Network Threat Hunting with ELK

Fast, repeatable threat hunting

https://github.com/DecayingSec/ConferenceTalks/tree/main/BloomCon/2021

Elevator Pitch

- The goal of this talk is to enable you to start threat hunting the network or improve your existing hunting
- This approach can speed up hunting dramatically
    - Spend more time on what matters not on wrangling data

GET /_security/user/btice

- Professionally
  - Proud BU Alumnus & returning BloomCon speaker
  - I try not to re-use “whoami” jokes
- At home
  - Recovering distro hopper
  - My other gaming rig is an overclocked Raspberry Pi
- At work
  - Full scope, full stack defense (for one office)
    - I wear *all* of the security hats for a full security team
    - I do daily/weekly threat hunting (network + endpoint)
  - Also our internal red team. One human purple team?
  - My approach is a little… *home alone*. Blue has the home-field advantage let's use it

Tell Them What You Are Going To Tell Them

- Formalities ← You are here
- Intro to Threat Hunting with ELK
  - What problem are we trying to solve?
  - What is Threat Hunting?
  - Before you begin hunting...
  - Why Hunt the Network?
  - Why leverage the Elk stack?
- Core Hunts
  - Top X
  - C2 Hunting
- Demos!
- Questions

What problem are we trying to solve?

- Adversary dwell time is too long
- Most breaches are discovered externally
- We need a way to detect compromises in-house
- There is no silver-bullet or easy-button to buy
    - Always People + Process + Technology

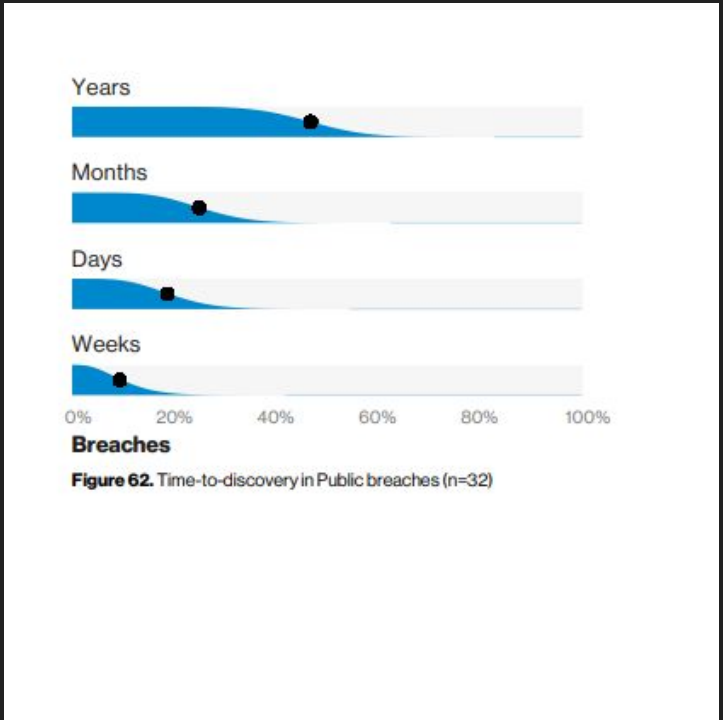

Figure 62. Time-to-discovery in Public breaches (n=32)

Source: Verizon Data Breach Investigations Report 2019

What is Threat Hunting?

- Threat hunting is the practice of proactively searching for compromise within your network
- Threat Hunting…
  - Increases the chance you detect the compromise internally
  - Puts a soft cap on how long a compromise can go on before you detect it
- Threat hunting is not…
  - Gathering OSINT on threat actors
  - “Threat intelligence”
    - You can use threat intelligence for hunting, but they are distinct
  - Reviewing alerts from IDS/EDR/SEIM
  - Creating new signature based detections

How does threat hunting work?

- Threat hunting works by putting the right data in front of a human analyst
  - Behavior based hunting looks for data that could be from from an attacker
  - Anomaly based hunting looks for unusual events
- We will be focusing on narrowing down the data that is worth an analysts time to investigate
  - If you want the whole package I recommend the (so far FREE) trainings from Active Countermeasures
    - https://www.activecountermeasures.com/original-threat-hunt-training-course/
  - They also have labs available
    - https://activecm.github.io/threat-hunting-labs/

Before you begin hunting...

- Get your house in order
- Hunting is how you enhance your detections beyond signatures which means you should already have the fundamental Protect and Detect things done
  - Block/Control Macros
  - Application Allowlisting/Whitelisting
  - Attack Surface Reduction Rules (Or equivalent)
  - Egress filtering
    - GeoIP filtering
    - Outbound port whitelisting
    - Threat intelligence (IOC) feeds
  - IPS + IDS (Tuned!)
  - Etc.

How often should you hunt? What time blocks?

- Two “knobs”
  - How often you hunt
  - How much time you cover with each hunt
- I like hunting…
  - At least once a week
  - Back to the last time I hunted
    - 1-day overlap to ensure nothing is missed
- Find what is right for your organization
  - Weekly? Monthly?
  - 3-day blocks? Longer?
  - These process decisions drive data retention needs

Why Hunt the Network?

