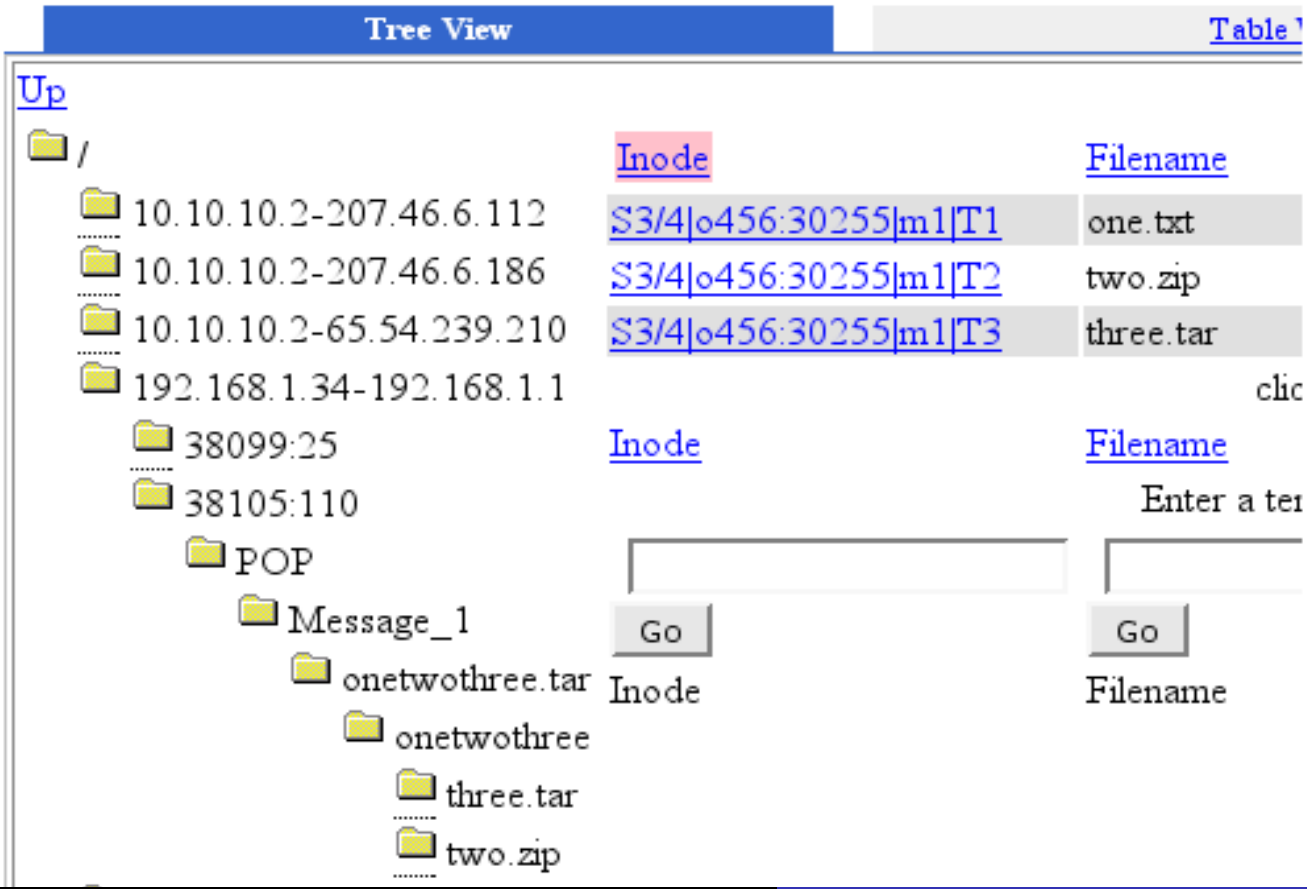
Xplico GUI - Images Page



Pyflag - Network Forensics Module

- Offers correlation with Log sources and File System/Memory Forensics
- Provide high level analysis information
- Using protocol scanner to scan against pcap file to reconstruct the stream for different network protocols
- Index pcap data

Pyflag - MSN/IRC Scanner



Pyflag - MSN Chat Session

Prox	Time Stamp	Packet	Sender Nick	Text
	2005-08-08 16:40:09	436	(Target)	hello
	2005-08-08 16:40:22	445	user022714@hotmail.com	hi there buddy
	2005-08-08 16:40:27	449	(Target)	whats up man?
	2005-08-08 16:40:39	460	user022714@hotmail.com	im getting a new plan ready
	2005-08-08 16:40:48	468	(Target)	really? what are u planning?
	2005-08-08 16:41:24	486	user022714@hotmail.com	we can hit that bank on the corner across my house
	2005-08-08 16:41:34	494	(Target)	we are gonna need some serious firepower for that

Image description: A chat session using MSN messanger. Note the packet link allowing direct access to the exact packet which contained the message.

Up

/

Frame 164 (72 bytes on wire, 72 bytes captured)

Ethernet II, Src: 00:11:50:63:6b:32 (00:11:50:63:6b:32), Dst: 00:50:bf:79:fc:8e (00:5

Internet Protocol, Src: 192.168.1.34 (192.168.1.34), Dst: 192.168.1.1 (192.168.1.

Transmission Control Protocol, Src Port: 38105 (38105), Dst Port: 110 (110), Seq:

Tcpxtract

- A tool for extracting files from network traffic based on file signatures
- Uses a technique called data carving (extracting files based on headers and footers)
- Easy to write file signaturess
- Can be used on live data capture (realtime analysis) or tcpdump capture files (offline analysis)
- http://tcpxtract.sourceforge.net/

Running tcpxtract

tcpxtract

```
tcpxtract -f malicious.pcap -c tcpxtract.conf -o
/nsm/tcpxtract
```

- tcpxtract.conf contains signatures (based on file header metadata)
- Easy to configure and add new signatures - http://filext.com/index.php

Tcpflow

- Reconstruct different TCP flow streams in separate files for analysis
- http://www.circlemud.org/ jelson/software/tcpflow/

Example

```
tcpflow -r malicious.pcap;
for i in `ls`; do file $i | awk 'print $1, $2' >
malicious.log; done
```

Chaosreader

- Full feature network forensics tool
- Reconstruct telnet, ftp, IRC, etc session from captured packet
- Able to replay session data, such as telnet session
- Generate HTML report by reading tcpdump packet capture files
- http://www.brendangregg.com/chaosreader.html

