A Federated Approach to Identifying Advanced Persistent Security Threats on Enterprise Computer Networks

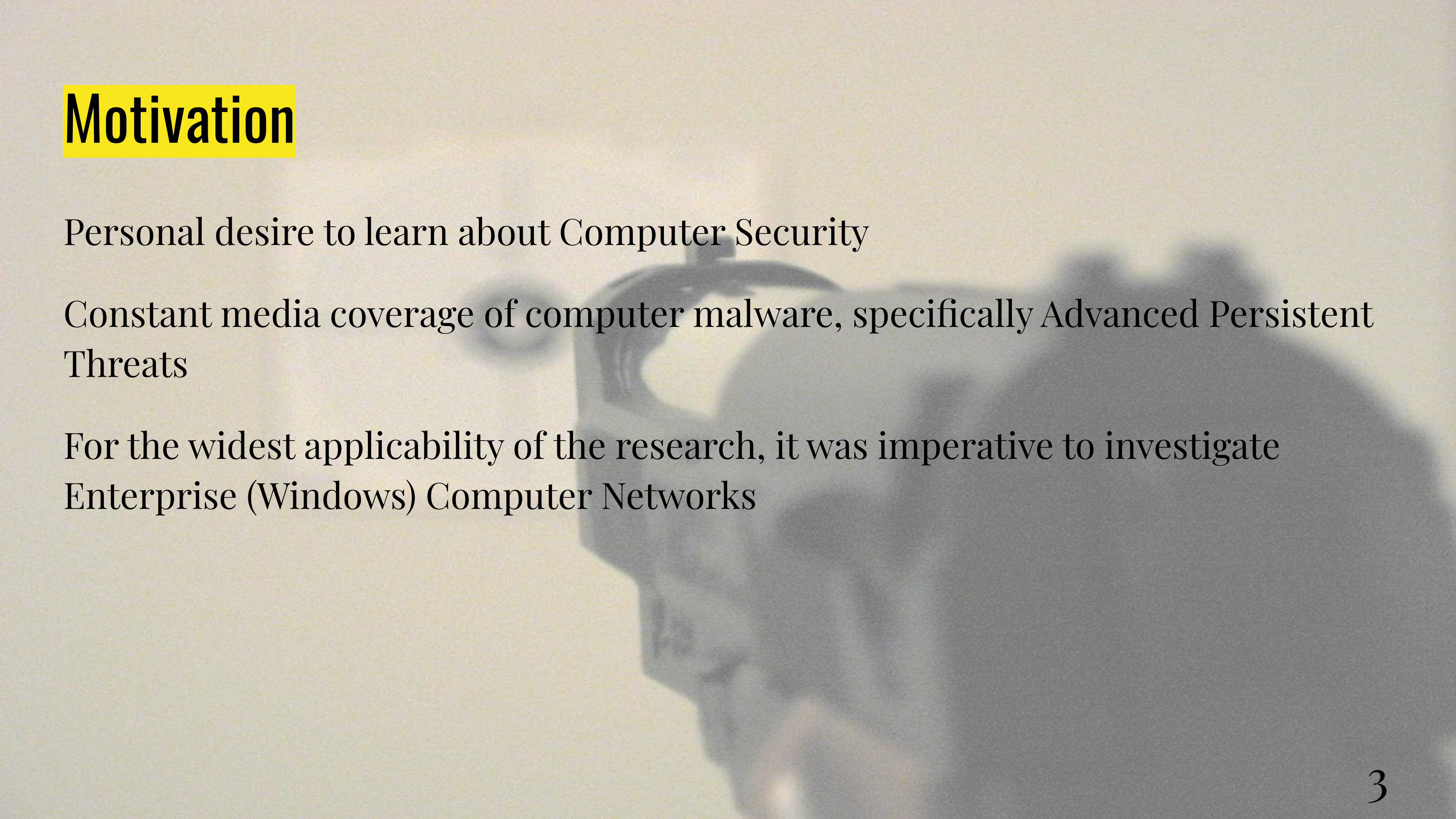
PROJECT PRESENTATION

Wade Wesolowsky
October 25, 2019

Acknowledgements

The Graduate Student Research Fund at Athabasca University for providing financial support to acquire computer hardware and networking components.

This research was not sponsored by any free or open source projects mentioned herein, and use of their representative symbols does not imply endorsement of any kind.



Motivation

Personal desire to learn about Computer Security

Constant media coverage of computer malware, specifically Advanced Persistent Threats

For the widest applicability of the research, it was imperative to investigate Enterprise (Windows) Computer Networks

Advanced Persistent [Security] Threats

- Advanced
    - Sophisticated, multi-stage nature
- Persistent
    - Establish non-volatile, permanent foothold in network
    - Long lasting, slow spread
- Threat
    - Sneaky / undetected exploits of the computer systems

Some Present Conundrums

- Wired networks are becoming “obsolete”
    - Network medium moving to wireless
    - Wireless transmission is “invisible” to our senses
- Valuable data is a target
    - Changing internal data
- New vulnerabilities are being discovered
    - Attacks on BIOS and firmware
- System complexity
- Aging infrastructure
- Malware countermeasures
- Cyberwarfare