- The network is the great equalizer
  - Everyone has to talk across it!
    - Including attackers
  - Excellent coverage, few blind spots
- Host level logging has limitations
  - Diversity of devices leading to lack of coverage
    - Ever try getting EDR software installed on a printer?
  - Diversity of log formats
    - Network is realistically 2 formats and maybe 3 sets of field names tops
      - JSON + CSV
- There is nothing wrong with host level hunting, but it isn't perfect
  - Both is good, but I recommend starting with network level hunting

Why leverage the Elk stack?

- Problems With Manual Hunting
  - Many manual steps
    - Use Zeek to generate logs
    - Analyze logs using CLI tools
    - Review reports
  - Limited time range
    - Often 1-day at at a time
  - Inconsistent hunting
    - Many tools, many commands
- Let’s Automate!
  - TAP -> NIC -> Zeek -> Logstash -> Elasticsearch -> Kibana -> Analyst
  - Run tools continuously and ingest results
  - Need newer data? Hit refresh in your browser
  - Need older data? Change time-range and click apply

What capabilities does ELK give us?

- Logstash can manipulate and enrich data
  - Subdomain splitting, GeoIP tagging, PCR calculations, entropy calculations
- Elasticsearch lets you… search
  - Need to review 15 days of logs? No problem
- Kibana saves your hunting calculations as visualizations
  - Command line is *fine*, but this is faster no matter how much of a guru you are
- Kibana (can) filter out known-good (In filters)
  - Never hunt the same thing twice

Technology Roundup

- Bro/Zeek for netflow and “deep packet analysis”
  - I would avoid any netflow/IDS that doesn’t track directions well
  - I would avoid any netflow/IDS that doesn’t parse layer 7 protocols
    - E.g. HTTP, TLS/SSL, SMB etc.
  - I would avoid any netflow/IDS that doesn’t provide passive DNS lookup
    - DNS query and response need to be in the same document…
  - Suricata can be used in a pinch, Zeek is your best option
- ELK stack for automation
- {ee-outliers,RITA} for beaconing detection
  - More on these later
- Feeling lazy? Start with a network monitoring platform
  - RockNSM
  - Security Onion

Behavior vs Anomaly Based Hunting

- Two approaches to putting the right data in front of an analyst
  - Look for traffic that could be generated by an attack
    - Based on high level (High up the pyramid of pain) patterns of behavior
    - In network threat hunting we look for patterns related to C2, staging, and exfil
      - Command & Control (C2) traffic is the most important for network threat hunting
  - Look for traffic that is unusual or anomalous
    - Metrics outside their normal range
    - Rare values in specific fields
      - Data stacking (“Least frequency analysis”)

Core Hunts (1/2)

- Top X
  - Connections by count
    - Top {client,server}
    - Top {port,protocol,service}
    - Top IP pair
  - Bandwidth
    - Top server {up,down}
    - Top client {up,down}
    - Top IP pair up/down
  - Locations
    - Country, City, Organization by connections, bandwidth up/down

Core Hunts (2/2)

- Persistent Connections
  - Longest connections (Max duration)
  - Persistent connections (Sum duration)
- DNS tunneling
  - Pay-level-domains by unique count of subdomains
  - (Optional) pay-level-domains by max/avg entropy
    - Can catch DGA
- Beaconing analysis
  - Pick one
    - RITA
    - ee-outliers

RITA vs ee-outliers (1/2)

- RITA
  - Scores IP pairs based on consistent timing, duration, and size
  - Creates a soft score instead of a Yes/No answer
  - Separate log generated
    - Cron job to ingest and generate reports
    - Ingest CSV with Logstash
  - High RAM requirements
  - Seperate DB (MongoDB) required
  - Port and protocol blind
  - Only IP pre-filtering
    - No way to filter out rejected traffic
  - Not real time (Run hourly, nightly)
  - Problems with non-default set separators (Bug report is in)

RITA vs ee-outliers (2/2)

- ee-outliers
  - Alerts on connections based on consistency of hourly bucket height
    - The same hunt as when you “look for a flat histogram”
  - Gives a Yes/No answer
  - Elastic-native
    - Tags existing data in Elasticsearch
  - Useful for more than beaconing
    - Can do anomaly detection as well
  - Can do SSL/HTTP beaconing by analyzing the hostname or SNI instead of server IP
  - Pre-filtering (query) and post-filtering (whitelisting)
  - Log format and field name agnostic
  - Runs in near-real-time (5ish minutes/Delay depends on exact analysis)
  - Can send you alerts (If you want that)
  - Problems with derived fields (Possibly an ES 7 incompatibility?)

Demos!

- Top X
- Persistent Connections
- DNS Tunneling
- Beaconing
    - Ee-outliers
    - RITA
- “Malware”
    - DNS Tunnel
    - Fast HTTP beacon with jitter
    - Slow HTTPS/TLS beacon
    - Persistent telnet connection
    - Disclaimer: No malware was harmed in the making of these demos

Recap

- Hunt the Network
- Use ELK to automate!
- Look for:
  - Beacons
  - DNS tunnels
  - Persistent connections
- Profit

That’s all folks!

- Questions?
  - Concerns?
    - Dreams?
      - Fears?
        - Aspirations?
          - Snide remarks?

Bonus: Anomaly Based Hunting!

- Metrics out of regular range
  - Bandwidth
  - Producer-Consumer-Ratio (PCR)
  - DNS TTL
  - DNS Query Entropy
- Rare values
  - Hostname/SNI
  - GeoIP City,Country
  - JA3 Hash
- Tools
  - Data table (Kibana)
  - TimeLion
  - ee-outliers