Network Forensic

http://webpages.sou.edu/~ackler/CF_II.Network_Forensics/index.htm

Overview

- Network forensics
- Sources of Network Data and Evidence
- Evidence Acquisition
- Protocol Analysis
- IDS
- Analysis of Network Traffic

Network Forensic

What is Forensic Network

- Network forensics is a sub-branch of digital forensics relating to the monitoring and analysis of computer network traffic for the purposes of information gathering, legal evidence, or intrusion detection

Network attack as Cybercrime

- “The Federal Bureau of Investigation (FBI) estimates that cyber crime costs more than $100 billion per year.” [2]
- Attacks can come from both inside and outside of the network.
- Not just basement hackers anymore
  - Employees
  - Business competition
  - Professional hackers for hire
  - City-states

General evidence

- Real evidence - physical objects that play a relevant role in the crime
  - Physical HHD or USB
  - Computer – box, keyboard, etc.
- Best evidence - can be produced in court
  - Recovered file
  - Bit – for – bit snapshot of network transaction
- Direct evidence – eye witness
- Circumstantial evidence – linked with other evidence to draw conclusion
  - Email signature
  - USB serial number
- Hearsay – second-hand information
  - Text file containing personal letter
- Business records – routinely generated documentation
  - Contracts and employee policies

Investigation Methodology

- OSCAR [3]
  - Obtain information
  - Strategize
  - Collect evidence
  - Analyze
  - Report

Obtain information[3]

- Incident description
- Information regarding incident discovery
- Known persons involved
- Systems and / or data known to be involved
- Actions taken by organization since discovery
- Potential legal issues
- Working time frame for investigation and resolution
- Specific goals
- Etc.

The Environment[3]

- Working business model and enforceable policies
- Potential legal issues involved with said business model and policies
- Organizational structure
- Network topology
- Possible network evidence sources
- Incident response management procedures
- Central communication systems (investigator communication and evidence repository)
- Available resources
  - Staff
  - Equipment
  - Funding
  - Time

Strategize[3]

- Understand the goals and time frame for investigation
- Organize and list resources
- Identify and document evidence sources
- Estimate value of evidence versus value of obtaining it
- Prioritize based on this estimate
- Plan of attack – both for acquisition and analysis
- Set up schedule for regular communication between investigators
- Remember that this is fluid and will most likely have to be adjusted

Collect evidence[3]

- Document, document, document
- Lawfully capture evidence
- Make cryptographically verifiable copies
- Setup secure storage of collected evidence
- Establish chain of custody
- Analyze copies only
- Use legally obtained, reputable tools
- Document every step

Analyze[3]

- Show correlation with multiple sources of evidence
- Establish a well documented timeline of activities
- Highlight and further investigate events that are potentially more relevant to incident
- Corroborate all evidence, which may require more evidence gathering
- Reevaluate initial plan of attack and make needed adjustments
- Make educated interpretations of evidence that lead to a thorough investigation, look for all possible explanations
- Build working theories that can be backed up by the evidence (this is only to ensure a thorough investigation)
    - SEPARATE YOUR INTERPRETATIONS FROM THE FACTS

Report[3]

- Every report must be:
  - Understandable by nontechnical people
  - Complete and meticulous
  - Defensible in every detail
  - Completely factual

Source of Evidence

Sources of network-based evidence

- On the Wire
- In the Air
- Switches
- Routers
- DHCP Server
- Name Servers
- Authentication Server
- Network Intrusion Detection / Prevention Systems
- Firewalls
- Web Proxies
- Application Server
- Central Log Server

On the wire

- Physical cabling carries data over the network
- Typical network cabling;
  - Copper : twisted pair or coaxial cable
  - Fiber-optic lines
- **Forensic Value:**
  - Wire tapping can provide real-time network data
  - Tap types
    - “Vampire” tap – punctures insulation and touches cables
    - Surreptitious fiber tap – bends cable and cuts sheath, exposes light signal
    - Infrastructure tap – plugs into connectors and replicates signal

In the air

- Wireless station – to – station signals
  - Radio frequency (RF)
  - Infrared (IR) – not very common
- **<u>Forensic Value:</u>**
  - Can be trivial as information is often encrypted, however valuable information can still be obtained
    - Management and controls frames are usually not encrypted
    - Access points (AP) advertise theirs names, presence and capabilities
    - Stations probes for APs and APs respond to probes
    - MAC addresses of legitimate authenticated stations
    - Volume-based statistical traffic analysis

switches

- “Switches are the glue that our hold LANs together” (Davidoff & Ham, 2012)
- Multiport bridges that physically connect network segments together
- Most networks connect switches to other switches to form complex network environments
- **<u>Forensic Value:</u>**
  - Content addressable memory (CAM) table
    - Stores mapping between physical ports and MAC addresses
  - Platform to capture and preserve network traffic
  - Configure one port to mirror traffic from other ports for capture with a packet sniffer

Routers

- Connect traffic on different subnets or networks
- Allows different addressing schemes to communicate
- MANs, WANs and GANs are all possible because of routers
- **Forensic Value:**
  - Routing tables
    - Map ports on the router to networks they connect
    - Allows path tracing
  - Can function as packet filters
  - Logging functions and flow records
  - Most widely deployed intrusion detection but also most rudimentary

DHCP Servers

- Dynamic Host Configuration Protocol
- Automatic assignment of IP addresses to LAN stations
- **Forensic Value:**
    - Investigation often begins with IP addresses
    - DHCP leases IP addresses
    - Create log of events
        - IP address
        - MAC address of requesting device
        - Time lease was provided or renewed
        - Requesting systems host name

Name Servers

- Map IP addresses to host names
- Domain Name System (DNS)
- Recursive hierarchical distributed database
- **Forensic Value:**
    - Configured to log queries
        - Connection attempts from internal to external systems
            - EX: websites, SSH servers, external mail servers
        - Corresponding times
    - Create timeline of suspect activities