Using chaosreader

```
chaosreader malicious.pcap
```

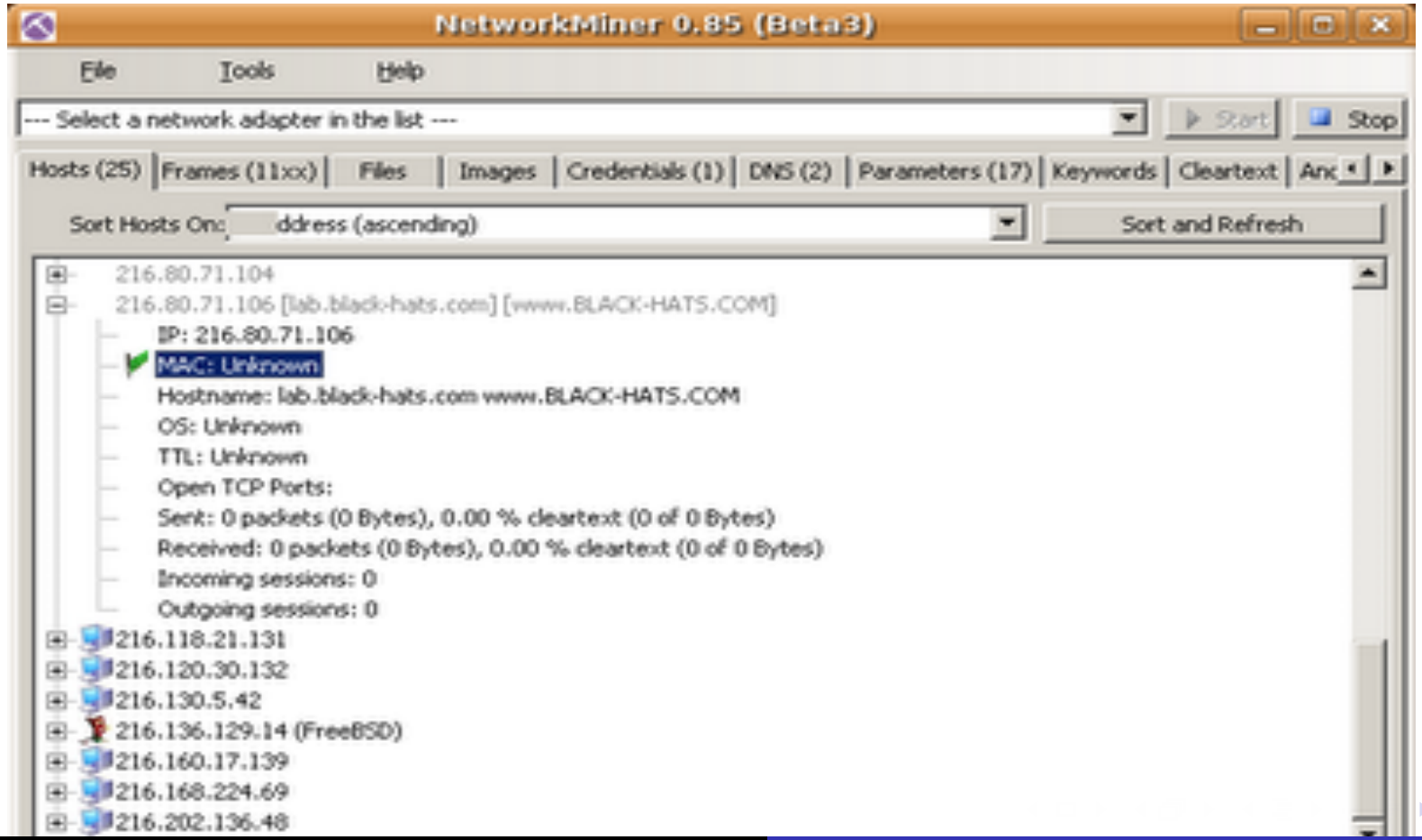
Wireshark

- Powerful expression filtering capabilities, very flexible for analyzing session data and application protocols
- Can be used to locate suspicious source and destination addresses
- Right click on the connection in the pane and follow TCP stream
- New version of Wireshark now supports "follow UDP stream" feature to reconstruct UDP event

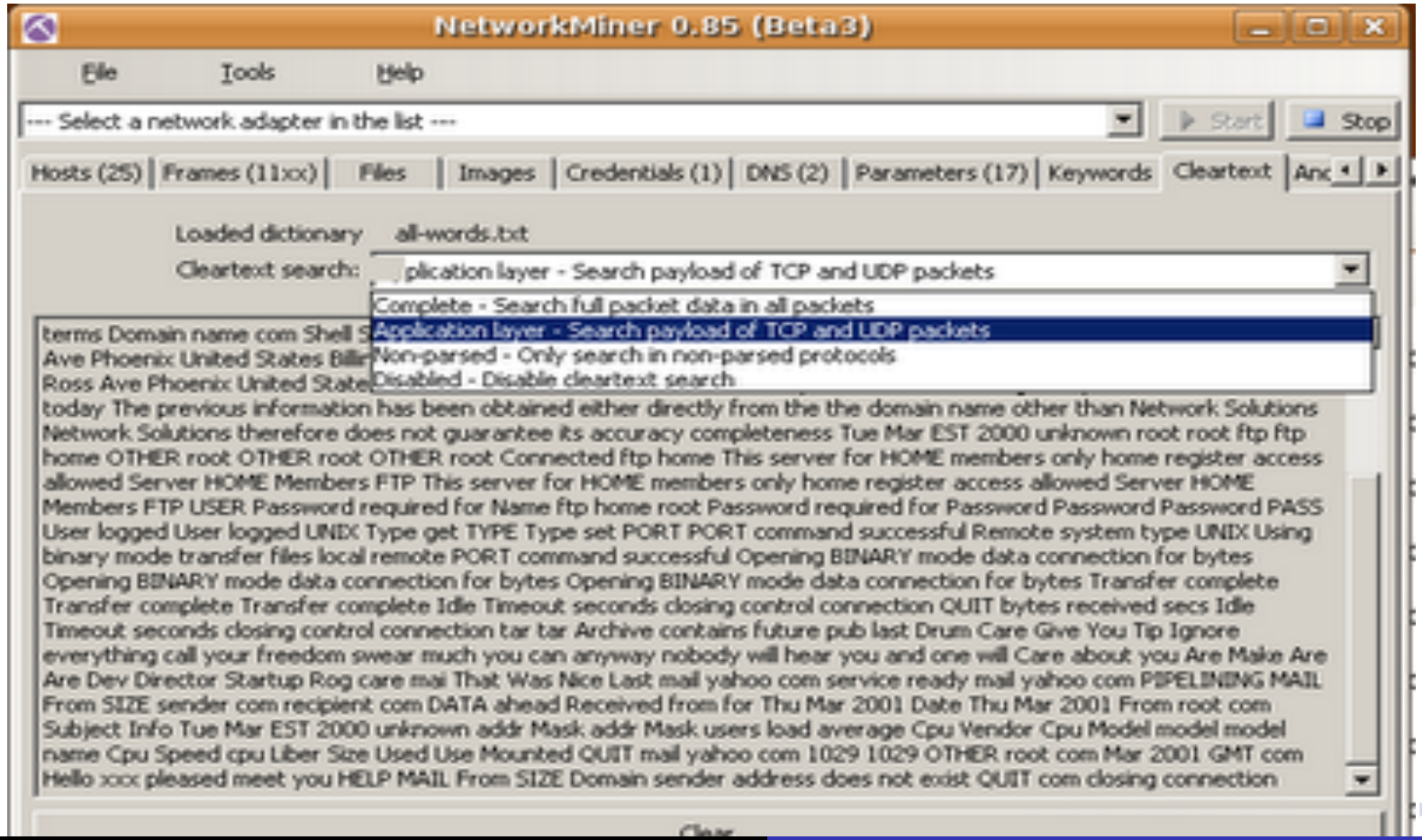
Networkminer

- Reconstruct network session and files from packet capture
- Host profiling and end point reporting
- Windows tool
- Supported protocol: http, ftp, smb and etc
- http://networkminer.sf.net/

NetworkMiner - Host Profiling



NetworkMiner - String Search



Foremost

- Able to extract files based on header, footer and file internal structure from dd image and also pcap files
- Uses a technique called data carving
- http://foremost.sourceforge.net/

Using Foremost

```
Foremost started at Sun Mar 26 13:17:03 2006
Invocation: foremost -i brontok_pcap -o /nsm/IR_20050326/extract/
Output directory: /nsm/IR_20050326/extract
Configuration file: /usr/local/etc/foremost.conf
------------------------------------------------------------------

File: brontok_pcap
Start: Sun Mar 26 13:17:03 2006
Length: 188 KB (192512 bytes)

Num Name (bs=512) Size File Offset Comment
0: 133.jpg 1 KB 68134
1: 102.gif 497 B 52662 (20 x 20)
2: 111.gif 589 B 57195 (13 x 13)
3: 5.htm 95 KB 2944
4: 372.htm 339 B 190890
5: 65.exe 41 KB 33483 01/01/1970 00:00:00
Finish: Sun Mar 26 13:17:03 2006

5 FILES EXTRACTED
jpg:= 1
gif:= 2
htm:= 2
exe:= 1
```

ClamAV

- Open Source anti virus tool
- Built-in support for various compression formats - RAR, ZIP, UPX, etc
- Malicious file detection based on signature
- Bought by Sourcefire (developers of Snort) in August 2007

Running ClamAV

```
freshclam -datadir=/var/lib/clamav/
clamscan-d /var/db/clamav -r files_dir/
```

ClamAV

```
binary/iexplore.exe: Trojan.Poebot-32 FOUND
binary/dksfjhd.exe: Trojan.Proxy.Ranky-29 FOUND
binary/fgdkj.exe: OK
binary/ffkdl: OK
binary/kgjdfhg.exe: OK
binary/fkjhg.exe: OK
binary/uiwlkja.exe: Trojan.Proxy.Ranky-29 FOUND
binary/nude_pic.scr: OK
binary/bnxiu.exe: Trojan.Proxy.Ranky-29 FOUND

----------- SCAN SUMMARY -----------
Known viruses: 60743
Engine version: devel-20060316
Scanned directories: 1
Scanned files: 9
Infected files: 4
Data scanned: 15.56 MB
Time: 7.448 sec (0 m 7 s)
```

Statistical Data Analysis

- Provides a general overview of network traffic over time, such as hours, days, weeks and months
- Answers the question (for example):
  - What is the top pair of destination ip and port?
  - Who are the "top talkers" in the network?
  - How much DNS queries are we receiving in a day?
  - Who downloads the most data?
- You can perform both offline and online statistical analysis
- In this chapter, we focus on offline analysis

Uses of Statistical Data

- Statistical data is used to form a baseline to look for deviations. For example, if a normal SMTP traffic is made up of 30% of an organization's traffic, a 40% consumption can be considered abnormal
- To have a better idea of what is happening on the network
- Statistical data does not provide an alarm but thresholding for abnormal traffic flows. Therefore we still need to use other form of NSM data (alert data) to provide intrusion detection context.
- Statistical data are usually used in a reactive manner - e.g. why the network is slow
- Also used to perform network troubleshooting, confirm in-progress DoS attacks or monitor network sessions in almost real-time

Network Session Analysis

Widely known as Network Flow Analysis

- A set of traffic that are related to the same stream
- Equivalent to a call or connection
- A flow by definition is a portion of traffic, delimited by start time and last time that belongs to one of the metered traffic groups above
- Attribute values such as source/destination addresses and ports, packet counts, byte counts, etc associated with a flow are aggregate quantities reflecting events which take place in the duration between start time and last time

Network Session Analysis (cont.)

- The start time is fixed for a given flow (usually the first packet seen in the flow)
- The last time may increase with the age of the flow and depends on the configuration of the flow meter
- In practice, a flow is a stream of packets observed by the flow meter as they pass between two endpoints

Session vs Flow

- A session is a conversation between two network endpoints
- Flow is the sequence of packets between two network endpoints that are belonging to certain network session (conversation) - there are many flows per session because the start time and last time is delimited based on the configuration of the flow meter

Unidirectional and Bidirectional Flow

- Unidirectional = single direction
- Bidirectional = both direction

```
1  Source Address | Direction | Dest Addr | Total Bytes
2  Host A           ->         Host B        50
3  Host B           ->         Host A       100
```

- This is a sample of unidirectional flow, where Host A sends 50 bytes to Host B, and Host B replies with 100 bytes

```
1  Source Address | Direction | Dest Addr | Total Bytes | Src Bytes | Dst Bytes
2  Host A           <->        Host B        150           50          100
```

Unidirectional and Bidirectional Flow (cont.)

- This is the same flow, but in bidirectional format
- Unidirectional - one direction at a time, every flow record contains the attribute of single endpoint only
- Bidirectional - both direction at a time, every flow record contains the attributes from both endpoints. The total bytes is the source bytes + destination bytes
- Argus uses bidirectional model for flow generation
- use `-M rmon` to convert Argus biflow to uniflow output
- Cisco Netflow Version 5 is unidirectional

Generic Flow Analysis

- Statistical and session based
- No false positives or negatives
- Content neutral
- Covert channel neutral
- Encryption neutral
- Used for anomaly detection, DDoS detection, network reconnaissance, etc

Generic Flow Analysis

- Statistical and session based
- No false positives or negatives
- Content neutral
- Covert channel neutral
- Encryption neutral
- Used for anomaly detection, DDoS detection, network reconnaissance, etc

Generic Flow Analysis

- Statistical and session based
- No false positives or negatives
- Content neutral
- Covert channel neutral
- Encryption neutral
- Used for anomaly detection, DDoS detection, network reconnaissance, etc

Generic Flow Analysis

- Statistical and session based
- No false positives or negatives
- Content neutral
- Covert channel neutral
- Encryption neutral
- Used for anomaly detection, DDoS detection, network reconnaissance, etc

Generic Flow Analysis

- Statistical and session based
- No false positives or negatives
- Content neutral
- Covert channel neutral
- Encryption neutral
- Used for anomaly detection, DDoS detection, network reconnaissance, etc

Generic Flow Analysis

- Statistical and session based
- No false positives or negatives
- Content neutral
- Covert channel neutral
- Encryption neutral
- Used for anomaly detection, DDoS detection, network reconnaissance, etc

GFP - Flow Accounting

- Start time, end time
- Session length
- Total packets per flow (aggregation)
- Data transfer
  - Length
  - Rate (Bps, bps, pps)
- Route info: mask, AS number

