Defeating the PCAP Problem: Making a Mountain into a Molehill

Leigh Metcalf PhD, Daniel Ruef, Dillon Lareau, Angela Horneman

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

*[Distribution Statement A] Approved for public release and unlimited distribution

Document Markings

Copyright 2019 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

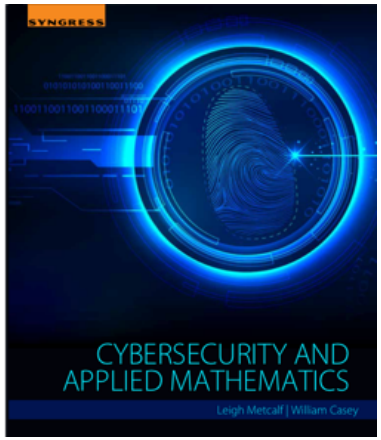
This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM19-1250

Who am I?

- PhD in Theoretical Mathematics, specializing in Algebraic Topology and Dynamical Systems
- Researcher at CERT for 9 years
- Primary Author of Applied Mathematics and Cybersecurity
- Co-EiC of the new journal ACM Digital Threats: Research and Practice
- My work is challenging assumptions made in Cybersecurity

and evidence based analysis



© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Analysis Pipeline…

…is not a spiffy User Interface.

It was developed to drop files for spiffy user interfaces to read.

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

What Can Analysis Pipeline Do?

- Narrow down where to look for bad guys
- Keep Inventory
- Network Profiling – Keep state
  - For example, suppose a system suddenly starts accepting web requests and it has never done that before
- It will only put actionable alerts into Security Information Management Systems
  - Can emit alerts in JSON for easy consumption
- Has thresholds for anomalies… custom thresholds even
- Can help you focus your analysis
- Analysis Pipeline uses SiLK concepts

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Analysis Pipeline

- On an order of magnitude better than an IPS in certain cases
  - For example, adding an IP to SNORT means writing a new rule
  - Adding an IP to the Analysis Pipeline monitoring is simple and efficient and doesn't change the inherent configuration
- For watchlists, it is incredibly efficient in processing speed and resource utilization. It costs almost nothing to add an indicator to a watchlist
- Works well in tandem with IPS

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

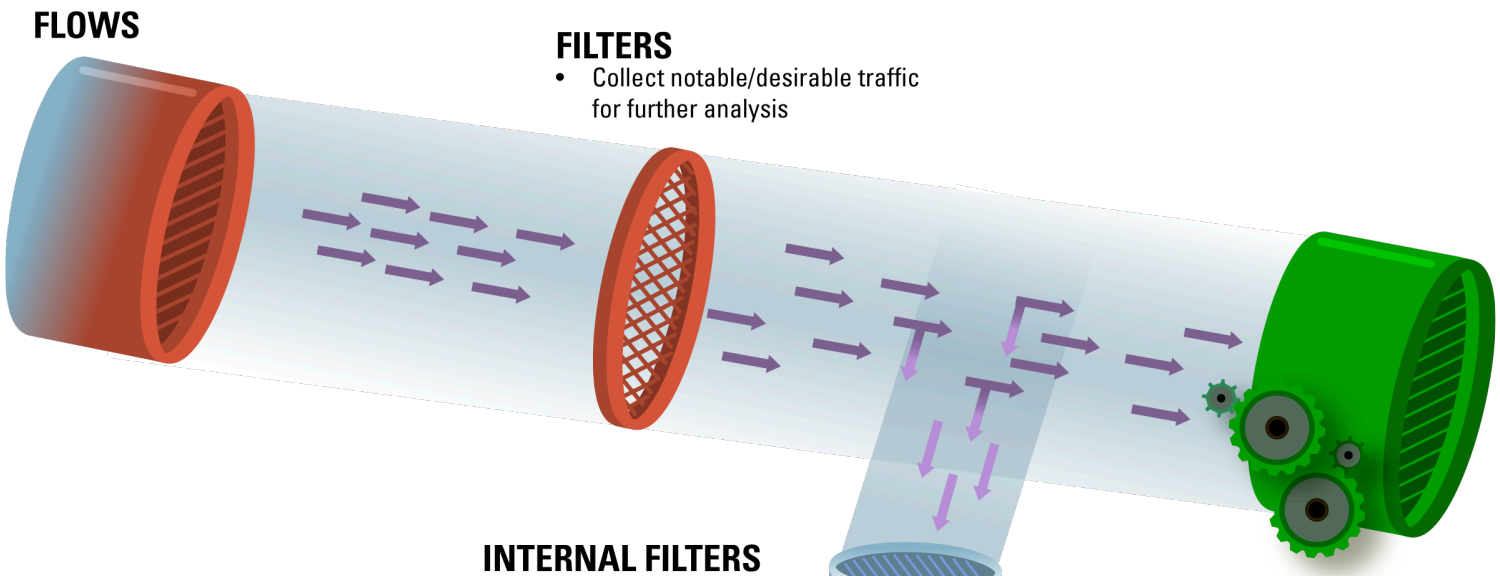
What Can't Analysis Pipeline Do

- Still isn't a cute User Interface
- And isn't a SIM
- It's only limitation is the amount of memory and disk on your system
- It also isn't an AI/ML tool. You can use it to label data for AI/ML but it isn't a flow passthrough tool to label all data.