APST Lifecycle Models

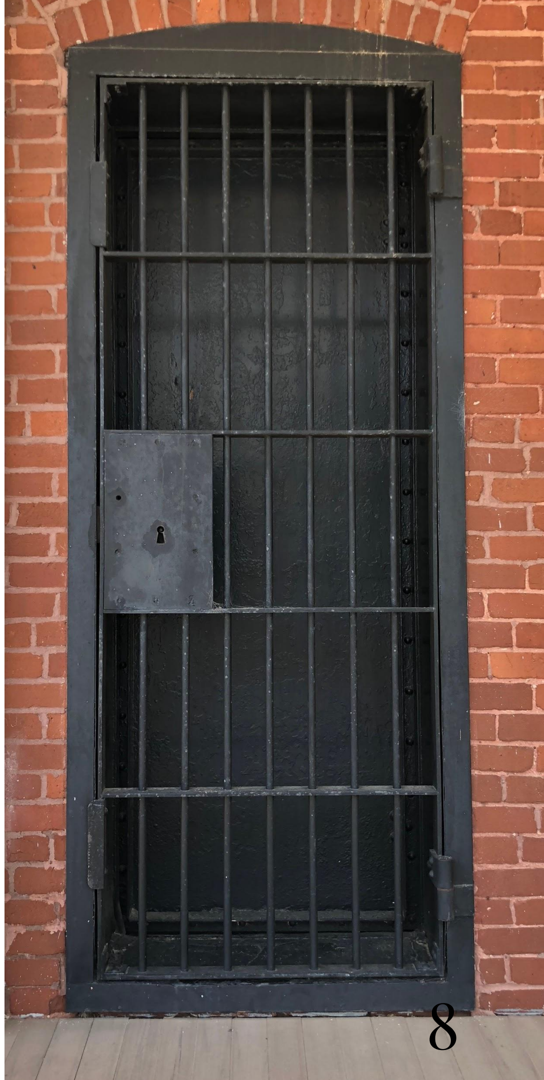
- Various models exist [1] and [2]
  - [1] shows various models and abstracts a recommended model
- We choose to go with [2] for simplicity
- Consists of four stages:
  - Prepare Stage
  - Access
  - Resident Stage
  - Harvest Stage

Prepare Stage

- Collect intelligence on the target
  - Open source intelligence
    - https://www.google.com
    - https://shodan.io
  - Port scanning, etc.
- Investigate ways for initial compromise

Access Stage

- Initial compromise of the fortress perimeter
  - Whaling (C-level Spear phishing)
  - Waterhole Attack (Malicious, frequently visited website)
  - “Lost” USBs
  - Insider threats
  - Large attack surface
  - Default passwords
  - Complex infrastructure
- Using exploits
  - Public vulnerabilities
  - Zero-days (Undisclosed vulnerabilities)



Resident Stage

- Find places to hide in the network
- Bring in more exploit and remote access tools
- Explore the network, expand the web
- Exploit, decide what to steal...

Harvest Stage

- Exfiltrate the valuable data
    - “Phone Home”
- Obfuscation of residual evidence
    - Destroy the evidence of infiltration

Search for Detection Tools

- Need a central way to understand the collected intelligence
- We considered Splunk, but the offering was not free
- ELK / Elastic Stack meets the requirements:
    - Logstash gathers the logs and stores them centrally
    - Elasticsearch indexes the logs and makes them searchable
    - Kibana offers a graphical front-end for tables, graphs, charts, etc.
- Together they are called “ELK” for Elasticsearch, Logstash, and Kibana

Some Desirable Traits for Selected Tools

- Open Source or Free Software
  - Our project was focused on using free and open source software
- Proven Technology
  - Has the solution proven itself in the “real world”
- Active Developer Community
  - Can we get support for questions?
  - Is the software still being developed and refined?
- Fit for Purpose
  - Does it log to the ELK stack?
  - Will it fit with overall design?
- Cost
  - Minimal cost

pfSense (Perimeter Firewall)

- Billed as one of the best Open Source Firewalls
- Under activate development with a stable and well-define following
- Detection of Prepare and Access stages

Snort (runs on pfSense)

- Suricata was a contender
- Snort was a match for our hardware (single processor machine with dual NICs)
- Has been around for a long time
- Detection in Preparation, Resident, Harvest stages

Sweet Security (Zeek)

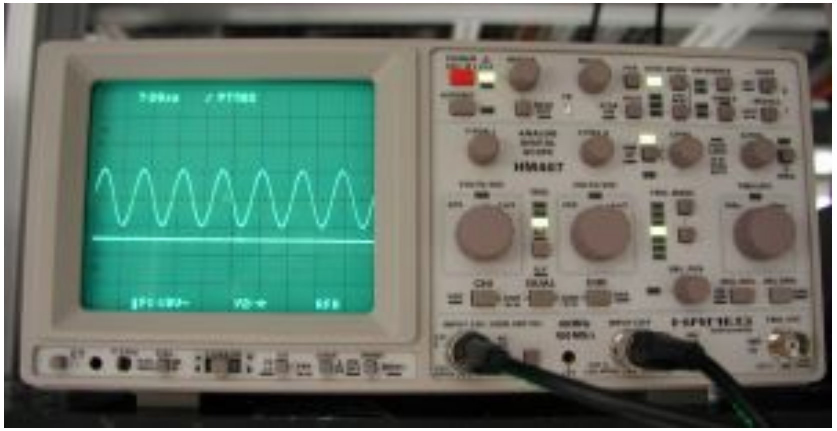
- Ties together Raspberry Pi and Zeek Network Security Monitor (Formerly Bro)
- Allowed us to see what is being transmitted on the network
- A switch with port mirroring helped
- Detection in Access, Resident, Harvest stages

Wazuh (OSSEC fork)

- A open source project built on ELK and and OSSEC.
  - OSSEC client has been upgraded with better functionality, support for more OSes, etc.
- Has "alert" rules which detect predefined signs of compromise or attack
- Does host level monitoring using clients on each monitored OS
  - Data sent back to the ELK stack
- Detection in Resident, Harvest stages

Sysmon

- Provides the ability to monitor system events on Windows
  - E.g. processes, network connections, registry key changes
- Winlogbeat and/or Wazuh can move these logs over to the ELK stack for further analysis
- Detection in Resident stage



HoneyTrap

- Pretends to be a high value target on our network
- You define a configuration file to control the behaviour
- Logs can be sent to the ELK stack
- Detection in Resident stage

Windows Defender

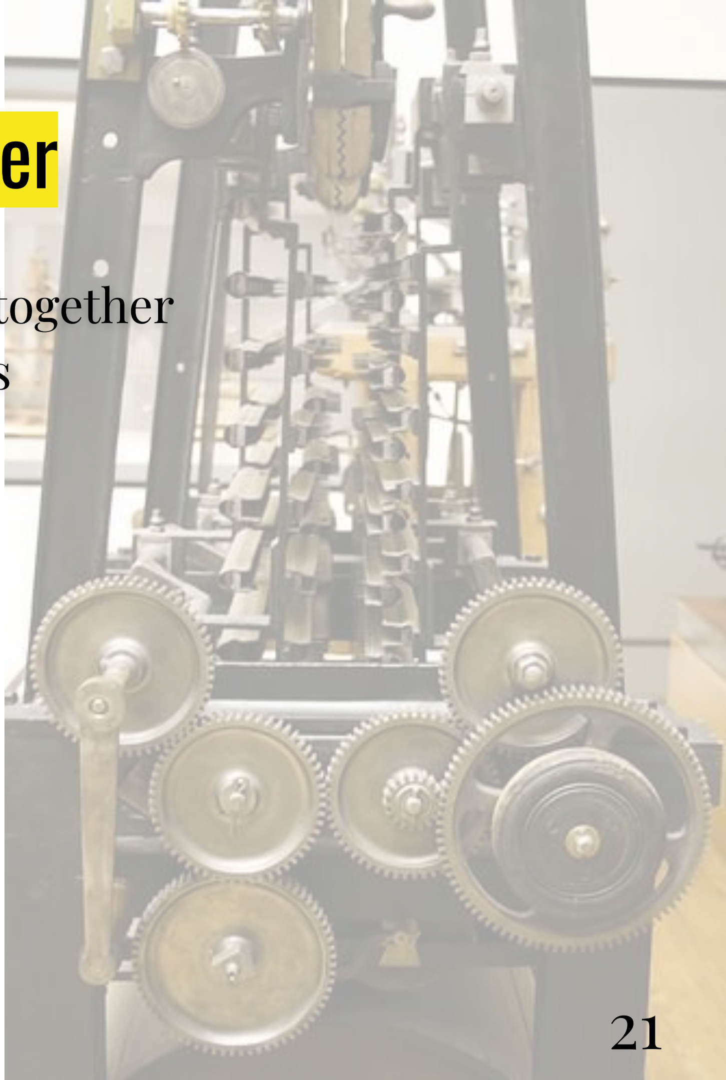
- Unable to find a free/open source virus scanner of sufficient quality
    - ClamAV
- Windows offers a virus scanner which is “free” to use on their OS
- A virus scanner is still an important component of the overall solution
- Assists with detection during the resident phase

Alternative Tools

- We would like to mention three other suites discovered during our research which have similar or overlapping functionality to our solution
  - Elastic SIEM – https://www.elastic.co/blog/introducing-elastic-siem (June 2019)
  - HELK – https://github.com/Cyb3rWard0g/HELK (Dec 2018)
  - AlienVault OSSIM – https://www.alienvault.com/products/ossim (~2003)

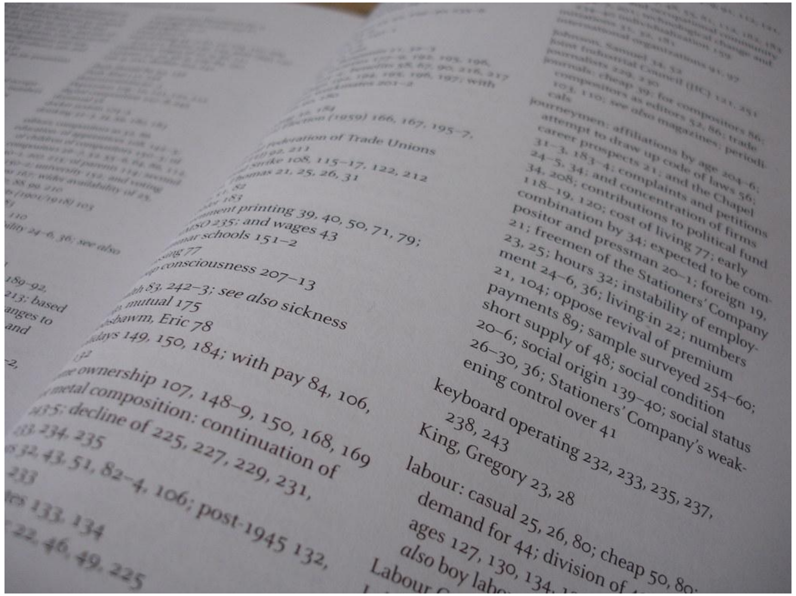
Tools Were Configured to Work Together

- Complex process to get the tools to log and act together
- Our paper includes a brief overview of the steps
  - See references in paper for more information



Indexes

- Log data is stored in Elasticsearch indexes
- Several indexes may exist for one tool
- Allow data to be queried to find results



Indexes

Elasticsearch Index	Source Tool	Type of Logs
honeytrap*	HoneyTrap	Status Messages, Honeypot Connection Attempts
logstash-*	Zeek	Zeek Network Traffic Monitoring
pfsense-*	pfSense Firewall, Snort	Firewall Status Messages, Snort Alerts
sweet_security	Sweet Security	Detected Device Information, Port Scans
sweet_secrutiy_alerts	Sweet Security	New (Unique) Sweet Security Log Events
tardis	Sweet Security	Historial Hosts, IP Addresses, and Websites
wazuh-alerts-3.x-*	Wazuh	Log Events Above the Alert Threshold
wazuh-monitoring-3.x-*	Wazuh	All Wazuh Monitoring Logs
winlogbeats-*	Windows Event Logs	Specific Windows Event Logs (Application, System , Security, Sysmon)

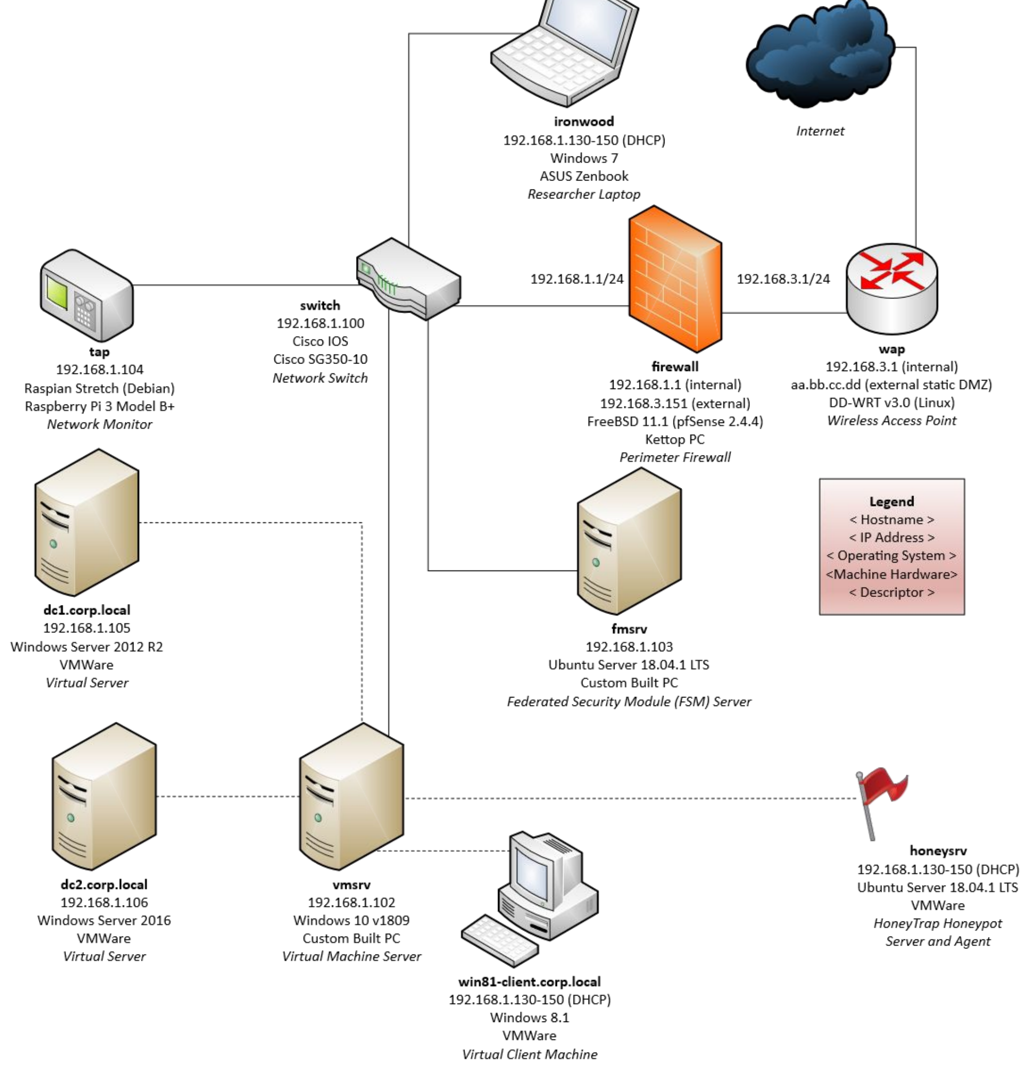
Federation

- Off-the Shelf Tools were adapted to be part of our solution
- They were configured to work together and log to the ELK stack
- Offer overlapping functionality in the various stages of APST attack

Testing

- Spend time connected to research network doing daily computing
- Simulated Enterprise (Windows) Active Directory Domain Environment
- Use tools to simulate APST attacks
    - APTSimulator (Host Level)
    - FlightSim (Network Level)
- Use tests to see examples of data returned by the tools
- Pinpoint the strengths of each tool

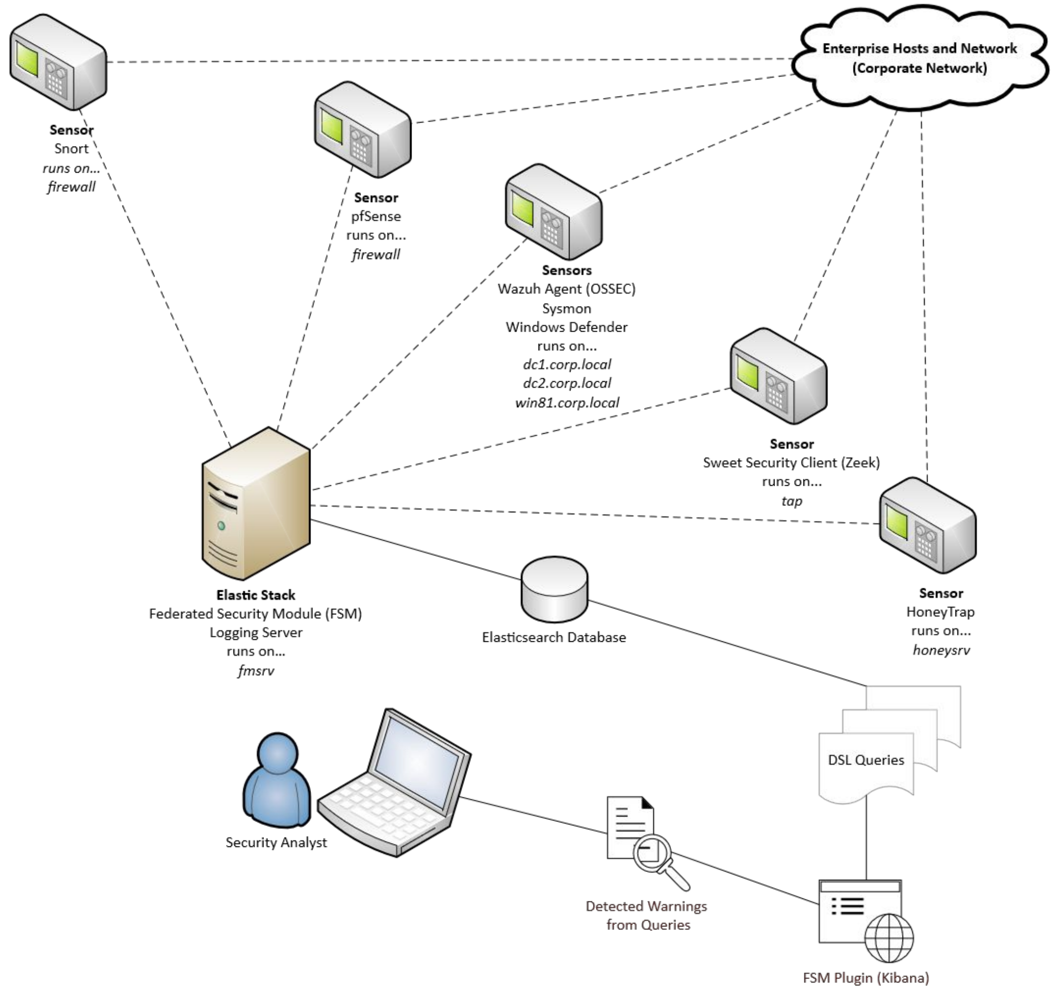
Testing


ironwood
192.168.1.130-150 (DHCP)
Windows 7
ASUS Zenbook
Researcher Laptop
Internet
tap
192.168.1.104
Raspian Stretch (Debian)
Raspberry Pi 3 Model B+
Network Monitor
switch
192.168.1.100
Cisco IOS
Cisco SG350-10
Network Switch
192.168.1.1/24
192.168.3.1/24
firewall
192.168.1.1 (internal)
192.168.3.151 (external)
FreeBSD 11.1 (pfSense 2.4.4)
Kettop PC
Perimeter Firewall
wap
192.168.3.1 (internal)
aa.bb.cc.dd (external static DMZ)
DD-WRT v3.0 (Linux)
Wireless Access Point
Legend
< Hostname >
< IP Address >
< Operating System >
<Machine Hardware>
< Descriptor >
dc1.corp.local
192.168.1.105
Windows Server 2012 R2
VMWare
Virtual Server
fmsrv
192.168.1.103
Ubuntu Server 18.04.1 LTS
Custom Built PC
Federated Security Module (FSM) Server
dc2.corp.local
192.168.1.106
Windows Server 2016
VMWare
Virtual Server
vmsrv
192.168.1.102
Windows 10 v1809
Custom Built PC
Virtual Machine Server
honeysrv
192.168.1.130-150 (DHCP)
Ubuntu Server 18.04.1 LTS
VMWare
HoneyTrap Honeypot
Server and Agent
win81-client.corp.local
192.168.1.130-150 (DHCP)
Windows 8.1
VMWare
Virtual Client Machine

