Network Forensics Introduction
Typical Attack Path
Networking Basics

• Communications.
• Two systems communicating with one another, typically over a local network (LAN) or the Internet.
• Variety of messaging protocols, both stateful and stateless depending on the type of message.
Networking Basics

- Encapsulation
- OSI Model
- Encryption
- Compression
TCP / UDP Refresher

• Transport-layer protocols.
• Learn your ports.
• Typically the most interesting forensically because these are the rough level of encapsulation where we’re watching connections between systems.
• Transmission Control Protocol
  — Stateful
  — Examples include: HTTP (web), SMTP/IMAP (mail), DNS (sometimes)
• User Datagram Protocol
  — Stateless
  — Examples include: DNS (most of the time)
Network Forensics is Hard

- Encryption
- Compression
- Volume of data.
- Amount of noise.
- Encryption.
Protocol Definition

• Request for Comments (RFC)
  – Internet Engineering Task Force (IETF)
  – Document that tracks technical protocol specifications.
  – Process / context lives in RFCs

• Key RFCs for networking:
  – RFC 1918 (private networking)
  – RFC 2616 / 7230 (HTTP 1.1)
  – RFC 7540 (HTTP v2)
IP Layer – How it Works

- IP address (192.168.8.8)
  - Source
  - Destination
- IPv4
  - 4 digits 0-255
- IPv6
  - 8 groups of 4 hex digits
  - More structured than IPv4
- NAT
  - Internal / external IP scheme
  - Internally will see internal IPs
  - Externally will see single gateway IP
Subnetting

• CIDR Notation
  – “/”
  – Uses a subnet mask

• Subnets
  – Separation / segmentation of IP networks

• 192.168.1.0-256 /24 (256)
• /16 (65,536)
• /8 (16,777,216)
IP Layer – Key Information

- **Loopback**
  - 127.0.0.1

- **All addresses on the machine**
  - 0.0.0.0

- **RFC 1918**
  - 10.0.0.0-10.255.255.255 (10.0.0.0/8)
  - 172.16.0.0/12
  - 192.168.0.0/16
A Normal Day in Networkland

• Type a URL (domain) + hit enter
• DNS request for domain.com
• DNS response for domain.com
• HTTP GET request for domain.com
• HTTP 200 response with content for domain.com
Capturing Network Forensics Data

- Capturing network data either requires a dedicated (and pre-positioned) network tap.
- A network allows for a copy of all traffic coming and going (RX and TX) to be sent to an additional interface.
- A capture interface can be leveraged to get access to process or capture traffic.
Network Forensic Capture Cont’d

• **Pros:**
  – Full capture of everything.
  – Can include files, non-standard protocols, and a lot more.

• **Cons:**
  – Typically have to decrypt in-line / MITM traffic.
  – Newer TLS versions are making “passive” decryption difficult.
  – Encryption not always possible (TLS1.3)
  – Harder and harder as people move to cloud environments.
Network Metadata Capture

• Capture is becoming less and less feasible due to data transmission and storage limitations.
  - 100 MBPS x 7d = 7.56TB
  - 10 GBPS x 7d = 756TB
  - 10 GBPS x 30d = 3.26PB

• Pros
  - Fast
  - Low storage requirements

• Cons
  - Processing overhead is also a challenge without hardware offloading and specialized drivers (like AF_PACKET which allow raw packet access).
Network metadata by environment

- **Cloud**
  - Typically netflow data, e.g., AWS, VPC Flow logs

- **On-premise**
  - Alert metadata: Suricata (most popular), other Network Intrusion Detection (NIDS) tools
  - Flow data: typically collected by a netflow collector
  - Network Security Monitoring (NSM) metadata: typically collects protocol metadata for some or all protocols
Core Network Forensics Goals

• Understand the nature of communication.
  – Who is communication.
  – About what.
  – With whom (source / destination pairs).

• Key investigative uses:
  – Bookend an investigation.
  – Confirm key events at network layer.
Common Protocols - DNS

• Domain Name System – TCP/UDP [53]
  — Always kind of broken.
  — Mapping names to IP addresses.
• DNS Query
  — Given a DNS name -> what IP?
• DNS Response
  — IP <=> Domain mapping.
• DNS caching
Common Protocols - Mail

- Simple Mail Transfer Protocol (SMTP) [TCP 25, 587, 465]
  - “secure” and “non-secure” ports.
  - “Hello and send” protocol
  - Connects to server -> Sends Content

- Internet Message Access Protocol (IMAP) [TCP 143, 993]
  - Folder support
  - Multi-part (large) messages

- Post Office Protocol (POP3) [TCP 110, 995]
Common Protocols - Web

- Hypertext Transfer Protocol (HTTP) [TCP 80,443]
  - “secure” and “non-secure” ports.
- Request and response protocol
- HTTP Request
  - Given a Uniform Resource Indicator (URI) -> get me the content.
  - Multiple verbs (GET / PUT / POST)
  - GET – variables are in the URI
  - POST – variables and data are a part of the request (commonly not logged)
Common Protocols – Web Cont’d

• Hypertext Transfer Protocol (HTTP) [TCP 80, 443]
  – “secure” and “non-secure” ports.
• Request and response protocol
• HTTP Response Codes:
  – 2XX -> Good
  – 3XX -> Redirect
  – 4XX -> Client Error
  – 5XX -> Server Error
• 200 Good
• 404 Not Found
• RFC 2616 // Wikipedia for additional details
Applied Network Forensics
Networking Implementation

• Network layers are implemented by different components.
• From network interface drivers -> OS Networking stack -> Application
Fingerprinting

• Fingerprinting is where we use specific identifying characteristics in evidence (in this case network data) to identify a system or operating system.

