Carnegie Mellon University

Distribution Statements

This material is based upon work funded and supported by FloCon reimbursable funding under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

DM-0004070


SEI WEBINAR SERIES | Keeping you informed of the latest solutions
Software Engineering Institute | Carnegie Mellon University

Building Analytics for Network Flow Records

Timothy Shimeall, Ph.D.

Matthew Heckathorn

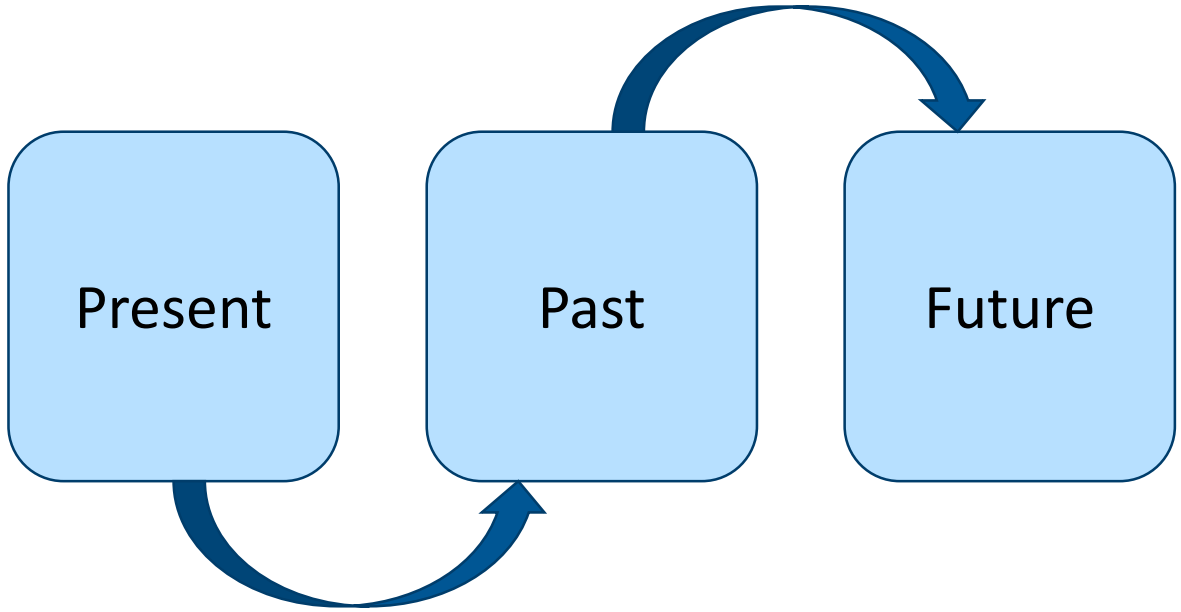
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Poll on Automated Analysis

How is your organization automating analysis for network situational awareness or network security?

- Uncertain: We don’t do network situational awareness
- Outsourced: We have a good managed security solution
- Feeds: We merge several external threat feeds to develop analytics (data-centric, not directed)
- Heroes: We hire good people and leave it in their hands
- Repeatable: We have a process, but have little management support
- Optimized: We have a process and a focus on continuing to improve it (mission-focused)
- Other (please explain)

Business Value Perspective



Present – current operational system

- Created
- Referenced / Modified
- Deleted

Past – data from repository

Future – proactive security

N. Sheikh, Implementing Analytics, Morgan Kaufmann, Boston, MA, 2013

What we won’t cover

Information security basics

- C.I.A. or Kill Chain or CAPEC
- Indicator analysis

Implementation details

- The basics of network flow analysis
- SiLK tool suite command syntax
- Scripting languages

Capability Maturity Models

Building Analytics for Network Flow Records

Analytic Development Process

Process

Explore

Model

Test

Analyze

Refine

Explore

Needs analysis – is there a prior analytic that addresses this?

Research analytic

- vendor documentation
- published papers
- data feeds

Identify unique attributes

- ports
- protocols
- associations
- behaviors

Model

Lessons learned from prior analytics

Build model

- identified behavior
- similar behavior

Program model

- Shell
- Python
- other

Test

Execute programmed model

- monitor progress
- debug

Save test results

- ‘raw’ files
- ‘set’ files
- ‘bag’ files
- other formats

Analyze

Review test results.

Reduce false positives.

Reduce false negatives.

Identify improvements.

Refine

Apply improvements

Update programs

Repeat

Mature the process

Maturing the process

Templates

- Common input / output options
- Documentation content
- Common style (and in-script documentation)
  - Invoking commonly-used tools in conventional way
  - Describing common aspects in conventional way

Test suite

- Data with known content
- Regression tests
- Example output for documentation

Source control with versioning (code and documentation)

Poll on Examples

Which analytic are you most interested in?

- Host characterization: what mix of services per address?
- Backwards: what hosts either send or receive traffic that appears reversed?
- Scanners: what external addresses are mapping our network?
- Profiling Popular Usage: what are our popular services and protocols?
- Profiling Active Talkers: what are the active addresses on our network?
- Profiling Inventory Assets: what assets are using/serving what services?

Building Analytics for Network Flow Records

Example Analytics

Understanding Host Roles

Explore: Characterize hosts that seem to act as email servers

- Of the hosts communicating on TCP port 25 (SMTP), how much non-SMTP traffic does each generate?
- Changes over time? After event?

Model:

- Input:
  - Assume small population of interesting hosts (specified as IP set)
  - Pre-retrieve traffic of interest (as rw file)
- Use rwfilter with rwuniq to pull out SMTP vs non-SMTP flow counts
- Output: Table of behavior per IP address

Test, Analyze, and Refine:

- Include test cases for addresses with known and unknown roles
- Reliability / performance / interpretable results
- How about non-SMTP email activity? (e.g., POP, IMAP)
- How about non-SMTP related to email? (e.g. DNS)

Initial Host Characterization Script

```
#!/bin/bash
rm -f  more-mail-saddr.txt more-nomail-saddr.txt more-nomail.rw
rwfilter in_month.rw --sipset=interest.set --pass=stdout \
      | rwfilter stdin --protocol=6 --aport=25 \
         --fail=more-nomail.rw --pass=stdout \
      | rwuniq --field=sIP --no-titles --ip-format=zero-padded \
         --sort-output --output-path=more-mail-saddr.txt
rwuniq more-nomail.rw --field=sIP --ip-format=zero-padded \
     --no-titles --sort-output --output-path=more-nomail-saddr.txt
echo '              sIP|      mail||  not mail|' \
      ; join more-mail-saddr.txt more-nomail-saddr.txt \
      | sort -t'|' -nrk2,2 \
      | head -n 5
```

Using SiLK for Network Traffic Analysis (Example 3.37)

Identifying Backwards Traffic

Explore: Identify hosts for which sensors record traffic that appears reversed (possible forged addresses).

- Of the hosts sending or receiving TCP traffic, for which do sensors record traffic that is “inbound” but from local hosts, or “outbound” and going to local hosts?
- Changes over time? Related to event?

Model:

- Assume we have hosts of interest, expressed as IP set
- Assume we have date/time range of interest, expressed as parameters
- Use rwfilter to pull traffic into files for later analysis

Test, Analyze, and Refine:

- Test against normal traffic and traffic engineered to be reverse
- Reliability / Interpretable results
- Traffic not completely reversed? Traffic observed in both groups?

Initial Backwards Traffic Script

```
#!/bin/bash
START=2009/4/20T12
END=2009/4/20T13
SENNAME=SEN1
rm -f strange_in.rw strange_out.rw
rwfilter --sensor=$SENNAME --type=in,inweb --start-date=$START \
             --end-date=$END --protocol=6 --bytes-per-packet=65- \
             --sipset=mynetwork.set --flags-all=SAF/SAFR,SAR/SAFR,SAFR/SAFR \
             --packets=4- --pass=strange_in.rw
rwfilter --sensor=$SENNAME --type=out,outweb --start-date=$START \
             --end-date=$END --dipset=mynetwork.set --not-sipset=mynetwork.set\
             --pass=strange_out.rw
exit 0
```

Using SiLK for Network Traffic Analysis (Example 4.22)

Finding Scanners

Explore: Identify external addresses that are mapping our network

- Much existing work in literature (Gates, Jung, etc.)
- Simple approach exploits attacker workflow (minimal advance knowledge)
  - Of quick TCP traffic with relatively few bytes per packet, which have flags that could reflect scanning? Are the sources frequent enough?

Model:

- Assume a date of interest, expressed as parameters
- Use rwfilter to isolate TCP traffic with few bytes per packet, then to split out flag combinations.
- Use rwbag to count sources per bag combinations
- Use rwbagtool to manipulate counts and isolate frequent sources
- Use rwbagtool to produce set of IP addresses for sources

Test, Analyze, and Refine:

- Use test cases with known scans and without known scans
- False positives and false negatives
- Trends over time?

Gates, C. et. al., “Detecting Scans at the ISP level”, http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=8073

Jung, J. et. al., “Fast Portscan Detection Using Sequential Hypothesis Testing”, http://nms.lcs.mit.edu/papers/portscan-oakland04.pdf

Initial Scanner Detection Script

```
#!/bin/bash
rm -f fastfile.rw fast-{low,high}.{set,bag} scan.set
rwfilter --start=2009/04/20 --sensor=S0 --type=in,inweb --bytes-per=40-65 \
   --protocol=6 --duration=0-1199 --pass=fastfile.rw
rwfilter fastfile.rw --flags-all=S/SRF --packets=1-3 --pass=stdout \
      | rwbag --sip-flows=fast-low.bag
rwfilter fastfile.rw --flags-all=SAF/SARF,SR/SRF --pass=stdout \
      | rwbag --sip-flows=fast-high.bag
rwbagtool fast-high.bag --maxcounter=10 --coverset --output-path=fast-high.set
rwbagtool fast-low.bag --mincounter=10 --coverset --output-path=fast-low.set
rwsettool --difference fast-low.set fast-high.set --output-path=scan.set
exit 0
```

Using SiLK for Network Traffic Analysis (Example 4.33)

Profiling Top Five Services and Protocols

Explore: Characterize traffic on our network by protocols and services

- What protocols are being used? What services are we requesting? What services are we providing?
- Changes over time? Anomalies?

Model:

- Input:
  - Pre-retrieve traffic for some time period, typically a full days worth.
- Use rwfilter with rwstats to generate top 5 statistics
- Output: Multiple tables containing summarizations of traffic

Test, Analyze, and Refine:

- Include test cases for how to handle large sets of data
- Reliability / **performance** / interpretable results
- Would a list of top 10's provide more value for the performance hit?
- Is generating a table of all protocols and services then filtering text, rather than SiLK binary, feasible?
- Inbound vs. Outbound traffic?

Profiling Top Five Services and Protocols Script

```
#!/bin/bash
echo Initial dataset:
rwfilter --type=out,outweb --start-date=2011/09/28:00 \
   --end-date=2011/09/28:23 --protocol=0- --pass=sample.rw
echo -e “\nTop protocols:”
rwstats sample.rw --fields=protocol --count=5
echo -e “\nTop services being requested:”
rwstats sample.rw --count=5 --fields=dport
echo -e “\nTop services being provided:”
rwstats sample.rw --count=5 --fields=sport
```

Network Profiling Using Flow(Section 3 Script)

Profiling Active Addresses

Explore: Identify addresses that have actively talked during a given time period

- How many active addresses are using TCP? What about other protocols? Are there addresses that are merely passing through our network (transiting)
- Awareness of network address topology. Changes over time? Anomalies?

Model:

- Input:
  - Pre-retrieve traffic for some time period, typically a full days worth.
- Use rwfilter with rwset tools to generate and understand address lists
- Output: Multiple tables containing summarizations of active addresses, Multiple set files

Test, Analyze, and Refine:

- Include test cases for how to handle large sets of data
- Reliability / **performance** / interpretable results
- What about separating out UDP and ICMP from other IP protocols?
- What about traffic heading into the network from an internal host?

Profiling Active Addresses Script

```
#!/bin/bash
echo Number of TCP talkers:
rwfilter sample.rw --type=out,outweb  --protocol=6 --packets=4- --ack-flag=1 \
    --pass=stdout | rwset --sip-file=tcp_talkers.set
rwsetcat tcp_talkers.set --count
echo -e “\nNumber of talkers on other protocols:"
rwfilter sample.rw --type=out --protocol=0-5,7- --pass=stdout \
| rwset --sip-file=other_talkers.set
rwsetcat other_talkers.set --count
rwsettool --union tcp_talkers.set other_talkers.set --output-path=talkers.set
rwsetcat talkers.set --network-structure
echo -e “\n\nClass C network blocks:"
rwsetcat talkers.set --network-structure=C
echo -e "Transit traffic:"
rwfilter sample.rw  --type=out,outweb  --not-sipset=talkers.set --pass=stdout \
| rwtotal --sip-first-8 --summation --skip-zeroes  --no-titles | cut -f 2 -d “|”
rwfilter sample.rw  --type=out,outweb  --dipset=talkers.set --pass=stdout \
| rwtotal --sip-first-8 --summation --skip-zeroes --no-titles | cut -f 2 -d “|”
```

Network Profiling Using Flow(Section 4 Script)

Profiling Assets by Service

Explore: Identify assets in our network by popular services

- What assets are running web servers? DNS Servers? Telnet servers? What assets are telnet clients?
- Changes over time? Policy Violations?

Model:

- Input:
  - Pre-retrieve traffic for some time period, typically a full days worth.
- Use rwfilter, rwstats, rwuniq, and rwset tools to generate asset lists and statistics
- Output: Multiple tables containing summarizations of traffic, Multiple set files

Test, Analyze, and Refine:

- Include test cases for how to handle large sets of data
- Reliability / **performance** / interpretable results
- What additional services are of interest?
- Do we have to limit our output in some manner? (e.g.: > 1% of packets)

Profiling Assets by Service Script

Network Profiling Using Flow(Section 5 Script)

Furthering Your SiLK Analysis Skills

SiLK tools site

- http://tools.netsa.cert.org

Using SiLK for Network Traffic Analysis

- http://tools.netsa.cert.org/silk/analysis-handbook.pdf

Tool Tips

- https://tools.netsa.cert.org/confluence/display/tt/Tooltips

Flow analysis research and advanced techniques

- http://www.cert.org/netsa

FloCon (January 2017, San Diego CA)

- http://www.cert.org/flocon