Network Capture Analysis

Introduction
Image Capture
Microsoft Filesystems
Linux Filesystems
Evidence Analysis
Live Forensics
Network Data Capture
Network Capture Analysis
Data Forensics
Investigation Planning and Process
Network Device Forensics

Digital Forensics

Begin With Capture

- Use a simple, high performance tool to capture traffic, such as tcpdump
- Gathering more is better than gathering less unless you have restrictions on what you are permitted to capture
- Capturing from multiple points may be useful if you are unsure where the suspect traffic may originate or propagate

ngrep

- Ngrep runs grep on reconstructed communications from a capture, can also run in live capture mode
- Useful for triage to determine what kinds of things are or are not in the capture
- Very broad application, man page gives lots of good info on useful options
- Matching packets can be written to a new file, allowing splitting of data into multiple files, each containing something you searched for
- Searching using IP address can show all communications to/from a particular host
- General form:

ngrep -I capfile regex [filter expression]

Wireshark

- Familiar tool for students in this program, you may not be as familiar with tshark, the cli version of the tool
- Once you determine a capture file has relevant communications, Wireshark can be used to fill in the details, and examine lower-level packet content such as Mac addresses
- Use higher level tools such as the packet stats summary to identify what kinds of traffic are present in a capture (e.g. tshark -z io,phs -q)

tcpflow

- Useful if the details of the packet transfers aren't important, but the content of communications is important
- Review command options to capture all data that is going to be useful in the investigation, in particular -a, -e, -o, -r, and -F (e.g. tcpflow -a -Fk -o outdir)
- Use the report files to add timestamps and sequencing to your forensic analysis

splitcap

- splitcap is similar to tcpflow, in that it splits a packet capture into multiple files, based on criteria you specify
- splitcap saves the split flows as pcap files instead of content files
- Useful for breaking up large captures into manageable chunks for more detailed analysis
- e.g. splitcap -r capture.pcap -o outdir

tcpstat

- Similar to netstat and can summarize traffic, but can use a pcap file as a data source
- Useful for identifying what can be explored in a capture
- Can be helpful by showing traffic volume changes over time in a capture to find unusual activity, formatting output is pretty much required to do useful things

```
tcpstat -r capture.pcap -o "Time: %r. \tbps: %b\tpps: %p\tARP: %A. \tTCP: %T.  \tUDP: %U.  \tSizes: %m-%M\n"
```

Network Miner

- Another GUI to extract data flows from packet captures
- Recognizes some traffic content that Wireshark doesn’t automatically extract
- Has search capabilities and can export files contained within flows
- Free limited version available, but full version without limits is not free

Network Investigation Example

- From Hack3rcon 2016
- https://youtu.be/yRyi_RxXd-M?t=2726 (35 minutes of the over 2 hour video are for the example investigation)
- Additional practice resources: http://netresec.com - also provider of splitcap, networkminer, and many other excellent tools, as well as pcap files containing various network activities
- Many tools that are distributed for use on Windows can be run on Linux using mono or wine, see networkminer example on netresec website