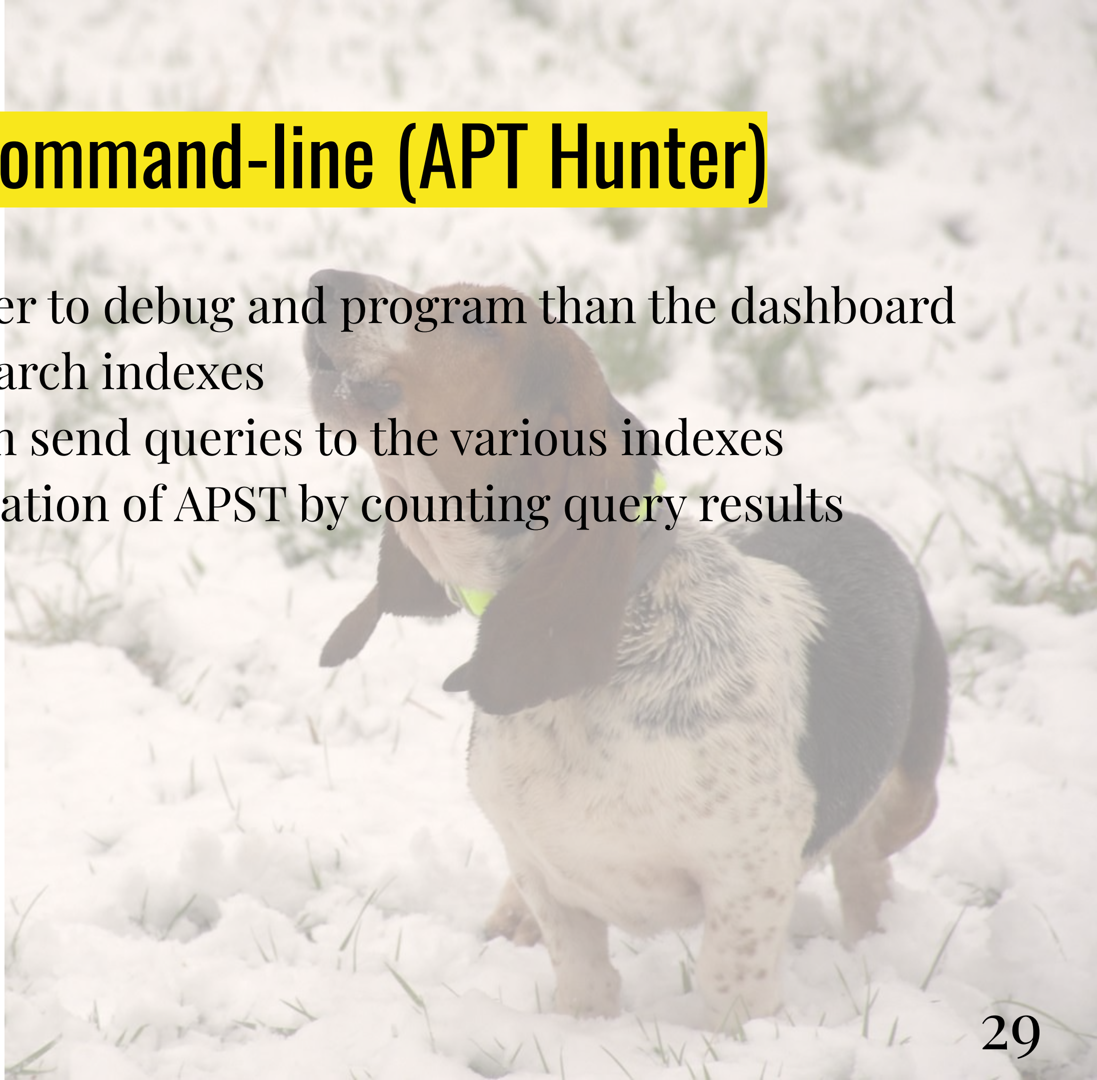
Querying the Data from Kibana (Plugin)

- Shows how some prebuilt queries can return data in Kibana
- Queries are predefined and return results in accessible table format
- Programmed in such a way as to be easily extensible

FSM Plugin Architecture

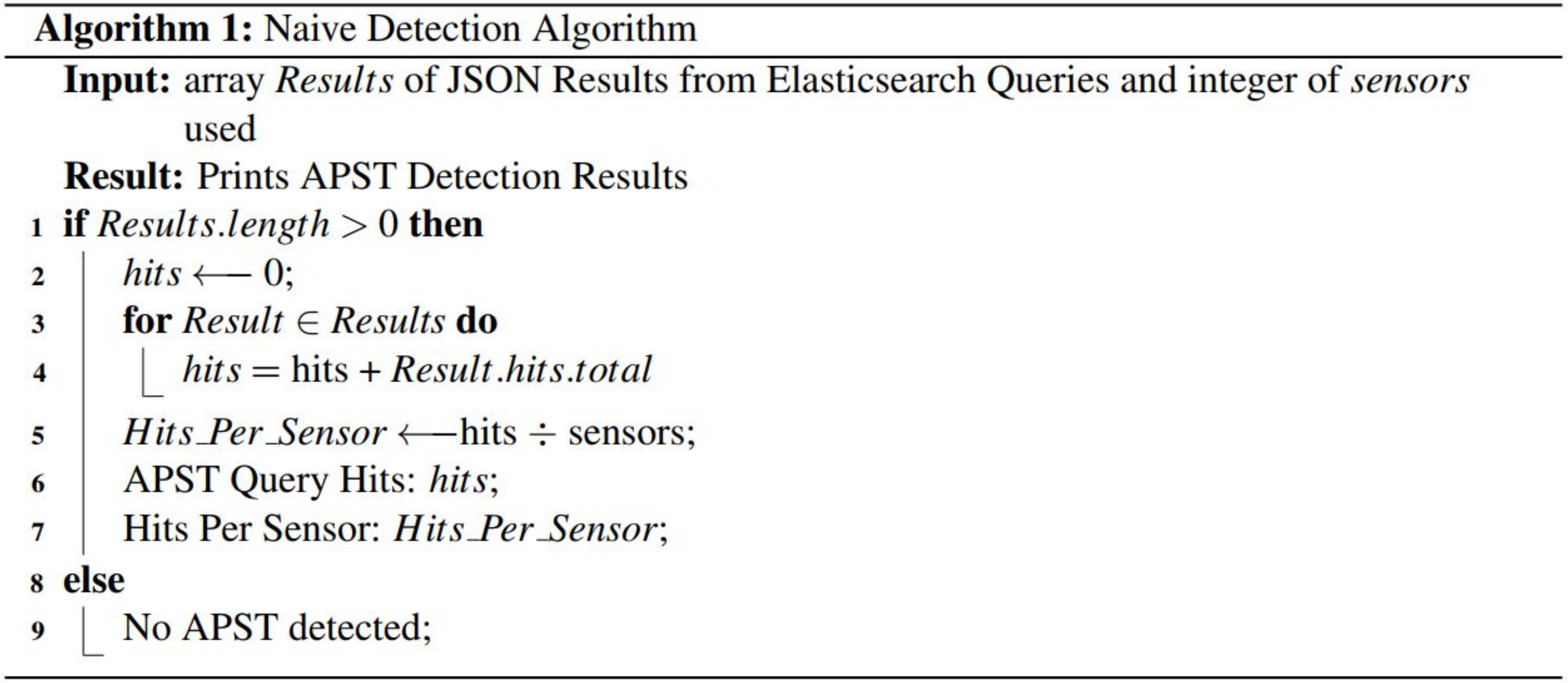


Querying the Data from Command-line (APT Hunter)



- Lower-level tool which is easier to debug and program than the dashboard
- Python tool to query Elasticsearch indexes
- A general query tool which can send queries to the various indexes
- Results returned give an indication of APST by counting query results

Example Detection Algorithm



Algorithm 1: Naive Detection Algorithm

```
Input: array $Results$ of JSON Results from Elasticsearch Queries and integer of $sensors$
       used
Result: Prints APST Detection Results
if $Results.length > 0$ then
    $hits \longleftarrow 0$;
    for $Result \in Results$ do
        $hits =$ hits + $Result.hits.total$
    $Hits\_Per\_Sensor \longleftarrow$ hits $\div$ sensors;
    APST Query Hits: $hits$;
    Hits Per Sensor: $Hits\_Per\_Sensor$;
else
    No APST detected;
```

Federation Continued

- APT Hunter offers the flexibility to query any/all indexes as needed
- Additional detection algorithms can be added, and then command-line arguments can be used to select between them
- Allows the tools to work together as one cohesive unit

Findings

- We have created a free and open source APST detection framework using existing security tools!
- GitHub code repository offers the ability for anyone to build on the work
- Our simulated enterprise network closely resembles a real-world enterprise network
  - Gather lessons learned to expand solution to the real world
- FSM Plugin and APT Hunter are a good starting point for the hunt

Findings

- Updating to the latest tools would help us
    - More features being added all the time to sensors
- Sysmon is very effective on the Windows host systems
    - Helps us track back the chain of events
- Using existing detection technologies would help expand the scope of investigation
    - Yara

Future Work

- Add more sensors to our network
    - SSL/TLS Decryptors
- Automation when updating the dashboard
- Machine learning to replace security analyst
- Active response
    - Alerting (SMS)