GFP - Flow Accounting

- Start time, end time
- Session length
- Total packets per flow (aggregation)
- Data transfer
  - Length
  - Rate (Bps, bps, pps)
- Route info: mask, AS number

GFP - Flow Accounting

- Start time, end time
- Session length
- Total packets per flow (aggregation)
- Data transfer
  - Length
  - Rate (Bps, bps, pps)
- Route info: mask, AS number

GFP - Flow Accounting

- Start time, end time
- Session length
- Total packets per flow (aggregation)
- Data transfer
  - Length
  - Rate (Bps, bps, pps)
- Route info: mask, AS number

GFP - Flow Accounting

- Start time, end time
- Session length
- Total packets per flow (aggregation)
- Data transfer
  - Length
  - Rate (Bps, bps, pps)
- Route info: mask, AS number

GFP - Flow Accounting

- Start time, end time
- Session length
- Total packets per flow (aggregation)
- Data transfer
  - Length
  - Rate (Bps, bps, pps)
- Route info: mask, AS number

GFP - Flow Accounting

- Start time, end time
- Session length
- Total packets per flow (aggregation)
- Data transfer
  - Length
  - Rate (Bps, bps, pps)
- Route info: mask, AS number

Session/Flow Data Metrics

- IP: source and destination ip, TTL
- TCP: source and destination ports, TCP flags
- UDP: source and destination ports
- ICMP: type and code

Session/Flow Data Metrics

- IP: source and destination ip, TTL
- TCP: source and destination ports, TCP flags
- UDP: source and destination ports
- ICMP: type and code

Session/Flow Data Metrics

- IP: source and destination ip, TTL
- TCP: source and destination ports, TCP flags
- UDP: source and destination ports
- ICMP: type and code

Session/Flow Data Metrics

- IP: source and destination ip, TTL
- TCP: source and destination ports, TCP flags
- UDP: source and destination ports
- ICMP: type and code

Traffic with high connection rate

- DoS attempt
- Worm activities
- Port scanning
- Most scanning activities can be detected by the number of attempts to connect to certain ports

Traffic with high connection rate

- DoS attempt
- Worm activities
- Port scanning
- Most scanning activities can be detected by the number of attempts to connect to certain ports

Traffic with high connection rate

- DoS attempt
- Worm activities
- Port scanning
- Most scanning activities can be detected by the number of attempts to connect to certain ports

Traffic with high connection rate

- DoS attempt
- Worm activities
- Port scanning
- Most scanning activities can be detected by the number of attempts to connect to certain ports

Traffic with high packet rate

- Indicates worm activities, portscanning
- A sudden appearance of high packet rates linked to a previous session with IDS alerts may indicate a successful compromise

Network Scanning

- Slow and single flows
- Large sum of small flows from and/or to an IP address
- Return packets are usually RST for TCP and ICMP and port unreachable for UDP

Worm activities

- May scan bogon space
- The payload maybe downloaded from specific malicious websites
- Source address is spoofed
- Each variant has almost identical payload size
- Target multiple hosts, but only targeting a single port on each host

Covert channel

- It's not undetectable (you think it's hidden!)
- It bypass filtering/detection devices
- Difficult to track in a network with heavy traffic
- Carried over well known protocols (HTTP, ICMP, DNS)
- Long flows while short ones are expected (lookups)
- Symmetric vs asymmetric traffic (e.g. HTTP)
- Large payloads and inconsistent payloads (e.g. ICMP request and reply)

Covert channel

- It's not undetectable (you think it's hidden!)
- It bypass filtering/detection devices
- Difficult to track in a network with heavy traffic
- Carried over well known protocols (HTTP, ICMP, DNS)
- Long flows while short ones are expected (lookups)
- Symmetric vs asymmetric traffic (e.g. HTTP)
- Large payloads and inconsistent payloads (e.g. ICMP request and reply)

Covert channel

- It's not undetectable (you think it's hidden!)
- It bypass filtering/detection devices
- Difficult to track in a network with heavy traffic
- Carried over well known protocols (HTTP, ICMP, DNS)
- Long flows while short ones are expected (lookups)
- Symmetric vs asymmetric traffic (e.g. HTTP)
- Large payloads and inconsistent payloads (e.g. ICMP request and reply)

Covert channel

- It's not undetectable (you think it's hidden!)
- It bypass filtering/detection devices
- Difficult to track in a network with heavy traffic
- Carried over well known protocols (HTTP, ICMP, DNS)
- Long flows while short ones are expected (lookups)
- Symmetric vs asymmetric traffic (e.g. HTTP)
- Large payloads and inconsistent payloads (e.g. ICMP request and reply)

Covert channel

- It's not undetectable (you think it's hidden!)
- It bypass filtering/detection devices
- Difficult to track in a network with heavy traffic
- Carried over well known protocols (HTTP, ICMP, DNS)
- Long flows while short ones are expected (lookups)
- Symmetric vs asymmetric traffic (e.g. HTTP)
- Large payloads and inconsistent payloads (e.g. ICMP request and reply)

Covert channel

- It's not undetectable (you think it's hidden!)
- It bypass filtering/detection devices
- Difficult to track in a network with heavy traffic
- Carried over well known protocols (HTTP, ICMP, DNS)
- Long flows while short ones are expected (lookups)
- Symmetric vs asymmetric traffic (e.g. HTTP)
- Large payloads and inconsistent payloads (e.g. ICMP request and reply)

Covert channel

- It's not undetectable (you think it's hidden!)
- It bypass filtering/detection devices
- Difficult to track in a network with heavy traffic
- Carried over well known protocols (HTTP, ICMP, DNS)
- Long flows while short ones are expected (lookups)
- Symmetric vs asymmetric traffic (e.g. HTTP)
- Large payloads and inconsistent payloads (e.g. ICMP request and reply)

Alert Data Analysis

- "Decides" what traffic is bad based on "rules" or "signatures"
- Looks for signs of intrusions and generate "alerts" based detection engine
- We cover two popular Open Source IDS software:
  - Snort IDS (http://www.snort.org)
  - BRO IDS (http://www.bro-ids.org)

Snort

- Popular Open Source Network Intrusion Detection And Prevention System
- Commercialized appliance is developed by Sourcefire
- Version 2.x relies on advance signature detection engine
- Version 3,x introduce Snort Security Platform(Snortsp) and it will become more modular
- Usage `snort -c $SNORT_CONF -b -y -r capture.pcap -l $LOG_DIR`
- This will generate `alert` and `snort.log.xxxxxxxx` files

Bro IDS

- Real-time network analysis framework, and can be used for purely network analysis
- Bro emphasize on application level-semantics, and rarely inspect individual packets
- Bro also tracks information over time, both between and across sessions
- It has no presumption of "good" or "bad" - Bro is policy-neutral
- Which means that Bro does not do signature-matching or anomaly detection unless you define them

Application Protocol Analysis

The primary objectives are -

1. Focus on application protocol specification and its properties
2. Understand widely used protocol deeply
3. Identify file transfer from certain protocol usage
4. Extract useful information