Authentication servers

- Centralized authentication services
- Streamline account provisioning and audit tasks
- **Forensic Value:**
  - Logs
    - Successful and/or failed attempts
    - Brute-force password attacks
    - Suspicious login hours
    - Unusual login locations
    - Unexpected privileged logins

Network intrusion detection / prevention systems

- NIDSs and NIPSs were designed for analysis and investigation
- Monitor real time network traffic
- Detect and alert security staff of adverse events
- **Forensic Value:**
  - Provide timely information
    - In progress attacks
    - Command – and – control traffic
  - Can be possible to recover entire contents of network packets
  - More often recovery is only source and destination IP addresses, TCP/UDP ports, and event time

firewalls

- Deep packet inspection: forward, log or drop
- Based on source and destination IP, packet payloads, port numbers and encapsulation protocols
- **Forensic Value:**
    - Granular logging
    - Function as both infrastructure protection and IDSs
    - Log
        - Allowed or denied traffic
        - System configuration changes, errors and other events

Web proxies

- Two uses:
  - Improve performance by caching web pages
  - Log, inspect and filter web surfing
- **Forensic Value:**
  - Granular logs can be retained for an extended period of time
  - Visual reports of web surfing patterns according to IP addresses or usernames (Active Directory logs)
  - Analyze
    - phishing email successes
    - Inappropriate web surfing habits
    - Web –based malware
  - View end-user content in cache

Application servers

- Common types:
  - Database
  - Web
  - Email
  - Chat
  - VoIP / voicemail
- **Forensic Value:**
  - Far too many to list!

Central log server

- Combine event logs from many sources where they can be time stamped, correlated and analyzed automatically
- Can vary enormously depending on organization
- **Forensic Value:**
  - Designed to identify and respond to network security events
  - Save data if one server is compromised
  - Retain logs from routers for longer periods of time then routers offer
  - Commercial log analysis products can produce complex forensic reports and graphical representations of data

A quick protocol review

- Why know internet protocol?
    - “Attackers bend and break protocols in order to smuggle covert data, sneak past firewalls, bypass authentication, and conduct widespread denial-of-service (DoS) attacks.” (Davidoff & Ham, 2012)
- OSI model for web surfing

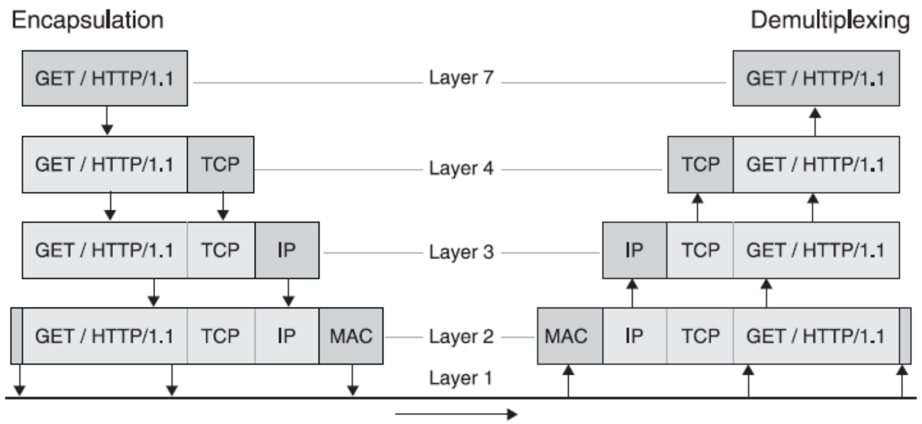


Figure 2–2. An HTTP “GET” request, shown in the framework of the OSI model.

Internet Protocol Suite review

- Forensic investigators must know TCP / IP very well, including key protocols and header fields.
- Must have a clear understanding of protocol including flow record analysis, packet analysis and web proxy dissection
- Designed to handle addressing and routing
- IP operates on layer 3 (network layer)
- Connectionless
- Unreliable
- Includes a header but no footer
- Header plus payload is called an IP packet

IPv4 vs IPv6

- 32-bit address space
- 2^{32} (approx. 4.3 billion) possible addresses

- 128-bit address space
- 2^{128} (340 undecillion possible addresses)

IPv4 Packet Header

Bits	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Bytes	0	1	2	3
0x00	Version | IHL	DSCP | ECN	Total Length	
0x04	Identification		R | D | M | Fragment Offset	
0x08	Time to Live	Protocol	Header Checksum	
0x0C	Source Address			
0x10	Destination Address			
0x14	Options			

Figure 2–3. The IPv4 packet header.

IPv6 Packet Header

Bits	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Bytes	0	1	2	3
0x00	Version | Traffic Class	Traffic Class | Flow Label	Flow Label	
0x04	Payload Length		Next Header	Hop Limit
0x08	Source Address (128 bits)			
0x0C				
0x10				
0x14				
0x18	Destination Address (128 bits)			
0x1C				
0x20				
0x24				

Figure 2–4. The IPv6 packet header.