Public Dissemination

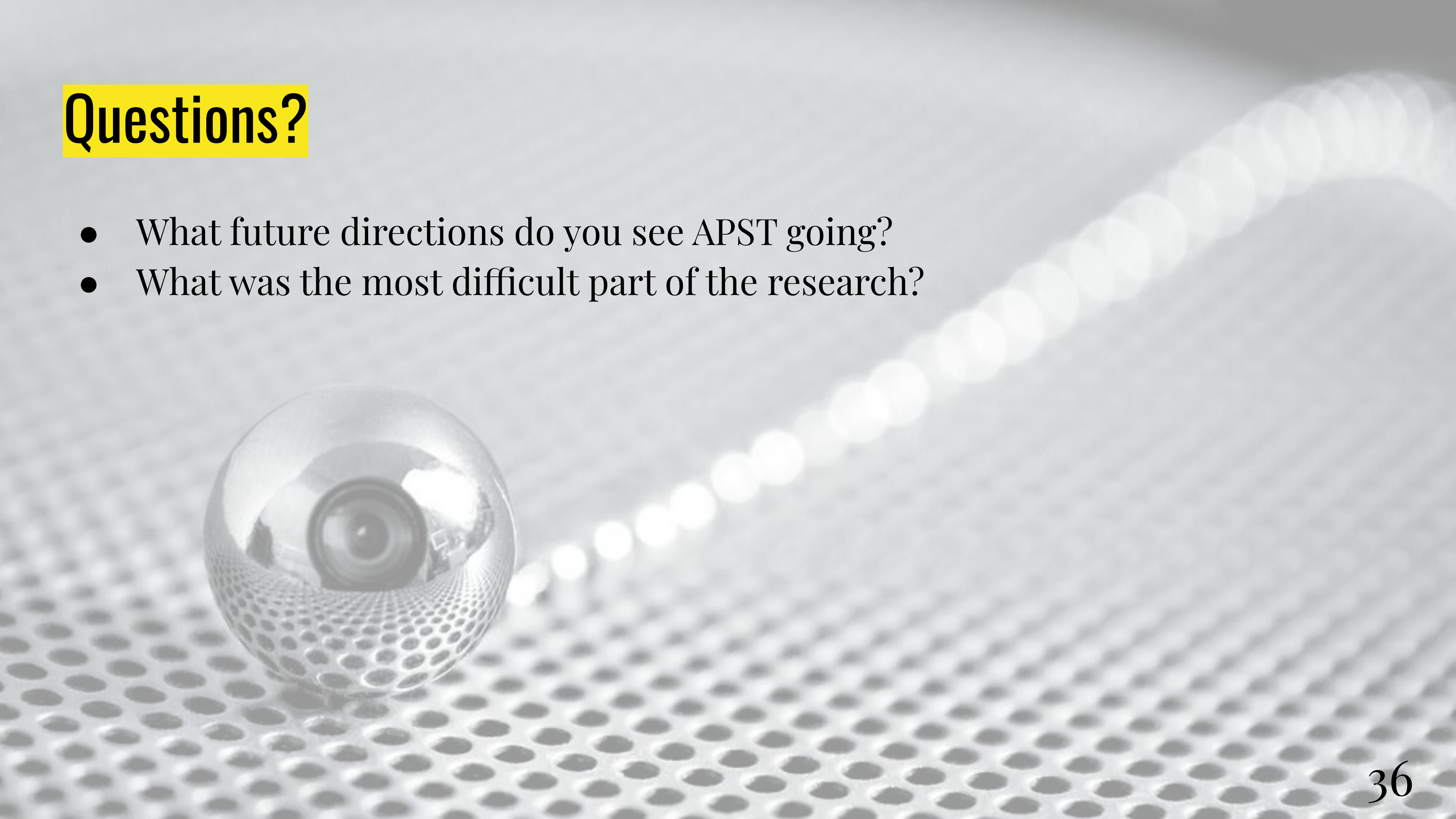
Public Code Repository on GitHub

https://github.com/rndrev/FederatedSecurityModule

This repository includes code used in project, project deliverables, and forked projects which we used or reference during the research.

Questions?

- What future directions do you see APST going?
- What was the most difficult part of the research?



References

[1] B. I. D. Messaoud, K. Guennoun, M. Wahbi, and M. Sadik, “Advanced Persistent Threat: New analysis driven by life cycle phases and their challenges,” in 2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS), Oct 2016, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ACOSIS.2016.7843932

[2] M. Li, W. Huang, Y. Wang, W. Fan, and J. Li, “The study of APT attack stage model,” in 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), June 2016, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICIS.2016.7550947

Image Credits

"File:Celowanie do tarczy 2010 ubt.JPG" by Tsca.bot is licensed under CC BY 3.0

"Circuit Board" by johnmuk is licensed under CC BY-NC-SA 2.0

"Unstructured LAN Patching" by webernetz is licensed under CC BY 2.0

"File:Quercus robur - sprouting acorn.jpg" by Amphis is licensed under CC BY-SA 3.0

"File:Magnifying glass.jpg" by Tomomarusan is licensed under CC BY 2.5

"File:Rockdale County Jail detail of jail entrance at SE corner.jpg" by Krelnik is licensed under CC BY-SA 4.0

"DSC_0093" by bobosh_t is licensed under CC BY-NC 2.0

"File:Phone booth, Lidgett Park Road, Leeds - DSC07610.JPG" by Green Lane is licensed under CC BY-SA 3.0

"File:Snort.jpg" by Tegshbuyan is licensed under CC BY-SA 4.0

Image Credits

"Wazuh_blue.png" by Zenidd is licensed under CC BY-SA 4.0

"Gears, at the Musée des Arts et Métiers" by ArkanGL is licensed under CC BY-NC-SA 2.0

"wp plugins" by Sean MacEntee is licensed under CC BY 2.0

"Basset Hound Crooning" by jeffmgrandy is licensed under CC BY-NC-ND 2.0

"Random Blue Sky with White Clouds" by warpdesign is licensed under CC BY 2.0

"File:2010-03-18 (27) Honey bee, Honigbiene, Apis mellifica.JPG" by VBuhl is licensed under CC BY-SA 3.0

"File:Oszilloscop.png" by Kae-ru is licensed under CC BY-SA 3.0

"Circles" by places_lost is licensed under CC BY 2.0

"Index" by Ben Weiner is licensed under CC BY-ND 2.0