Network Data Capture

Digital Forensics

Network Activity Investigations

- Network forensics focuses on network activity
- It can be used as part of an investigation to provide supporting data for documenting online activities
- The other main use of it is to investigate malware
- Network Forensic Analysis Tools (NFATs) are used to identify activity of interest on the network

NFAT Goals

- To be useful, NFATs need to do at least 3 things:
- Capture network traffic in full, reliably, and with integrity
- Analyze traffic captures to find what the investigator is looking for or deny its existence
- Discover interesting things and bring them to the examiner’s attention even if they weren’t explicitly requested

NFAT Capture Component

- For the NFAT to capture useful traffic, it must have access to that traffic
- The network architecture must be understood well enough to know where the NFAT will see the traffic of interest without interfering with it or altering it
- Repeating captures to verify them is not always an option, so process integrity is mandatory
- Captures can generate huge volumes of data quickly

Capture Host Considerations

- Are you trying to capture ingress and egress traffic, or application traffic specific to a particular host?
- If the target traffic is encrypted, can you place the NFAT where it is able to see a decrypted version of the data?
- Is the transaction awareness sufficient, or do you require the transaction content?
- Can you tap into the traffic flow using infrastructure equipment, or are you dealing with consumer grade gear?
- Does the capture host need to be undetectable?

Capture Data Considerations

- Where will the capture be stored?
  - You don’t want the storage activity to create network traffic if it can be avoided
- How will the capture be secured?
  - It may contain sensitive information
- How will the capture data be filed?
  - Automatically splitting files is usually necessary
- Software and options to provide highest capture performance is necessary (e.g. tcpdump with a filter, without name lookups, writing to an SSD)

Capture Performance Considerations

- Modern networks have the ability to sustain the transfer of very large amounts of data
- WAN connections up to gigabit speeds are becoming more common
- Capture hosts cannot discard packets when busy, and cannot wait for storage of the capture while capturing

Capture Performance Considerations

- Capture hosts are voyeurs, not participants, they cannot request retransmissions
- The amount of fast, low latency storage available for wire-speed captures will limit the timespan of a capture
- Filtering can be helpful, but limits the ability to examine anything that was not fully identified prior to the capture

tcpdump

- tcpdump is a very popular packet capture tool (tcpdump.org)
- It is available for all major platforms and is open source
- It has been around for a long time and is a credible tool for this use, it is often used as part of a larger software toolset
- It does not provide much sophistication in storage management for captures although it does provide for splitting and circular storage control

tcpdump

- tcpdump can live display traffic, save traffic to a file, or display traffic from a saved capture file or files
- Saves can be done in pcap format and lots of software can read and write this format
- tcpdump options and filters are easily looked up on the man page and cheat sheets (e.g. packetlife.net tcpdump cheat sheet)
- Examples:
  - tcpdump -i en0 -w en0.pcap
  - tcpdump -r en0.pcap host 192.168.2.1 -X

Wireshark

- Wireshark (wireshark.org) is another very popular tool
- Wireshark was called ethereal, the name was changed to be more cool
- Provides complex filtering capabilities for packets
- GUI design is intended to support live analysis, but can also save captures - very similar to tcpdump, CLI version is called tshark
  - tshark -i en0 -w wireshark.pcap

Snort

- Snort is a full featured intrusion detection software suite
- Included in snort is a packet capture capability
- Traffic analysis and filtering in snort is designed to support detection of specific types of network traffic, but is also completely customizable
- Snort can do deep packet inspection to build views of transactions between hosts at multiple levels of the TCP/IP stack; it has robust analysis capabilities
- Snort can do packet capture without running any graphical tools, similarly to tshark, but with the added sophistication and complexity of the snort ruleset to control the capture and analyze the results
- See https://www.safaribooksonline.com/, Chapter 1 section 18 of the Snort Cookbook, for a brief writeup on using snort for packet capture

Tcpflow

- Tcpflow is a packet capture tool designed to make it easier to analyze traffic flows
- Tcpflow stores captured flows in separate files, one for each direction of each flow, and only the flow data is stored in the files
- Tcpflow can make flow files from existing pcap dump files
- This can make it easier to identify the transactions of interest by searching to find the flows separately from analyzing them
  - tcpflow -a -i en0 -o tcpflowdata -Fk

Capture Appliances

- Many companies make packet capture devices you can add to your network
- They typically provide capture and analysis functions in a pre-configured machine running on hardware designed for high-speed packet capture
- Kali, SIFT, etc. provide all the capture tools you need, but you have to build the box and configure/update the tools - Snort is no fun to properly configure

Network Storage Capture

- When doing a forensic investigation that includes data stored on a network-based storage device or in the cloud, different or complementary approaches may be required
- Network storage capture is a very important part of modern investigations and does not necessarily use the same skills as network activity capture or local storage capture
- There are multiple approaches to capturing data stored in the network or cloud, they tend to focus on getting copies of files, not imaging devices or capturing traffic
- Identifying users to associate with network or cloud storage access may require matching network traffic flows to filenames and timestamps - cloud storage protocols may make this impractical

Capturing Network Storage

- If the files are on shared network storage, attaching an investigative station to the network and mounting the storage read-only is an option, although this may cause file timestamp updates depending on the file sharing service used
- If the files are on private network storage such as a SAN iSCSI device, the storage host can make a copy of the files, or of the entire device which can then be accessed by an investigative station
- If the files are on private storage where storage host access is not possible, capture the data transactions in flight using a packet capture agent on the client's switch, router, or a proxy - captures of data transactions in flight may be made easier using tcpflow, but you can always do bulk captures if that is the only option and analyze your captures later
- If adding sniffing devices is not an option, and the storage host is not available for our purposes, or the stored information is encrypted with login-based keys, then the capture may have to be done on the client system either by using an agent or with the client user's co-operation