TCp vs UDP

- Transmission Control Protocol
  - Reliable
  - Handles sequencing
  - Connection – oriented
  - Port range 0 – 65535
  - Header but no footer
  - Header plus payload – TCP segment

TCP Segment Header				
Bits	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Bytes	0	1	2	3
0x00	Source Port		Destination Port	
0x04	Sequence Number			
0x08	Acknowledgement Number			
0x0C	Length Reserved	C E U A P R S F	Window Size	
0x10	Checksum		Urgent Pointer	
0x14	Options			

Figure 2–5. TCP

- User Datagram Protocol
  - Unreliable
  - Connectionless
  - Port range 0 – 65536
  - Header but no footer
  - Header plus payload – UDP datagram

UDP Datagram Header				
Bits	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Bytes	0	1	2	3
0x00	Source Port		Destination Port	
0x04	UDP Length		UDP Checksum	

Figure 2–6. UDP

Acquiring Evidence

goal

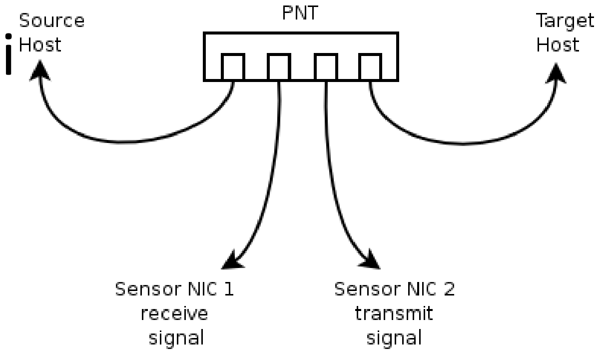
- Best possible outcome (impossible):
  - Perfect-fidelity evidence
  - Zero impact on network environment
  - Preserve evidence
- Reality:
  - Not possible to achieve a zero footprint investigation
  - Must use best practices to minimize investigative footprint
  - Verify evidence authenticity with cryptographic checksums
- Active vs. Passive
  - Passive – "… gathering forensic-quality evidence form networks without emitting data at Layer 2 and above." (Davidoff & Ham, 2012)
  - Active – "collecting evidence by interacting with workstations" (Davidoff & Ham, 2012)
  - Both techniques are used on a continuum

Physical Interception

- Capturing or sniffing packets
  - Passive packet acquisition as data is transmitted normally over the wire
- Available tools
  - Inline Network Tap
  - Vampire Taps
  - Induction Coils – not commercially available
  - Fiber Optic Taps – similar to an inline tap

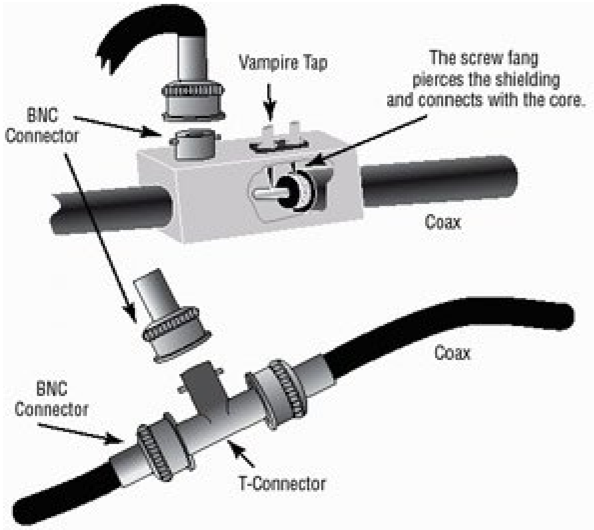
Inline Network tap

- Layer 1 device
- Inserted between two physically connected devices
  - Minor data disruption while installi
  - Potential point of failure
- Physically replicates copies to a separate port/s
- Common to have four ports
  - Two connected to inline to allow normal traffic
  - Two sniffing ports that mirror traffic (one for each direction of data flow)
- High-end taps have load-balancing for intrusion detection



Vampire tap

- Punctures the coating on the wire to physically touch the wire
  - Can break down the link
- Standard issue in a “butt kit” used by phone companies



Radio frequency

- 802.11 IEEE standard – Wi-Fi
- Signals travel through the air (shared medium)
- Stations can capture all RF traffic regardless if it is part of the link
- Although US regulates the distance RF is “allowed” to be broadcasted, directional receivers can pick up signals from many miles away
  - Importance to forensic investigator
    - Illegal WLAN access possible from a great distance
    - Investigators’ wireless links are open season for monitoring
- Wi-Fi is usually encrypted, however usually a single pre-shared key (PSK) for all stations connected to WLAN
  - Anyone with access to PSK can monitor wireless traffic of WLAN
  - Usually PSK available through IT staff

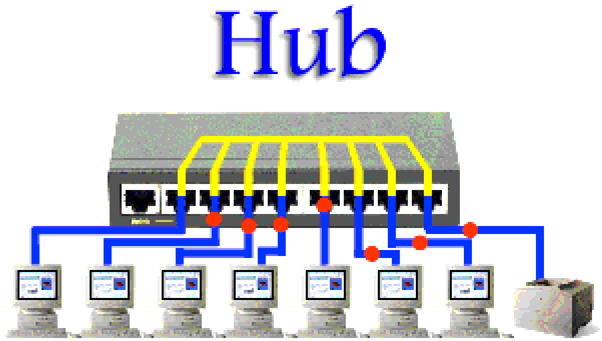
RF continued

- Lots of information even if data is encrypted:
  - SSIDs
  - WAP MAC addresses
  - Client MAN addresses
  - Sometimes full Layer 3+ packet conten

- Wi-Fi packet capture requires special hardware
  - Many standard wireless NICs do not support monitor mode
  - Commercial NICs can operate completely passively
    - Monitor wireless traffic but do not transmit any data
    - Popular choices are in a portable form factor - USB

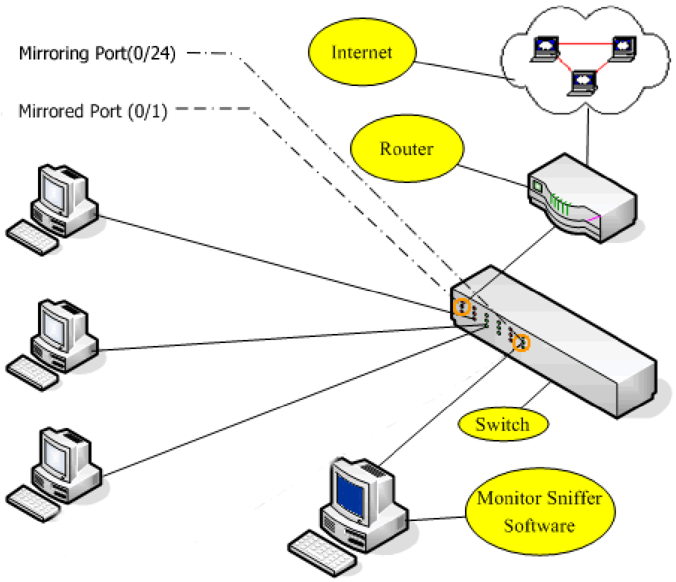
Hubs

- Dumb layer 1 device
- Transmits all packets to every port
- Allows for easy monitoring for both the good guys as well as the bad



switches

- Most common Layer 2 device, sometimes Layer 3
- CAM table keeps track of port assignment and forwards packets accordingly
- Traffic can be obtained by port mirroring
  - Configure one port to receive all of the data transmitted over a different port
  - Must have administrative ac
    - Attackers use:
      - MAC flooding
      - ARP spoofing
  - Setup is vendor specific



Traffic Acquisition software

- Libpcap
  - UNIX C library
  - Provides an API for capturing and filtering data link-layer frames
  - WinPcap
    - Based on libpcap but designed for windows
  - Most popular tools that use this library
    - Tcpdump
    - Wireshark
    - Snort
    - Nmap
    - Ngrep
  - Captures packets at Layer 2 and stores them for later analysis

TCPdump

- UNIX tool
- WinDump for Windows
- Purpose
  - Capture network traffic for later analysis
  - Capture traffic on a target segment over a period of time
- Captures bit-by-bit
- High fidelity
- Can be used with BPF to weed out traffic that is not pertinent to investigation

TCPDUMP example

- This example excludes TCP port 80 traffic from the eth0 network interface using BPF

```
# tcpdump -nni eth0 'not (tcp and port 80) '
tcpdump: verbose output suppressed , use -v or -vv for full protocol decode
listening on eth0 , link -type EN10MB (Ethernet), capture size 65535 bytes
12:49:33.631163 IP 10.30.30.20.123 > 10.30.30.255.123: NTPv4 , Broadcast ,
length 48
12:49:38.197072 IP 192.168.30.100.57699 > 192.168.30.30.514: SYSLOG local2.
notice , length: 1472
12:49:38.197319 IP 192.168.30.100.57699 > 192.168.30.30.514: SYSLOG local2.
notice , length: 1472
12:49:38.197324 IP 192.168.30.100 > 192.168.30.30: udp
12:49:38.197327 IP 192.168.30.100 > 192.168.30.30: udp
12:49:38.197568 IP 192.168.30.100.57699 > 192.168.30.30.514: SYSLOG local2.
notice , length: 1472
12:49:38.197819 IP 192.168.30.100.57699 > 192.168.30.30.514: SYSLOG local2.
notice , length: 1472
12:49:38.197825 IP 192.168.30.100 > 192.168.30.30: udp
12:49:38.197827 IP 192.168.30.100 > 192.168.30.30: udp
12:49:38.197829 IP 192.168.30.30.39879 > 10.30.30.20.53: 16147+ PTR?
100.30.168.192.in -addr.arpa. (45)
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

TCPDUMP command-line usage

tcpdump command-line usage:

```
-i   Listen on interface (eth0, en1, 2)
-n   Do not resolve addresses to names.
-r   Read packets from a pcap file
-w   Write packets to a pcap file
-s   Change the snapshot length from the default
-C   With -w, limit the capture file size, and begin a new file when it is
    exceeded
-W   With -C, limit the number of capture files created, and begin
    overwriting and rotating when necessary
-D   List available adapters (WinDump only)
```

TCPDUMP – 5 common commands

- tcpdump -i eth0 -w great_big_packet_dump.pcap
  - Listening in eth0 and writing all the packets in a single file
- tcpdump -i eth0 -s 0 -w biggest_possible_packet_dump.pcap
  - Same as above except by setting the snaplength to 0 it grabs the entire frame regardless of its size (this is not necessary in newer versions)
- tcpdump -i eth0 -s 0 -w targeted_full_packet_dump.pcap 'host 10.10.10.10'
  - Grab packets sent to or from 10.10.10.10
- tcpdump -i eth0 -s 0 -C 100 -w rolling_split_100MB_dumps.pcap
  - Grabs every frame but splits the capture into multiple files no larger than 100MB
- tcpdump -i eth0 -s 0 -w RFC3514_evil_bits.pcap 'ip[6] & 0x80 != 0'
  - Targets first byte of the IP fragmentation field, bitmask narrows it to single highest order bit "IP reserved bit" and finally packets are only stored if this value is nonzero

wireshark

- Open source GUI
- Captures – shows in real time and saves in a file
- Filters – easy filtering with many options
- Analyzes – powerful protocol analyzer
- Includes tshark
  - Command line network protocol analysis tool
  - Reads and saves files in same format
  - Ex:

```
# tshark -i eth0 -w test.pcap 'not port 22'
Capturing on eth0
235
```

- Includes dumpcap
  - Especially designed for packet capturing
  - Ex:

```
$ dumpcap -i eth0 -w test.pcap 'not port 22'
File: test.pcap
Packets: 12
Packets dropped: 0
```

Active Acquisition

- Modifies the environment – forensic investigators must minimize the impact!
- Common interfaces
  - Console
  - Secure Shell (SSH)
  - Secure Copy (SCP) and SSH File Transfer Protocol (SFTP)
  - Telnet
  - Simple Network Management Protocol (SNMP)
  - Trivial File Transfer Protocol (TFTP)
  - Web and proprietary interfaces

SSH

- Common remote access
- Replaces insecure telnet
- Encrypts authentication credentials and data
- OpenSSH – widely used implementation
    - Open source
- Command line interaction

SCP and SFTP

- Used in conjunction with SSH for secure file transfer and handling

Telnet

- Early design means limited security
  - Plaintext
  - Unencrypted credentials and data
- Sometimes it is the only option
  - Network devices have limited hardware or software
  - Not capable of upgrades to SSH
- Ex:

```
$ telnet lmgsecurity.com 80
Trying 204.11.246.1...
Connected to lmgsecurity.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: lmgsecurity.com

HTTP/1.1 200 OK
Date: Sun, 26 Jun 2011 21:39:33 GMT
Server: Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny10 with Suhosin-Patch
    mod_python/3.3.1 Python/2.5.2 mod_ssl/2.2.9 OpenSSL/0.9.8g mod_perl/2.0.4
    Perl/v5.10.0
Last-Modified: Thu, 23 Jun 2011 22:40:55 GMT
ETag: "644284-17da-4a668c728ebc0"
Accept-Ranges: bytes
Content-Length: 6106
Content-Type: text/html
```

SNMP

- "Most commonly used protocol for network device inspection and management" (Davidoff & Ham, 2012)
- Poll network devices from a central server
- Push information from remote agents to central collection point
- Used in two ways
  - Event-based alerting
  - Configuration queries
- Basic operations
  - Polling: GET, GETNEXT, GETBULK – retrieve information
  - Interrupt: TRAP, INFORM – timely notification
  - Control: SET – control configuration of remote devices

TFTP

- Transfers files between remote systems
- Transfers without authentication
- Services are small and limited, but still widespread
- UDP on port 69
- VoIP
- Firewalls
- Network devices often communicate with central servers
  - Backup configurations on routers and switches
- Forensic investigators uses
  - Export files form network devices not supported by SCP or SFTP

Web and proprietary interfaces

- New network devices come with web-based management
  - Access configuration menus
  - Event logs
  - Other common data
- Typically HTTP
- Forensic challenge
  - GUI inhibits logging
  - Best fallback is often screenshots and notes

Inspection without access

- Port scanning
  - Nmap
    - Will generate network traffic
    - Can modify the state of the target device
- Vulnerability scanning
  - Provide clues as to how breach or compromise may have occurred
  - Generate network traffic
  - Can modify the state of target device
  - Can crash target device

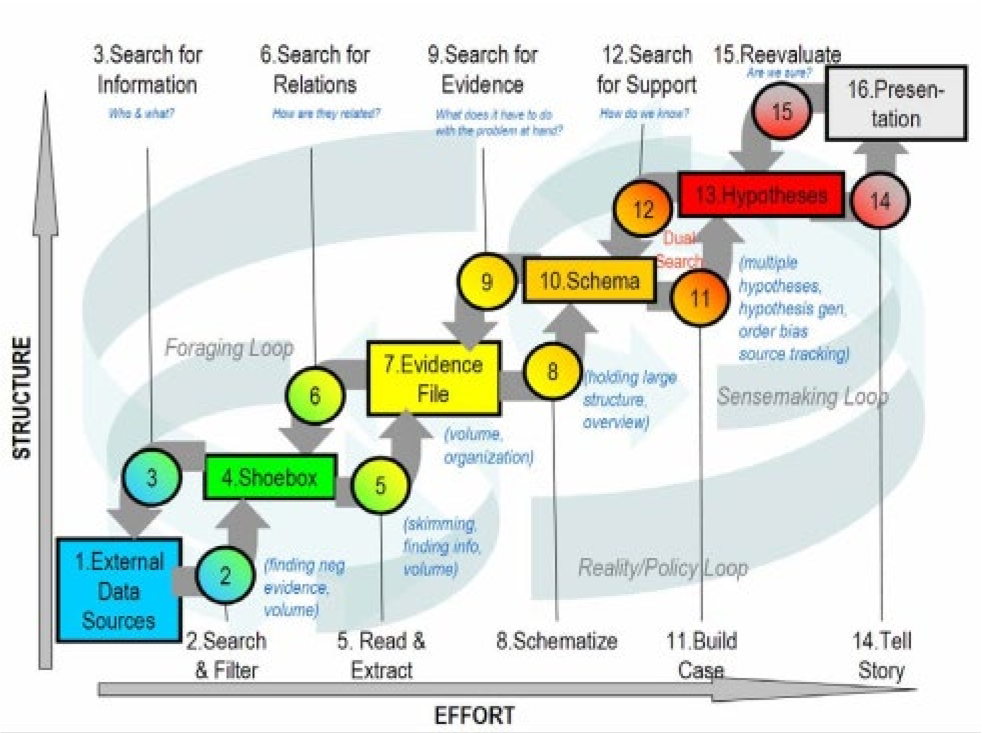
Strategy

- Refrain from rebooting or powering devices down
  - Volatile data lost in reboot
    - Ex: ARP tables, current state of devices
  - May modify persistent logfiles
- Connect via console instead of remotely over network
- Record system time
  - Check time skew
- Collect evidence according to volatility
  - When all else is equal go with data most likely to change or be lost
- Document all activities
  - Record commands – using “screen” or “script”
  - Important to make a record of all activities – mistakes and all
  - Screenshots of all GUI related activities

Protocol Analysis

Protocol analysis techniques

- Protocol Identification
- Protocol Decoding
- Exporting Fields

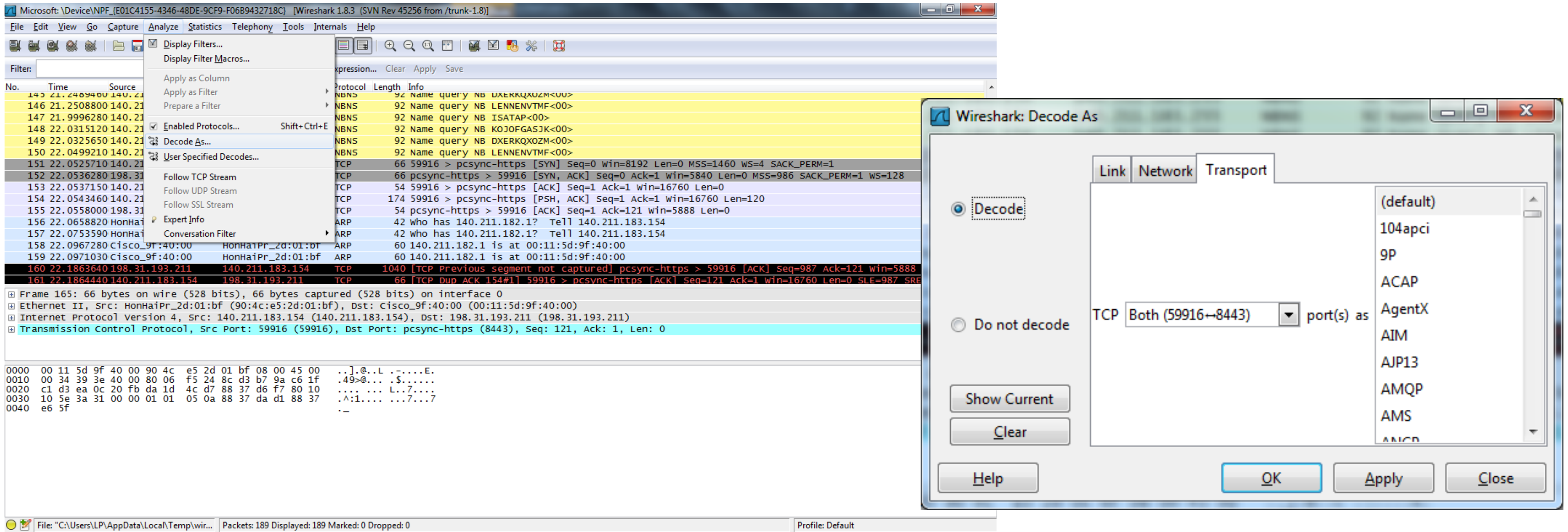


Protocol Identification

- Look for common binary/hex/ASCII values that are associated with specific protocols
  - Ex: 0x4500 marks the beginning of an IPv4 packet
- Use information in the encapsulating protocol
  - Ex: Byte 9 of the IP header indicates protocol, 0x06 corresponds with TCP
