pcapFS
Mounting Network Data for On-the-Fly Analysis

Fraunhofer Institute for Communication, Information Processing and Ergonomics

OSDFCon 2018
October 17th

Jan-Niclas Hilgert*
jan-niclas.hilgert@fkie.fraunhofer.de

Martin Lambertz
martin.lambertz@fkie.fraunhofer.de
State-of-the-art network forensics

- Wireshark is great, but...
State-of-the-art network forensics

- Wireshark is great, but

1. Usability

Packets: 791615
State-of-the-art network forensics

- Wireshark is great, but
  1. Usability
  2. Performance
State-of-the-art network forensics

- Wireshark is great, but
  1. Usability
  2. Performance
  3. Resources
State-of-the-art network forensics

- Wireshark is great, but
  1. Usability
  2. Performance
  3. Resources

- How else can you access a pcap?
Idea

- **File systems** organize unstructured data and make them available to the user
  - Create a file system for pcaps
Idea

- **File systems** organize unstructured data and make them available to the user
  - Create a file system for pcaps
- Create a structure, which can be used when accessing the same network capture again
  - Create an **index file** keeping track of the files in the file system
Idea

- **File systems** organize unstructured data and make them available to the user
  - Create a file system for pcaps

- Create a structure, which can be used when accessing the same network capture again
  - Create an *index file* keeping track of the files in the file system

- Extracting data in order to process it creates unnecessary overhead
  - Point directly into the data in the pcap
Concept

TCP file

UDP file

pcap
Concept

TCP file

UDP file

TCP and UDP files point directly into the pcap
Application protocols can then point into the TCP and UDP files.
Other protocols add new virtual layers in between
An index file is stored together with each pcap.
pcapFS

- pcapFS is a FUSE module mounting captured network data as a virtual file system
- Filesystem in Userspace is part of the Linux kernel and available for multiple operating systems including FreeBSD, OpenBSD and MacOS
- Another “pcapFS” was already released as part of the PyFlag framework by Michael Cohen
  - Unfortunately deprecated and not maintained 😞
- Index files can be stored in memory or on disk for future mounts
- Protocols are implemented by virtual file classes
Demo
pcapFS vs. Wireshark
Demo
pcapFS vs. Wireshark

- **Usability**
  - Data is presented using the virtual file system
  - Its hierarchy can be specified using multiple sorting options

- **Performance**
  - First mount of a pcap creates an index file
  - Browsing through the mounted data takes almost no time
  - Mounting with a used index is significantly faster than Wireshark

- **Resources**
  - Files in pcapFS point directly into the pcap or other virtual files
  - They are only extracted on demand
Demo
Beyond Wireshark

Demo
Demo
Beyond Wireshark

- pcapFS supports mounting of split pcap files
- File system level tools can be used on the mounted data without any extraction
- Metadata can be preprocessed and displayed as an own file as for example:
  - HTTP header
  - DNS requests and responses (e.g. as JSON)
- Missing data in streams can easily be padded for reconstruction
Demo

Working with pcapFS
Demo

Working with pcapFS

- **Decryption** of data by providing the corresponding key files
  - More cipher suites for SSL will be added in the future
  - Key files can be implemented for multiple protocols
- **Configuration files** force a protocol decoding on files with specified properties:
  - e.g. XOR dstPort 31489 protocol http
Summary

- pcapFS gives investigators the possibility to
  - quickly take a look at the relevant data of a network capture
  - order the data by different criteria
  - use file system level tools for their analysis

- Keeping an index file for each pcap significantly increases the performance of analyzing pcaps
- Using virtual files eliminates the overhead of extracting data out of pcaps
Future Plans

- Add support for more protocols (wishes are more than welcome!)
  - Particularly add support for other cipher suites in SSL
  - BitTorrent, HTTP2, SMB
- Add support for more metadata
  - e.g. SSL certificates
- Make use of Symbolic Links (e.g. reverse connections)
- Add support for pcapng
Thanks for your attention!

https://github.com/fkie-cad/pcapfs

jan-niclas.hilgert@fkie.fraunhofer.de