Application Protocol Analysis

The primary objectives are -

1. Focus on application protocol specification and its properties
2. Understand widely used protocol deeply
3. Identify file transfer from certain protocol usage
4. Extract useful information

Application Protocol Analysis

The primary objectives are -

1. Focus on application protocol specification and its properties
2. Understand widely used protocol deeply
3. Identify file transfer from certain protocol usage
4. Extract useful information

Application Protocol Analysis

The primary objectives are -

1. Focus on application protocol specification and its properties
2. Understand widely used protocol deeply
3. Identify file transfer from certain protocol usage
4. Extract useful information

Application Protocol Analysis

The primary objectives are -

1. Focus on application protocol specification and its properties
2. Understand widely used protocol deeply
3. Identify file transfer from certain protocol usage
4. Extract useful information

Application Protocol Analysis

Protocol analysis offers deeper insights into attacks, and confirm hypotheses created during generic packet analysis, for example:

- Confirms a successful FTP transaction
- Confirms a successful file deletion
- Confirms a successful DNS poisoning
- Determine the exploit and payload used
- To a certain extent, gauge the skills of the attacker, e.g.
  - an attacker that uses evasion techniques may exhibit a level of expertise and sophistication
  - this also indicate a targeted attack as it shows that the attacker is being cautious

DNS Packet: DNS Txt Query

```
    1 0.000000        192.168.170.8         192.168.170.20        DNS      Standard query TXT google.com
▷ Ethernet II, Src: AsustekC_b1:0c:ad (00:e0:18:b1:0c:ad), Dst: QuantaCo_32:41:8c (00:c0:9f:32:41:8c)
▷ Internet Protocol, Src: 192.168.170.8 (192.168.170.8), Dst: 192.168.170.20 (192.168.170.20)
▷ User Datagram Protocol, Src Port: 32795 (32795), Dst Port: domain (53)
▽ Domain Name System (query)
    Transaction ID: 0x1032
  ▽ Flags: 0x0100 (Standard query)
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data OK: Non-authenticated data is unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▽ Queries
    ▽ google.com: type TXT, class IN
        Name: google.com
        Type: TXT (Text strings)
        Class: IN (0x0001)

0000  00 c0 9f 32 41 8c 00 e0  18 b1 0c ad 08 00 45 00   ...2A... ......E.
0010  00 38 00 00 40 00 40 11  65 47 c0 a8 aa 08 c0 a8   .8..@.@. eG......
```

DNS Packet: DNS Txt Respond

```
2 0.000530        192.168.170.20        192.168.170.8        DNS        Standard query response TXT

▷ User Datagram Protocol, Src Port: domain (53), Dst Port: 32795 (32795)
▽ Domain Name System (response)
    [Request In: 1]
    [Time: 0.000530000 seconds]
    Transaction ID: 0x1032
  ▷ Flags: 0x8180 (Standard query response, No error)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▽ Queries
    ▷ google.com: type TXT, class IN
  ▽ Answers
    ▽ google.com: type TXT, class IN
        Name: google.com
        Type: TXT (Text strings)
        Class: IN (0x0001)
        Time to live: 4 minutes, 30 seconds
        Data length: 16
        Text: v=spf1 ptr ?all

0020  aa 08 00 35 80 1b 00 40  c7 25 10 32 81 80 00 01   ...5...@ .% 2....
0030  00 01 00 00 00 00 06 67  6f 6f 67 6c 65 03 63 6f   .......g oogle.co
```

Example - Analyzing DNS using tcpdump/tshark

- `tshark -t ad -nr capture -R 'tcp.port==53 or udp.port==53'`

→ display DNS traffic

- `tshark -t ad -nr capture.pcap -R 'dns.flags==0x0100'`

→ display DNS Query traffic

- `tshark -t ad -nr capture.pcap -R 'dns.flags==0x8180'`

→ display DNS Query Response traffic

HTTP Packet

```
4 0.322706    192.168.1.100          64.13.134.48          HTTP    GET /dist/nm

Ethernet II, Src: Apple_b0:ac:fc (00:1e:c2:b0:ac:fc), Dst: BillionE_67:d6:43 (00:04:ed:6
Internet Protocol, Src: 192.168.1.100 (192.168.1.100), Dst: 64.13.134.48 (64.13.134.48)
Transmission Control Protocol, Src Port: 61580 (61580), Dst Port: http (80), Seq: 1, Ack
Hypertext Transfer Protocol
  GET /dist/nmap-4.76-setup.exe HTTP/1.1\r\n
    Request Method: GET
    Request URI: /dist/nmap-4.76-setup.exe
    Request Version: HTTP/1.1
  Host: nmap.org\r\n
  User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.0.3) Gecko/2
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
  Accept-Language: en-us,en;q=0.5\r\n
  Accept-Encoding: gzip,deflate\r\n
  Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
  Keep-Alive: 300\r\n
  Connection: keep-alive\r\n
```

Example - Analyzing HTTP using tcpdump/tshark

- `tshark -xVr test.pcap -R 'http contains "application/xml"'`

→ display HTTP traffic that contains binary file or possible executable

- `tshark -xVr test.pcap -R 'http contains "Cookie"'`

→ display HTTP traffic that contains the string 'Coookie'

- `tshark -r capture.pcap -nqz http,stat`

→ display HTTP statistics

HTTP Request Methods

- GET
- POST
- HEAD
- PUT
- DELETE
- TRACE
- OPTIONS

These are some of the examples of HTTP request methods. Along with the request methods, there are HTTP headers that are sent along by the client, explained the slide "HTTP Header Fields"

HTTP Response Codes

- 1xx - Informational
- 2xx - Successful
- 3xx - Redirection
- 4xx - Bad Request (Client Error)
- 5xx - Unsuccessful (Server Error)

These are the response codes returned by the HTTP server for HTTP requests. Along with the codes, response headers will be sent to the client as well.

Common HTTP Status Codes

- 200 - OK
- 206 - Partial Content
- 301 - Moved permanently
- 302 - Found
- 304 - Not modified
- 401 - Unauthorized (password required)
- 403 - Forbidden (e.g. no index.html or indexing off)
- 404 - Not found
- 500 - Internal server error

For a full list and explanation, read
http://en.wikipedia.org/wiki/List_of_HTTP_status_codes and
http://tools.ietf.org/html/rfc2616#section-10

Common HTTP Response Codes

- Not all successful HTTP attack returns HTTP 200
- For example, in most cases of SQL injection, the return code is HTTP 500, which may indicate a successful SQL injection attack

HTTP Header Fields

These are some of the example of HTTP header fields. Some of the fields are sent by servers, while some are sent by the client. When performing application forensics, pay attention to the fields sent by both client and servers. A compromised server (with covert channel) may behave differently.

- Accept (request)
- Accept-Charset (request)
- Accept-Encoding (request)
- Accept-Language (request)
- Content-Encoding (response)

HTTP Header Fields (cont.)

- Etag (response)
- User-Agent (request)
- Host (request)
- Cookie (request)

More at
http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

HTTP Content-Type (Internet Media Type)

- application/zip
- application/msword
- image/jpeg
- image/tiff
- video/mpeg
- video/quicktime
- application/octet-stream

Interesting HTTP Headers

- User-Agent - identifies the client that request a HTTP resource.
- http://securitylabs.websense.com/content/Blogs/2763.aspx - an example of a site conditionally serving up malicious content depending on user-agent string
- http://www.emergingthreats.net/rules/emerging-web.rules - contains rules to detect non-standard user-agent strings

Decoding HTTP Protocol

- `tshark -t ad -Tfields -e 'http.request.method' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.response.code' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.user_agent' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.content_type' -nr capture.pcap | grep -o '.*'`

Decoding HTTP Protocol

- `tshark -t ad -Tfields -e 'http.request.method' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.response.code' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.user_agent' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.content_type' -nr capture.pcap | grep -o '.*'`

Decoding HTTP Protocol

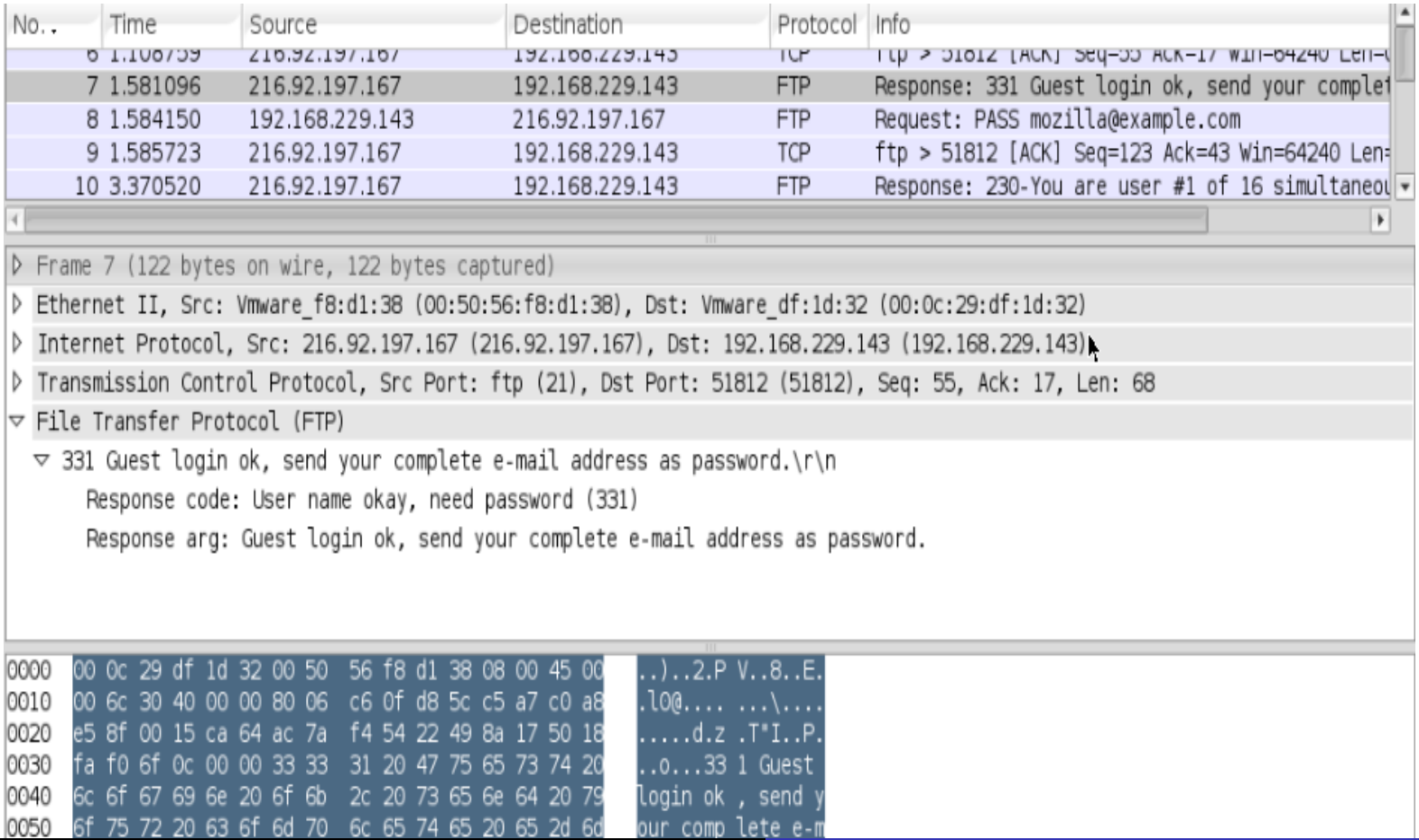
- `tshark -t ad -Tfields -e 'http.request.method' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.response.code' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.user_agent' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.content_type' -nr capture.pcap | grep -o '.*'`

Decoding HTTP Protocol

- `tshark -t ad -Tfields -e 'http.request.method' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.response.code' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.user_agent' -nr capture.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'http.content_type' -nr capture.pcap | grep -o '.*'`

Wireshark HTTP Display Filters

FTP Protocol



FTP Connection Method

- Active Mode
- Passive Mode
- Extended Passive Mode

Active FTP Mode

1. Client initiates connection to FTP server's port (port 21), the client's port is sent to the FTP server as well (port greater than 1023, e.g N)
2. FTP server will respond to the client (port 21 on server ¿ port N on client)
3. FTP server will initiate a connection from port 20 to the client on port N+1
4. Client will send an ACK from port N+1 to FTP server's port 20

PORT command is used by the client in order to use active FTP mode. Summary:

```
1 command: client > 1023 -> server 21
2 data:    client > 1023 <- server 20
```

Passive FTP Mode

In passive mode, client initiates both connection to the FTP server in order to solve the problem of firewall filtering connections from the FTP server to the client

- Client initiates connection to FTP server's port (port 21) from client port greater than 1024 (e.g N)
- FTP server acks the connection (from port 21 to port N)
- Client initiates a connection to a port specified by the FTP server from client's port N+1
- FTP server acks the connection and sends data

PASV command is used to initiate passive FTP mode. Summary:

```
1 command: client > 1023 -> server 21
2 data   : client > 1023 -> server > 1023
```

Extended Passive Mode

- In extended passive mode, the FTP server operates exactly as passive mode, however it only transmits the port number (not broken into high and low bytes)
- The client will then connect to the FTP server

FTP Commands

- USER
- PASS
- DELE
- STOR
- STOU
- RMD
- RETR
- ABOR
- QUIT

FTP Commands (cont.)

- LIST
- SIZE
- PORT
- ... and many more!

More information can be found at RFC959 - http://www.faqs.org/rfcs/rfc959.html

FTP Return Codes

- The FTP responses make it possible to ensure synchronisation between the client and FTP server. So, at each command sent by the client, the server will potentially carry out an action and systematically send back a response.
- The responses are made up of a 3 digit code indicating the way in which the command sent by the client has been processed. However, since this 3 digit code is hard to read for human, it is accompanied by a text (Telnet character string separated from the numeric code by a space).
- The response codes are made up of 3 numbers the meanings of which are as follows:

FTP Return Codes (cont.)

- → The first number indicates the status of the response (success or fail)
- → The second number indicates what the response refers to.
- → The third number gives a more specific meaning (relative to each second digit)

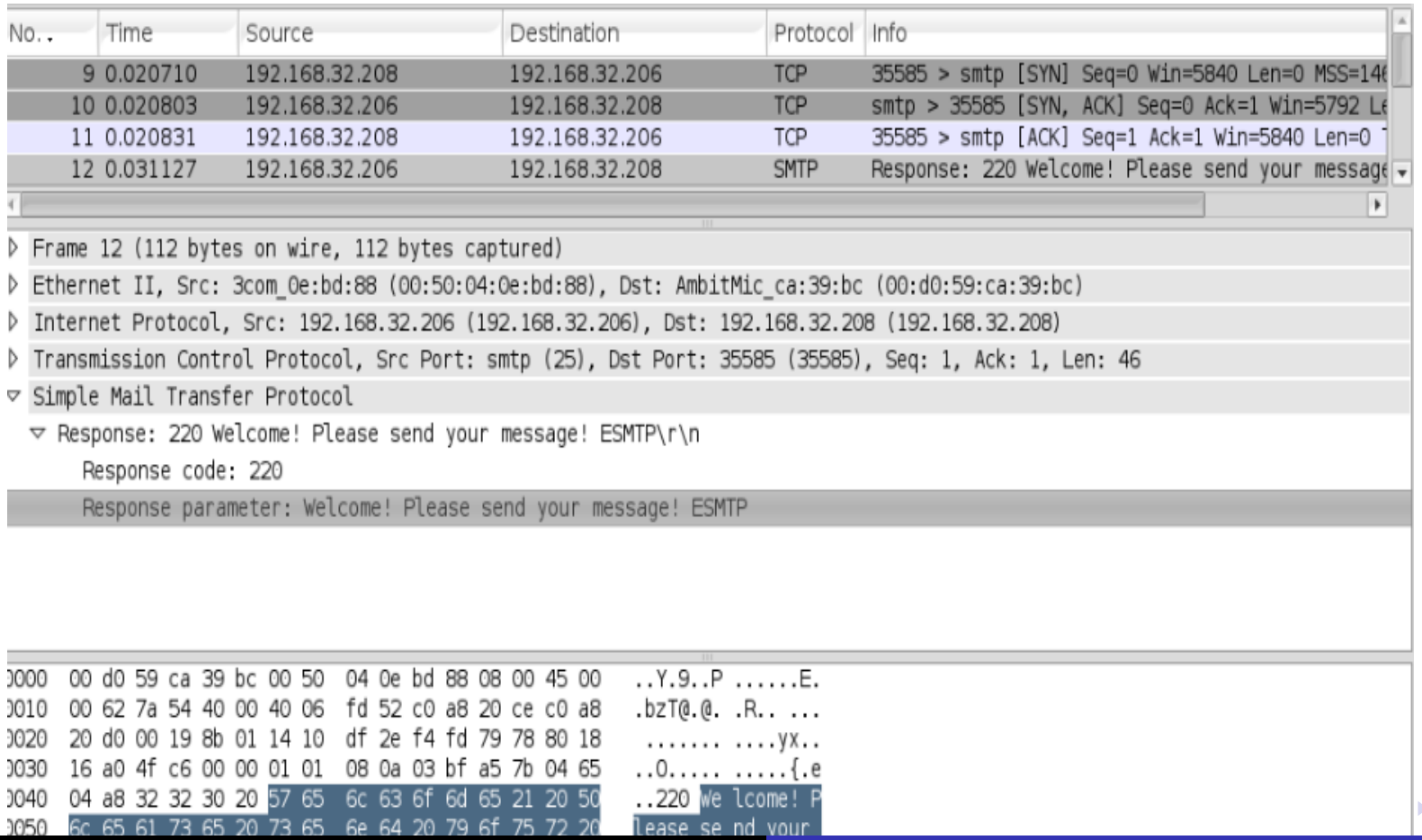
FTP Return Codes

- 1xx - Preliminary positive response
- 2xx - Positive fulfillment response
- 3xx - Intermediary positive response
- 4xx - Negative fulfillment response
- 5xx - Permanent negative response
- x0x - Syntax
- x1x - Information
- x2x - Connections
- x3x - Authentication and Account
- x4x - Unused
- x5x - File system

Decoding FTP Protocol

- `tshark -t ad -Tfields -e 'ftp.request.command' -nr ftp-download.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'ftp.response.code' -nr ftp-download.pcap | grep -o '.*'`
- `tshark -t ad -Tfields -e 'ftp.passive.ip' -nr ftp-download.pcap | grep -o '.*' | sort | uniq`