- Use port numbers for TCP/UDP
  - Ex: port 443 indicates TLS/SSL, check to see if packet is indeed encrypted
- Analyze the function of the src or dst server
  - Use IP address and do a WHOIS lookup
- Look for recognizable protocol structures
  - Refer to RFCs

Protocol decoding

- A way to interpret frame data based on known frame structure
- To use specific protocol specs
  - Use publically available automated decoders and tools
  - Manually decode traffic with publically available documentation
  - Write you own decoder



Packet analysis

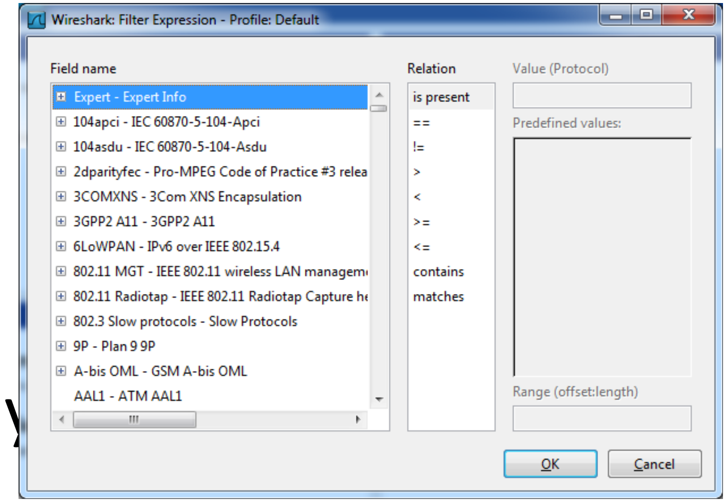
- Defined
  - “Packet Analysis—Examination of contents and/or metadata of one or more packets. Packet analysis is typically conducted in order to identify packets of interest and develop a strategy for flow analysis and content reconstruction.” (Davidoff & Ham, 2012)

Packet analysis tools

- Wireshark And Tshark Display Filters
- Ngreg
- Hex Editors

Wireshark and tshark display filters

- Over 105,000 display filters
- Supports open plugin architecture
  - Build your own protocol parser
- “Expressions” button to build a filter of y



- Tshark uses –R for filters
  - Example:
    - $ tshark -r capturefile.pcap -R "ip.src ==192.168.1.158 && ip.dst ==10.1.1.10“ 28. “

ngrep

- Looks for packets based on particular string, binary sequences or patterns within the packet
- Recognizes common protocols: IP, TCP, UDP, and ICMP
- No flow reconstruction
  - Will not detect if data spans multiple packets
  - Detects matching packet not matching flow
- Example:
  - $ ngrep -I capturefile.pcap "string to search for“
  - $ ngrep -I capturefile.pcap "string to search for" 'src host 192.168.1.20 and dst port 80'

Hex editors

- View and manipulate raw bits of data
- Indispensable for isolation of specific packet fragments and file carving
- Sometimes regular tools are not equipped to handle data
  - Example:
    - Loki tunneling protocol is often not recognized by tools like Wireshark
    - Most tools will not see inside compressed files
- Bless, Winhex, FTK Imager

Packet analysis techniques

- Pattern Matching
- Parsing Protocol Fields
- Packet Filtering

Pattern matching

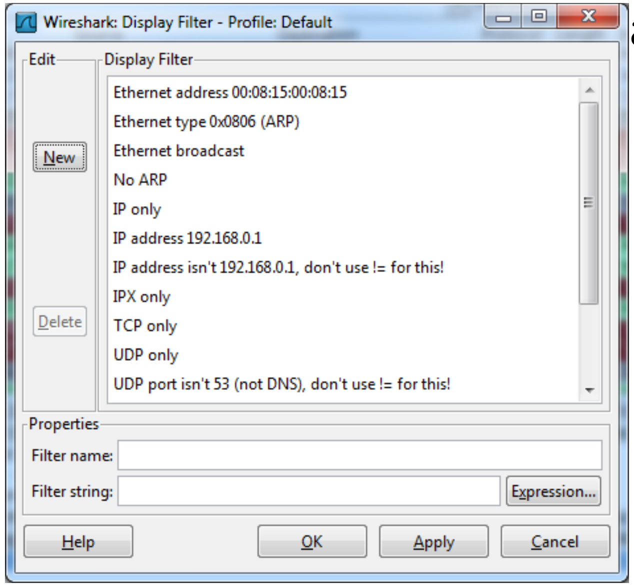
- “dirty word search”
  - List of strings, names, patterns that are related to suspect activity
- ngrep is the best tool for these searches
  - Example:
    - $ ngrep -I evidence01.pcap ‘words|search|for‘

Parsing Protocol Fields

- Application of extracting the contents of protocol fields within packets of interest.
- Example:
  - $ tshark -r evidence01.pcap -d tcp.port ==443 , aim -T fields -n -e "aim. messageblock.message“
- Good tshark reference
  - http://www.packetlevel.ch/html/tshark/tshark.html

Packet filtering

- “…the art of separating packets based on the values of fields in protocol metadata or payload.” (Davidoff & Ham, 2012)
- Use tcpdump with a BPF filter to dump out suspicious converstions
    - Example using IP addresses
        - $ tcpdump -s 0 -r evidence01.pcap -w evidence01 -talkers.pcap 'host 64.12.24.50 and host 192.168.1.158 ‘ ading from file evidence01.pcap , link -type EN10MB (Ethe
- Use Wireshark



IDS

NIDS/NIPS & HIDS/HIPS

- Intrusion detection, prevention and analysis
- HID(P)S – host based intrusion detection(prevention) systems
- NID(P)S – network based intrusion(detection) systems
  - Functionality
  - Modes of detection
  - Types of NIDS/NIPS
  - Evidence acquisition
  - Packet logging
  - Systems – Snort ($_{*\ *}$)

Functionality

- IDS's are rule based
- Issues alerts
- Configured to capture suspicious packet sequences
- Sniffing
  - Multiple layer inspection
  - Protocol awareness
  - Protocol reassembly
- In a NIPS processing time is critical
- In a NIDS offline analysis and alerting is tolerable
  - Deep packet analysis is possible
- Some sort of normalization of packet contents may be required

Modes of Detection

- Signature based analysis
- Protocol analysis
- Behavioral analysis

Types of IDSs

- Commercial