• We can use this technique to identify operating systems and (sometimes) installed software on a system.
Time to Live

- TTL = Time to Live
- IP-layer component
- Packets circulate / have a lifetime
- TTL sets the max lifetime of a packet (IP)
Fingerprinting Operating Systems

- **Windows TTL**
  - TTL = 128 (number of router hops)
- **Specific update / OS services**
  - Windows Update
  - Telemetry
    - v10.events.data.microsoft.com
- **Auth / Cross-communications**
  - DCERPC (445 TCP)
  - EPMAPPER (135 TCP)

- **Linux TTL**
  - TTL = 64
- **Specific update / OS services**
  - (distro mirrors)
- **Auth / Cross-communications**
  - SSH (22 TCP)
Fingerprinting Applications

• **User agents**
  - Chrome
    - dl.google.com
  - Firefox
    - aus<X>.mozilla.org

• **Productivity**
  - Office
    - nexusrules.officeapps.live.com
  - Gsuite
    - Googledrive.com
Traffic Analysis

• Brim
  – Inputs:
    • Raw Packet Capture (PCAP)
    • Logs (binary or non-binary formats)
      – Binary: ZNG
      – Non-binary: CSV
Traffic Analysis

• Brim
  – Outputs:
    • Metadata
      – Bro / Zeek
    • Alert data
      – Suricata (Emerging Threats)
Bro/Zeek

- Used to be called “bro”
- Now called “zeek”
- Takes packet data and produces logs
- Many logs are easy to figure out:
  - Protocol based...
  - http: web traffic
  - conn: netflow
- Some are not
  - “weird”: things Bro thinks are weird
  - “files”: file log generated from traffic (there’s no “file” protocol)
Working with Brim

• Start with built-in queries
• Then query traffic by protocol:
  — _path==“dns”
  — _path==“http”
  — …
• Then summarize using count() 
• Then drill-down with “pivot to logs”
Brim Query Language

- "splunk like" uses "|"
- Select > function > present
  - _path=="dns" | count() by query | sort -r
  - "count()" = GROUP BY
  - "sort" = sort by amount
- May want to roll / unroll to use "pivot to logs" and other features. Certain aggregations will break features.
Summarizing Traffic

• Start with built-in bro logs
• Drill into protocols
  – _path=="dns" | count() by query
• Drill into alerts
  – event_type=="alert" | count()
    by alert.severity,alert.category | sort count
Building a Traffic Summary

• Who’s talking?
  – Source IP (what is it)
  – Destination IP (what is it)

• What’s the nature of the communication?
Example Traffic Summary

- DNS
  - Clients
  - Local DNS Server
  - Standard / non-standard traffic
    - Queries and responses
    - (ex: A record requests / responses)
  - No non-standard activity observed
    - Look for outliers (query type / etc)
Example Traffic Summary

- **SMTP**
  - Clients (IP / OS / etc)
  - List of email servers
  - Standard / non-standard traffic
    - Ports / protocols / encrypted
    - `<normal|not normal>` to see # of servers / servers contacted by `<workstation|server>`
    - Mail client observed
  - No non-standard activity observed
    - Look for outliers / total activity
    - # of servers / etc.
Is it Bad / Interesting

- Lots of approaches – top 3 are:
  - Signature-based matches
    - Typically suricata alerts
  - Outlier analysis
    - Looking for very frequent (beaconing)
    - Looking for very rare (IP / malware download)
  - Key events
    - Binary download
Working with Wireshark

- Wireshark logo

- This launches wireshark:
TCP Stream Extraction

- Follow TCP Stream
TCP Stream Extraction

- Follow TCP Stream
Files in BRIM

- Pulls from Bro/Zeek “file” log
- Gives you a hash
Files?

- Sometimes we have full files, sometimes we have metadata.
- In the case of metadata (typically a hash value) – we can go to Virustotal to find out more about the file.
Virustotal A/V Results

- Among other things, VT aggregates AV Engines.
- Not always perfect but can be a good barometer / starting point to see if a file is malicious.

# of AV Engines that detect as malicious

Type of file

AV Detection names, can help to ID malware.
Extracting Files

• PCAP is a binary file format – essentially file extraction can be performed with 2 methods:
  – Bruteforce file headers
  – Look for “file” sections in protocols (mail attachments, etc.)

• NetworkMiner will do this for us, however, there are other options if we’re not getting what we need.
Using NetworkMiner

• NetworkMiner – Free Edition
  — Enables us to load a PCAP
  — Extract Files
• Start by loading the PCAP
• Then will list and can extract files.