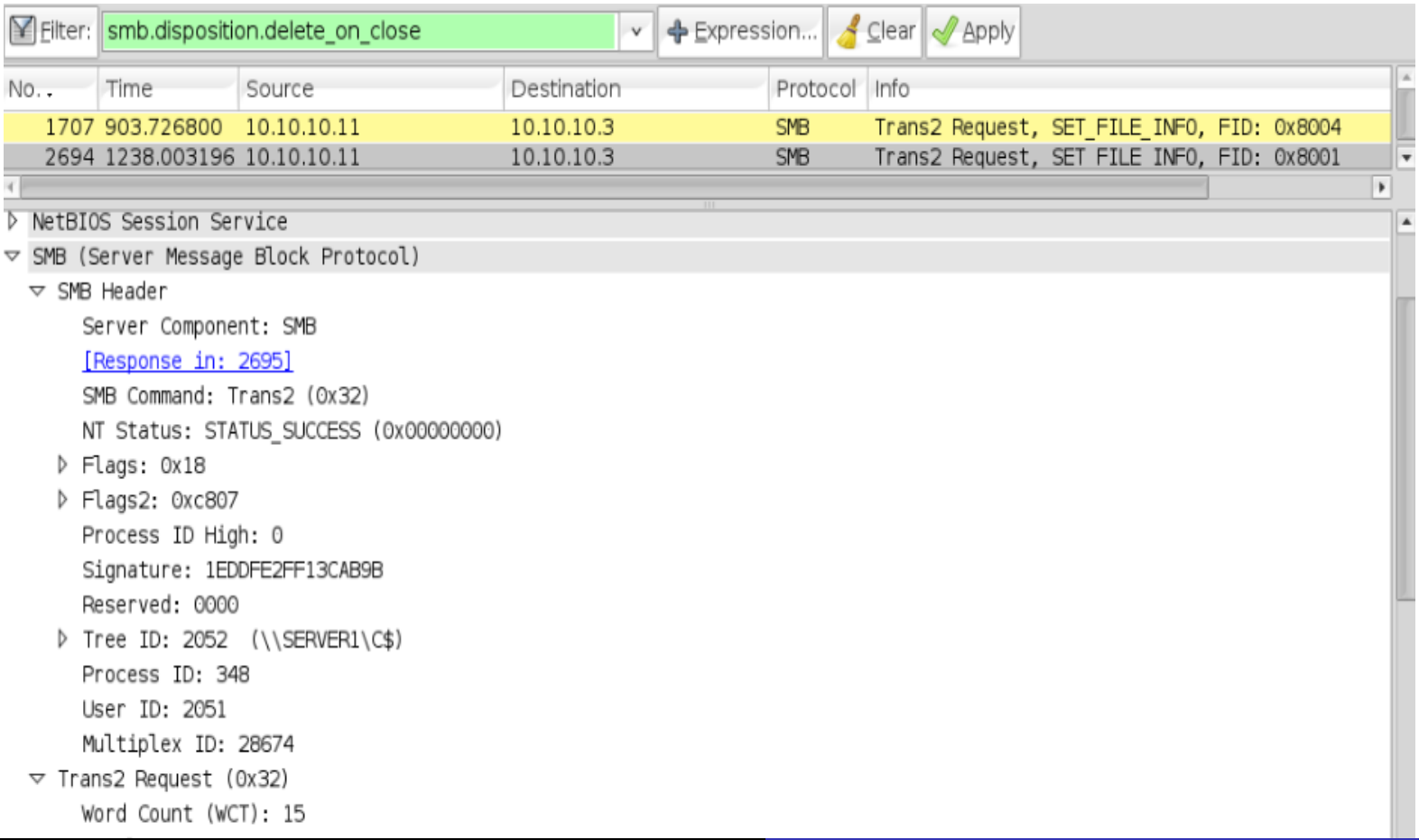
SMTP Protocol



Decoding SMTP Protocol

- `tshark -t ad -Tfields -e 'smtp.req.command' -nr trg-open-proxy.pcap -R 'smtp.req==1'`
- `tshark -t ad -tfields -e 'smtp.response.code' -nr trg-open-proxy.pcap -R 'smtp.rsp==1'`
- `tshark -Tfields -e 'smtp.req.parameter' -t ad -nr trg-open-proxy.pcap -R 'smtp.req==1'`
- `tshark -Tfields -e 'smtp.rsp.parameter' -t ad -nr trg-open-proxy.pcap -R 'smtp.rsp==1'`

SMB Protocol



Decoding SMB Protocol

- `tshark -t ad -nnr capture.pcap -R 'smb.disposition.delete_on_close==1'`
- `tshark -t ad -nnr capture.pcap -R 'smb.disposition.delete_on_close==1 or (smb.cmd==0x04 and (smb.fid==0x8004 or smb.fid==0x8001))'`
- `tshark -t ad -V -nnr capture.pcap -R 'smb.disposition.delete_on_close==1 or (smb.cmd==0x04 and (smb.fid==0x8004 or smb.fid==0x8001))' | egrep '(Path:|FID:|File Name:)' | less`

Specific Tools For Protocol Analysis

There are specific tools developed to handle certain type of network protocols and they are very useful when handle the specific protocol stack

- HTTP - httpry
- HTTPS - ssldump
- DNS - dnscap, dnstop
- TFTP tftpgrab
- MSN - msndump
- AIMS - aimsniff
- YM - yahsnarf
- Etc

Challenges in Protocol Header Analysis

- On the part of the analyst, there's a myriad of protocols that needs to be understood
- Without proper and in-depth knowledge of application protocols, the analysis will be severely hindered
- Writing application protocol decoders are not easy, and incorrect tool implementation may provide inaccurate representations of the protocol
  - Wireshark has evolved over the years and proven itself capable of decoding all kinds of application protocols

Challenges in Protocol Header Analysis

- On the part of the analyst, there's a myriad of protocols that needs to be understood
- Without proper and in-depth knowledge of application protocols, the analysis will be severely hindered
- Writing application protocol decoders are not easy, and incorrect tool implementation may provide inaccurate representations of the protocol
  - Wireshark has evolved over the years and proven itself capable of decoding all kinds of application protocols

Challenges in Protocol Header Analysis

- On the part of the analyst, there's a myriad of protocols that needs to be understood
- Without proper and in-depth knowledge of application protocols, the analysis will be severely hindered
- Writing application protocol decoders are not easy, and incorrect tool implementation may provide inaccurate representations of the protocol
  - Wireshark has evolved over the years and proven itself capable of decoding all kinds of application protocols

Challenges in Protocol Header Analysis

- On the part of the analyst, there's a myriad of protocols that needs to be understood
- Without proper and in-depth knowledge of application protocols, the analysis will be severely hindered
- Writing application protocol decoders are not easy, and incorrect tool implementation may provide inaccurate representations of the protocol
  - Wireshark has evolved over the years and proven itself capable of decoding all kinds of application protocols

Purpose of Traffic Content Analysis

These are the purposes of traffic content analysis

- Extract evidence, such as files that are transferred between two endpoints
- Determine attack methods and level of sophistication
  - For example, an adversary that uses evasion may exhibit a level of sophistication. This may also indicate targeted attacks, and the attacker is careful in executing her attacks

Full Content Data Analysis

- Payload data is useful for the following types for forensic analysis
  - Extracting files
  - Detecting covert channel
  - Finding specific packets
- Full content data analysis toolkit
  - tcpdump
  - ngrep
  - tcpxtract - able to extract files based on file header metadata from tcpdump packet capture
  - Usage: `tcpxtract -f dumpfile.pcap -c tcpxtract.conf -o output_dir`
  - chaosreader - able to reconstruct TCP/UDP and ICMP sessions
  - Usage: `chaosreader.pl dumpfile.pcap`

Filetypes

- Misconception:
  - → File type is identified by file extension. E.g, .doc is word document, .txt is text file
- Correct perception:
  - File type is identified by file metadata.
  - Metadata is structured, encoded data that describes the characteristics of information-bearing entities to aid in the identification, discovery, assessment, and management of the described entities
    - tar
    - Hex (position 257): 75 73 74 61 72
    - ASCII: ustar

File Carving

- File Carving is the process of reassembling computer files from fragments in the absence of filesystem metadata
- Makes use of knowledge of common file structures, information contained in files, and heuristics regarding how filesystems fragment data
- In other words - searching for files based on other kinds of data (such as headers and footers) rather than metadata

Tcpxtract

- A tool for extracting files from network traffic based on file signatures
- Uses a technique called data carving (extracting files based on headers and footers)
- Easy to write file signaturess
- Can be used on live data capture (realtime analysis) or tcpdump capture files (offline analysis)
- http://tcpxtract.sourceforge.net/

Challenges in Traffic Content Analysis

- Encryption
- Compression
- Volume - lots of packets to analyze. For example, a single FTP transaction can span hundreds of packets
- Time-consuming

Challenges in Traffic Content Analysis

- Encryption
- Compression
- Volume - lots of packets to analyze. For example, a single FTP transaction can span hundreds of packets
- Time-consuming

Challenges in Traffic Content Analysis

- Encryption
- Compression
- Volume - lots of packets to analyze. For example, a single FTP transaction can span hundreds of packets
- Time-consuming

Challenges in Traffic Content Analysis

- Encryption
- Compression
- Volume - lots of packets to analyze. For example, a single FTP transaction can span hundreds of packets
- Time-consuming