  - Check Point IPS-1
    - http://www.checkpoint.com/products/ips-software-blade/
  - Cisco IPS
    - http://www.cisco.com/web/services/portfolio/product-technical-support/intrusion-prevention-ips/index.html
  - Enterasys IPS
    - https://www.enterasys.com/company/literature/ips-ds.pdf
  - Tipping Point IPS
    - http://h17007.www1.hp.com/us/en/whatsnew/040511-1.aspx

Types of IDSs

- Open Source
  - Snort
    - Sourcefire
    - Just bought by Cisco
    - Get it soon before they screw it up
  - Bro

Evidence Acquisition

- Types of evidence
  - Configuration
    - The configuration of each sensor is important
    - The location of each sensor within the network is also important
    - Running configuration is important
    - The rule set is important
  - Alert data
  - Packet header info
    - Flow data
  - Packet payloads
  - Correlation across multiple sensors

Configuration Files

- Alerts can be different on different sensors
  - The configuration of each sensor is important
  - The location of each sensor within the network is also important
  - Running configuration is important
  - The rule set is important

Comprehensive Logging

- All Packets all the time
  - Massive amounts of storage space
  - Difficult to archive except for NSA
  - Lots of CPU
  - Large risk
- Perhaps filter
- Only flow data

SNORT

- Most widely used IDS
- Open-source code (???)
- Open rule language
- Extremely versatile
- Commercial Support
- Community/commercial business model

- Cisco's impact is a big question
- Maybe it will be forked?

Architecture

- Uses libpcap to capture packet
- Passes through 4 preprocessors
  - Layer 3: reassembles frragments
  - Layer 4: reassembles streams
  - Layer 5: reassembles sessions
  - Layer 6: reassembles transactions
- Can issue an alert at any layer
- After reassembly and anomaly detection
  - Information is handed off to rule engine
- Handed to alerting engine
- Subsequent related packets can be marked for capture

Configuration

- Location of snort files
  - /etc/snort/snort.conf – global SNORT values are stored
    - Various network addresses
    - Location of rules
    - Location of services
  - /etc/snort/rules
    - Home of the actual rules
  - /var/log/snort
    - Directory of the SNORT logs
    - Can be very large

Rules

- The basis for logging or not logging a packet
- Can be more that one line long – now
  - Each line to be continued must be terminated with a ' \‘

»That is “space \”

- Generic syntax

rule_header (rule_options)

•Rule header

»Action, addresses, ports, masks

•Rule options

»Messages, what to look for, where to look

Simple Rule

- Snort rule example
- alert tcp any any -> 192.168.1.0/24 111 \

Action Protocol Src IP mask Source Port Des IP mask Dest Port

(content:"|00 01 86 a5|"; msg: "mountd access";)

content: what to match in the packet

msg: log message heading

Key Words

- Include

  include /etc/snort/rules/ping.rules

- Variables

  var HOME_NET 192.16.13.0/24

  var RULE_PATH /etc/snort/rules

  include $RULE_PATH/ping.rules

- Config

  config reference: bugtraq ttp://www.securityfocus.com/bid

Rule Actions/Types

Field 1

- Alert, log, pass
    - Alert – generates an alert message and then logs the packet
    - Log – logs the packet
    - Pass – ignores the packet
- Activate, dynamic
    - Activate – sends an alert and then turns on a dynamic rule
    - Dynamic – idle until activated and then acts as a log rule
- User defined rule types

Protocols

Field 2

- tcp, udp, icmp, ip
  - Todate
- arp, igrp, gre, ospf, rip, etc.
  - The distant future

Addresses

Fields 3 & 5

- Usual dotted decimal notation with mask indicated
  - 192.16.13.0/24
- Dereferenced variable
  - $HOME_NET
- Keyword any
- List [192.16.13.0/24,10.1.1.0/24]
- Negation !192.16.13.1

Ports

Fields 4 & 6

- Numerical, “any”
    - 80, 21, 23, etc.
    - 100:1024 - ports 100 through 1024
    - :600 - ports 0 through 600
    - 500: - ports greater than or equal to 500
- Typical address/port fields

  !192.16.13.0/24 any -> 192.16.13.0/24 111

Rule Options

- Key words:

- msg	prints a message in the log
- ttl	test the ip header's ttl value
- tos	test the tos field
- id	test the ip header's id field
- fragbits	test the fragmentation bits
- dsize	test the packet's payload size
- flags	test tcp flags
- seq	test the sequence number for a specific value
- ack	test the ack bit for set or clear
- itype	test icmp type
- sid	snort rule for id
- rev	rule revision number
- ip_proto	ip header's protocol number
- reference	external attack

Options

Examples

- msg
  - Puts a message in the log record to identify the snort rule

    msg: “SYN packet malformed”;
- ttl
  - Tests for a specific ttl value

    ttl: “127”;
- dsize
  - Tests for a specific size of the packet, >, <, <>

    dsize: “400<>500”;

Options

Examples cont'd

- fragbits
  - Tests for configuration of the IP dgram frag bits

    RB, MF, DF (reserved bit, more frags bit, do not frag bit)

    modifiers: + all have to match

    * any have to match

    ! match if bits are not set

    fragbits: R+;

Options
Examples cont'd

- content
  - Tests for specific content within the payload packet

    Binary data enclosed by “| ... |”

    ASCII data enclosed by “ ... “

    ! tests that the content does not contain the string

    content: “|90CB C0FF FFF|/bin/sh”;

    content: !”GET”;

Options

Examples cont'd

- offset
  - Dictates the starting position of the content search

    offset: 3;
- depth
  - Dictates the maximum depth of the content search

    depth: 22;
- nocase
  - Content search is not case sensitive

    nocase;

Options

Examples cont'd

- flags
  - Tests for TCP flags for a match

    F, S, R, P, A, U, 2, 1, 0

    1 & 2 are the reserved bits in the flag octet

    0 no flag is set

    ! tests that the content does not contain the string

    Modifiers: + all have to match

* any have to match

! match if bits are not set

flags: SF;

Options

Examples cont'd

- ip_proto
  - Checks the IP Protocol field, permissible are in /etc/protocols

    ip_proto: 6;

- # /etc/protocols:
- # $Id: protocols,v 1.3 2001/07/07 07:07:15 nalin Exp $
- #
- # Internet (IP) protocols
- #
- #   from: @(#)protocols   5.1 (Berkeley) 4/17/89
- #
- # Updated for NetBSD based on RFC 1340, Assigned Numbers (July 1992).
- #
- # See also http://www.iana.org/assignments/protocol-numbers

- ip   0   IP   # internet protocol, pseudo protocol number
- #hopopt   0   HOPOPT   # hop-by-hop options for ipv6
- icmp   1   ICMP   # internet control message protocol
- igmp   2   IGMP   # internet group management protocol
- ggp   3   GGP   # gateway-gateway protocol
- ipencap   4   IP-ENCAP   # IP encapsulated in IP (officially ``IP'')
- st   5   ST   # ST datagram mode
- tcp   6   TCP   # transmission control protocol
- cbt   7   CBT   # CBT, Tony Ballardie <A.Ballardie@cs.ucl.ac.uk>
- egp   8   EGP   # exterior gateway protocol
- igp   9   IGP   # any private interior gateway (Cisco: for IGRP)

Options

Examples cont'd

- classtype
  - Categorizes snort detects into attack classes

    classtype: <class name>;

    Listed in classification:config

    classtype: misc-attack;

Options

Examples cont'd

- itype
  - Checks the value of the ICMP type field

    itype: 0;
- icode
  - Checks the value of the ICMP code field

    icode: 8;

Options

Examples cont'd

- reference
  - References to external attack identification systems

    Bugtrack, CVE, Arachnids McAfee, url

    reference: <id-system>,<id>

    reference: arachNIDS,IDS287; reference: bugtraq,1387;

Options

Examples cont'd

- flow
  - Used with TCP stream reassembly, applies to certain directions
  - Applies to either client or server

    to_client - triggers on server responses

    to_server – triggers on client requests

    from_client – triggers on client requests

    from_server – triggers on server responses

    established – triggers only on established TCP connections

    flow: from_server;

The End…..