Correlated Security Enforcement
Enriching security events using network traffic and event monitoring data

Nick Buraglio
Network Engineer, Network Planning Team
Energy Sciences Network
buraglio@es.net

TNC 17
May 31, 2017
Correlated Security Enforcement

- What does that *actually* mean?
  - Utilize existing network data not typically used for security purposes
  - Create a functional data repository able to store, and reference
  - Execute actions based on variable data sources from across a given set of systems (e.g. a transit ASN)
  - Analogous to SARNET “analyze”
Motivations

• Simplify or remove need for on-demand forensics during a given event
• Increase the detail and sensitivity of events being cross referenced
• Add diversity to the data sources used to take action
• Reduce false positives and negatives by corroborating events with data
• Understand where best to take action given the network topology
Motivations

• Significantly narrow margin for human error
• Allows for extensive programmatic changes to complex elements
• Facilitate more seamless “undoing” of both manual and automated actions (e.g. Black Hole block scaling based on abuse level)*
• Make sure we’re not building wheels

*Some networks do this now with black hole routing
Goals

- Leverage existing network data traditionally used for network triage and root cause analysis for network events
- Like a SIEM for the WAN
- Give it “all of the data”
- Index “all of the data”
- Allow for broad and flexible data inputs
- Provide mechanisms for extensive output actions
- Don’t build wheels
Goals

• Extend current offerings
  – More granular filtering
  – Faster responses to events
  – Software based checks and balances (whitelist, blacklist, intel feeds)
  – Extend what we’re already doing to a wide area context
  – Feed back into the system to create more specific and accurate triggers
Simple design

Input

Index / Correlation / Enhancement

Actions
Input

● The usual [network] suspects
  ○ Syslog data
  ○ Flow data (Sampled or not - at least 1:2000)
  ○ DNS Query Logs
  ○ Community Intel feeds
  ○ Existing IDS Alerts

● The not-so-usual suspects
  ○ SDN Controller data
  ○ Routing topology
Input / Corroboration

- Initial searches were very slow
  - Time to completion was hours and not seconds
- Initial builds worked well (due to simplicity)
- NetFlow files take the longest to manually crunch
- In order to speed up processing added stretch goal of “Index all of the data” (that it is possible to index)
Correlated Security Enforcement - Correlation

- Very simple
- Very modular
- Very lightweight

Input

Index / Correlation

Actions
Correlation

- Concept code written in python
  - Built to handle everything from flat files (flow data, syslog) to APIs (packet design and Arbor)
  - Ran in a mid-tier Ubuntu VM
  - Inputs were diverse
  - Output action was a stretch goal

- Bad actors are sourced from bro logs
  - 3 bro systems in diverse geographical and topological locations
  - Search all provided inputs for relevant data in bro alert
  - Match relevant data (src/dst ip, ports)
  - Build topological paths based on route table
Enter: Indexing

- “Traditional” Indexing tools
  - ELK Stack
  - Splunk
- Cloud Tools
  - Google BigQuery / Data Flow*

*Under investigation
Actions

- Actions are underway with future support for
  - OpenFlow Flowmod
  - BGP Null Route
  - BGP FlowSpec
  - API
    - Alarm via slack
    - API calls to NCSA BHR instance
    - Build intel base
  - Eventually no alerts (only a report) - actions should be automatic*

*
Workflow

1. Alerts
   - Process Alerts
      - Queue (id, alert)

2. Extract Flow data
   - Queue (id, flow_pair)

3. Search queued flows
4. Map flows to netflow
5. Add netflow results

6. Guess possible routes
   - Queue (id, routes)

7. Add potential routes

8. Export Results

Note: Pool = 20

Symbols:
- Nfdump
- Queue (id, flow_pair)
- Queue (id, netflow_data)
- Queue (id, routes)
Correlated Security Enforcement
Correlated Security Enforcement

ESnet

AS Border Router

Policy Enforcement Point

High Value Peerings

BGP Flowspec

Aggregation and Correlation (CoreFlow)

Intel Feeds

API Request for query

Corroborated events

Routing Data via pmacct

Syslog DNS query logs Black Hole Router Logs

NetFlow data from backbone routers

Indexed Data Repository

ELK Stack

Intel Feeds

Bro Alarms

16
Correlated Security Enforcement - Workflow
Correlated Security Enforcement - Workflow
Correlated Security Enforcement - Workflow

- Bro IDS
- Sensitive infrastructure
- AS65293
- AS65099
- Policy Enforcement Point
- BGP Flowspec
- Aggregation and Correlation (CoreFlow)
- Syslog message: Unauthorized ssh login attempt from X.Y.Z.17
- NetFlow Record: scrip X.Y.Z.17 to AS65099 A.B.C.146 dst port 22
- Bro conn log from scrip X.Y.Z.17 dst port 22
- DNS query info for eset-hax0r.io
- ELK Stack
- Indexed Data Repository
- Routing Data via pmacct
- Bro Alarms
- NetFlow data from backbone routers
- ELK Stack
- Intel Feeds
Correlated Security Enforcement - Actions
Stretch Goals

- Leverage machine learning
- Automate and learn from events
- Leverage high power, out of band resources to discover patterns and similarities not easily seen otherwise
- Add additional bro sensors across the WAN
  - Monitor low speed infrastructure “in the wild”
  - Integrate Layer 7 patterns
- Keep adding data sources
- Move processing to the cloud
- Integrate into NSF CICI project #1642142 (Secure SDX)
Correlated Security Enforcement - Analogs

- Commercial options are few and far between
  - Mostly enterprise focused
  - Some WAN options - but mostly different or incomplete
- Components are plentiful
  - Build it like Lego
- **Apache Metron**
- **Kentik**
- **Arbor**
- ?
“Code or it didn’t happen”

- Code available at (private repository):
  
  https://github.com/esnet/CoreFlow
Contact

- Nick Buraglio (ESnet)
  buraglio@es.net
  https://www.es.net/about/esnet-staff/network-planning/nick-buraglio/
- Ralph Koning (UvA)
  r.koning@uva.nl
  https://staff.fnwi.uva.nl/r.koning/