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Analysis Pipeline



© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Analysis Pipeline Configs

The configurations consist of two parts:

Filter – The step that determines if the record should be passed to an evaluation

Examples: dIP, sIP, Ports, Protocol, Domain names, etc

Evaluation – The step that does the work

Examples: statistics, thresh holds, moving time windows

Evaluation also handles the alerts

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Case Study #1

1) Only worry about tcp traffic
2) Identify IP addresses that send more than 1MB in 5 minutes

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Case Study #1

```
FILTER tcpFilter
      PROTOCOL == 6
END FILTER
EVALUATION CaseStudy1
      ALERT EVERYTHING
      FILTER tcpFilter
      CHECK THRESHOLD
            SUM BYTES > 1000000
            TIME_WINDOW 5 MINUTES
      END CHECK
END EVALUATION
```


© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Complexifying…

Identify web servers

- Sends more than 20,000 bytes in 10 minutes
- Talks to more than 15 different DIPs in 15 minutes

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Configuration

```
FILTER ourNetwork
  SIP == 193.0.0.0/8 #arbitrarily chosen network
  SPORT IN LIST [80, 8080] #Yes, there are more ports, but 80 and 8080 keep it simple
END FILTER
```


© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Configuration Continued

```
EVALUATION findWebServers
    SEVERITY 2
    OUTPUT_TIMEOUT 1 DAY
    OUTPUT_LIST SIP
webServerList
    CLEAR ALWAYS
    ALERT EVERYTHING
    ALERT 1 TIMES 1 HOUR
    FILTER ourNetwork
```

```
FOREACH SIP
  CHECK THRESHOLD
    DISTINCT DIP > 15
    TIME_WINDOW 15 MINUTES
  END CHECK
  CHECK THRESHOLD
    SUM BYTES > 20000
    TIME_WINDOW 10 MINUTES
  END CHECK
END EVALUATION
```


© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Case Study #2 DDoS Detection

What is DDoS?

What are the characteristics of flows that participate in a DDoS attack?

What is the threshold event that causes us to say a DDoS attack is occurring?

Are there exceptions to the threshold?

Is this behavior ever expected and if so, what do we whitelist?

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

DDoS DNS Reflection

Rather than finding if we're the victim of a DDoS

…Let's look at if we're the inadvertent part of one.

DNS Reflection attack detection

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

DDoS DNS Reflection

```
FILTER dnsFlows
    PROTOCOL IN_LIST [6,17]
    ANY PORT == 53
    ANY IP IN_LIST "whiteListDns.set"
END FILTER
```


© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

DDoS DNS Reflection

```
EVALUATION dnsThreshold
 FILTER dnsFlows
 CHECK RATIO
   OUTGOING 27 TO 1
 END CHECK
 FOREACH SIP DIP
   CHECK THRESHOLD
    RECORD COUNT > 100
    TIME_WINDOW 5 MINUTES
   END CHECK
 OUTPUT_TIMEOUT 5 MINUTES
   ALERT ALWAYS
   CLEAR ALWAYS
END EVALUATION
```


© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Another Example

```
EVALUATION beacon
    CLEAR NEVER
    SEVERITY 3
    ALERT JUST_NEW_THIS_TIME
    ALERT 1 TIMES 60 SECONDS
    FILTER all
    CHECK BEACON
        COUNT 5 CHECK_TOLERANCE 5 PERCENT
        TIME_WINDOW 300 SECONDS
    END CHECK
END EVALUATION
```


© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Analysis Pipeline

These Examples look at Network Flow as the input.

Analysis Pipeline also takes input from YAF (IPFIX-based file format) and can operate on that as well. In fact, it works on all flat IPFIX Data

Analyzing:

- User agent strings
- Domain names
- URLs…
- ….anything you can put in IPFIX…

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Pipeline Can Also..

- Find system misconfigurations
- Find DNS Parking and Fast Flux
  - https://www.usenix.org/conference/laser2017/presentation/metcalf
- Watchlists
- Find Sensor Outages
- Passive FTP detection
- Network Profiling
- Baseball Statistics
- ….

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Analysis Pipeline Primitives

- Arithmetic Basics (Count / Sum / Average / # Distinct)
- Statistics Basics (Mean / Standard Deviation)
- Moving Average Windows
- Beacon Detection
- Persistence of Value Detection
- …And many more!

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Analysis Pipeline

Analysis Pipeline:

https://tools.netsa.cert.org/analysis-pipeline5/index.html

SiLK:

https://tools.netsa.cert.org/silk/index.html

YAF:

https://tools.netsa.cert.org/yaf/index.html

Email for help, requests, other: netsa-help@cert.org

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution

Thank you for your time!

Questions/Comments?

© 2018 Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution