Finding Little Things in Big Data

BU Security Camp 2016

Patrick Cain

Patrick.Cain@bc.edu, pcain@coopercain.com, Patrick.Cain@tufts.edu

Suggested Alternate Titles for this Talk

- Free as In * (puppy, cat, beer, wife, ….)
- Losing Important Things in Big Data
- Finding a Haystack with a Needle

What I Did on My Summer Vacation…a story

(i.e., the project that's gonna take four years)

So, What’s the Challenge?

- We all “want” logs
  - They help us find “things”: some bad, some good, some terrifying
  - Every log is in a different format
- We all “store” logs
  - Keep them around for analysis
  - Mostly to fill up storage volumes
  - The storage formats are all different
- We want to “use” the log contents
  - Maybe use them for forensics
- Maybe, we even want to find stuff in them
  - In Real-time or Historical

My Stored System Logs are Bigger than yours

- Some example sizes of log storage

Syslog+snort+IDS	850/s - 3000/s
Netflow	20000/s – 50000/s
Domain Logins	150/s – 350/s
Firewall Deny-s	400/s – 3500/s
DNS	3500/s -7000/s
DHCP	350/s - 2200/s
802.1x	600/s - 1800/s

- The lawyers want logs for a YEAR, so your storage could be:

Customer #1:	currently 28TB (90d), a year is ~70TB
Customer #2:	currently 6TB (265d), a year is ~7TB
Customer #3:	currently 17TB (30d), a year is ~194TB

UNIVERSITY NAME

Here Comes Bro…

- ‘bro’ is a network security monitor
  - Generates a detailed log of network activity
  - Boy, does it generate logs…..
- Bro storage
  - #a: 87GB/day
  - #b: 130GB/day
  - #c: 3GB/day (pbbthppt!)

  - So a year is… 31,755 GB or 47,450 GB or 900GB

This talk will be boring ☺

- I’m not going to do any catch-the-hacker things
- Many of us have 10GB internet links
  - Some of us may have 40GB intra data center links
  - Most of us are challenged to keep up with the data flowing by
- Soon, we’ll have 40GB Internet links (Or 100GB)
  - Even if we shunt the “good” traffic
    - How do we store the bad packets?
    - How do we search back 90 days?
    - (Hmmm, how much storage will I need?)
- So this is a “how do we deal with the flood of data” talk.

- You may have the mother-of-all splunks
- I may have the father-of-all SIEMs
- But how do I sift through 500 Tb of events?
  - Better, how do I get the student workers to do it?

- Do I really want to store this data in expensive flash?
- Can my SIEM/syslog/splunk hold this much data?

The Nagging Issue

So How Do we Use the TBs or PBs of data?

- “search splunk”
- “search arcsight logger”
- “grep the disk drive”

- But every “user” needs an account on all the search systems
- How do we not kill the budget on events/sec licensing?

- But these are raw searches. ☹
- How fast could we make the searches?

What to do?

- “Put it all in splunk”
  - A 90-day search still takes a while
- What if the searchee went through a proxy
  - There’s some correlation necessary
  - Or we want a user and we use DHCP

- The normal process:
  - Search flows -> get some data
  - Search dhcp/user data -> get more data
  - Try to find something in the SIEM
  - For data beaches first get IPs then search flows

- Zcat or zgrep on logs took forever
  - 3 weeks for 90 days of logs
- If you wanted 10.0.0.10 and typed 10.0.0.1 you had to wait a second time
- At times, search speed was important

Functions for normal searches

- The Firewall did it?
  - Did the FW block something it shouldn’t have?
  - NOT!
- What did the (bad) User do?
  - IDS, IPS, AV, etc., events
- Who clicked on the phish?
- What happened to a device 3 months ago?
  - *****

Correlation is Useful, but…

- If you have a SEIM-ish system.
  - Correlate IP activity with a user
  - Did the user at that time have other IP addresses?
  - Respond to multiple firewall deny events
  - Look for repeatedly poor activity in WordPress
- The real goal was to do historical searches
  - Did an IP talk to another in the last 6 months?
  - Did anyone else click on this link in the last month?
  - Due to volume, keeping all this data, on-line, in the SIEM, is not practical.
    - Why make the SIEM index all this data we may never look at?

The adventure begins…

We called it “research”

Building your own searcher is a lot of work

1. Get a server or VM
2. Design the web page
   - Code it up
3. Craft a database
   - Write lots of (no)SQL
4. Get data into the database
5. Keep the sucker running

- Could we find something that does 2, 3, & 4? On the cheap.

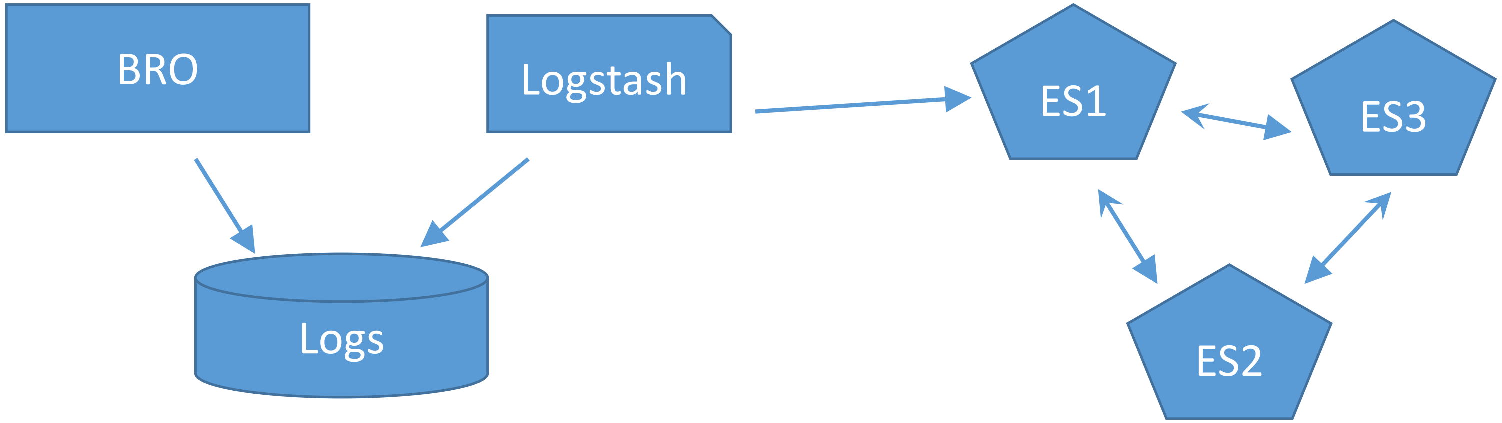
Database Choices

- Event searching needed to be fast (and easy) (and cheap)
- Major Database Types:

1. Relational (Oracle, mysql)
    - Doesn’t handle large data sets easily
2. Key-value (Berkeley DB, REDIS)
    - Easy insert; searching can be slow; light fault-tolerance
3. Document (Mongo DB, ElasticSearch)
    - Data goes in and comes back in blobs; good fault tolerance
4. Graph (Neo4j, InfoGrid)
    - More data relationships; fault tolerance provided by file system
5. Distributed File (HADOOP)
    - Multiple tasks on same data set at once; “more than a DB”

We start…

- ELK platform: ElasticSearch as database
- 3 VMs (2 proc; 4G mem, 500GB disk)
- Each VM got a data node; anybody can be master
  - Data was replicated on two data nodes
- Send bro logs through logstash to ES node



Lessons Learned (1 of many)

- 500GB * 3 of disk is not that big
- ES nodes talk to each other to stay in sync
    - Session timeouts cause them to panic
- “Tuning” logstash took a while. And continues
- Keeping a multi-system database running takes more work than you think.

Second Try…

- We’re tired of proving the firewall wasn’t blocking your traffic.
- So … Stop sending bro logs to ES; Send Firewall Logs into the ElasticSearch cluster instead

- The setup
  - Re-use some older hardware
  - one VM with a customized Kibana instance and a no-data ES node
  - The 3 VMs from before

Screenshots

Back to the original goal…

- Joke: Is it “Big data” if it fits in one box?
- Since the FW log viewer was now “in production” we took a different system for the bro logs:
  - 8 cores, 24 GB mem, 5TB disk
  - Installed ELK
  - Blast logs at it.

Lessons still learning (#2 of many)

- ES can’t handle large volumes of events quickly
  - We need to buffer the data going into the ES cluster
    - Using Kaftka, redis, mq, etc.
- Logstash (or equivalent) is a pain
  - New fields requires new “tuning”
  - Can we export bro logs in json and not have to run the through logstash?
- Disk drives fill up real quick
  - Particularly if you’re keeping two copies of the data

The next adventure…

Searching is fun but correlation causes fun…depression…more work

How to Correlate? Searches Don’t do that!

- But … We saw the flows, but who were those people?
  - Can we connect the user db with the logs?
- Bro logs are linked:
  - Conn.log -> protocol.log -> files.log

  - 11G http
- One could run blacklists and only forward hits to the SIEM.

Trying to find the little things

1. We have an IP Address of interest (IP1)
2. Flow data said we have traffic to/fro it from IP2
3. There are no security sensor hits for IP1
4. Re-search flows for other activity from IP2 on campus
5. Correlate from bottom up to find out what happened

Or

- Snort is good at generating alerts
  - But it’s uni-directional
- Can we link snort alerts with the response
  - i.e., snort alerts on a packet blocked downstream
  - i.e., did the host AV delete the virus?

Doing it the hard/easy way

- Get the hash of the file that was downloaded
- See who else got that hash value
- Correlate, correlate, correlate
- Block whatever caused the issue

- Why?
  - Did the AV catch all of this virus?
  - Are the recipients actually running AV?

The Future

- Link multiple correlators into one event
  - Backtrace the AV hit to what caused it (automatically)
- Could Maltego or neo4j or some graphing system make the obvious more obvious?
  - Think “make the invisible apparent”
- Can we define traffic filters to shunt good traffic?

Fini

- If you have solved this problem, please tell me.
- If you are an encase expert -> we need panelists for the spring. ☺

Thank you

UNIVERSITY NAME