The TLDR Version

Slides 1- 19 are copy pastable to achieve the results demonstrated during our talk

Slides 20-60 were used in the actual presentation

Network Baselines

OVERVIEW

Bropy 3

1. git clone https://github.com/hashtagcyber/bropy3.git
2. cd bropy3
3. vi etc/bropy.cfg
   - Update Protected Network variable
   - Ensure paths are correct for Bro logs/binaries
4. sudo ./bropy3.py
   - Select “Install”
   - Restart Bro
   - Wait a few hours
   - Use the menu to build baseline

Application Baselines

Microsoft Applocker

- Application Identity Service
  - Verifies file attributes
    - If service is not running enforcement will no longer be enforced

- Configuring appidsvc to auto-start

```
sc config appidsvc start=auto
```

```
sc stop appidsvc && sc start appidsvc
```

- Apply to Domain with GP Editor

Computer Configuration>Windows Settings>Security Settings>System Services>Application Identity

Microsoft Applocker

- Verify Service is set to Auto-start

```
PS C:\> Get-Service "Application Identity" | Select-Object Status, Name, DisplayName, starttype

 Status      Name       DisplayName            StartType
 ----------  -----------  -----------------     -------------
Running     AppIDSvc   Application Identity   Automatic
```

Microsoft Applocker

- Putting it all together
  - Gather file information and create new policy

```
PS C:\> Get-AppLockerFileInformation -Directory C:\Windows\System32 -Recurse -FileType exe, script, dll |
New-AppLockerPolicy -RuleType Publisher,Hash -User Everyone -IgnoreMissingFileInformation -
RuleNamePrefix System32 -XML | Out-File .\System32.XML
```

  - Test policy

```
PS C:\> Test-AppLockerPolicy -Path 'C:\Users\Carl.Isdead\Downloads\HxD.exe' -XmlPolicy
'C:\Users\Carl.Isdead\Desktop\System32.xml'

FilePath                                    PolicyDecision MatchingRule
--------                                    -------------- ------------
C:\Users\Carl.Isdead\Downloads\HxD.exe      DeniedByDefault
```

Microsoft Applocker

- Set-Applocker

```
PS C:\> Set-AppLockerPolicy -XMLPolicy C:\System32.xml
```

- Get-GPO

```
Get-GPO -All -Domain zombee.corp | Select-Object DisplayName, Path
```

- Apply to GPO

```
PS C:\> Set-AppLockerPolicy -XMLPolicy C:\System32.xml -LDAP "LDAP://Zom-DC.corp/cn={31B2F340-016D-11D2-945F-00C04FB984F9},cn=policies,cn=system,DC=zombee,DC=corp"
```

Microsoft Applocker

- Additionally you can create a New-Policy from Audited events

```
C:\PS>Get-AppLockerFileInformation -EventLog -LogPath "Microsoft-Windows-AppLocker/EXE and DLL" -EventType Audited | New-AppLockerPolicy -RuleType Publisher,Hash -User Everyone -IgnoreMissingFileInformation | Set-AppLockerPolicy
```

CHIRON TRAINING

Blue Team Sprint

Troopers 18

CHIRON TRAINING

Disclaimer

- We “borrowed” an employers slide template
  - Creating .POT files is hard
- This is NOT any employers material
- TLDR; You can sue us, not our employers

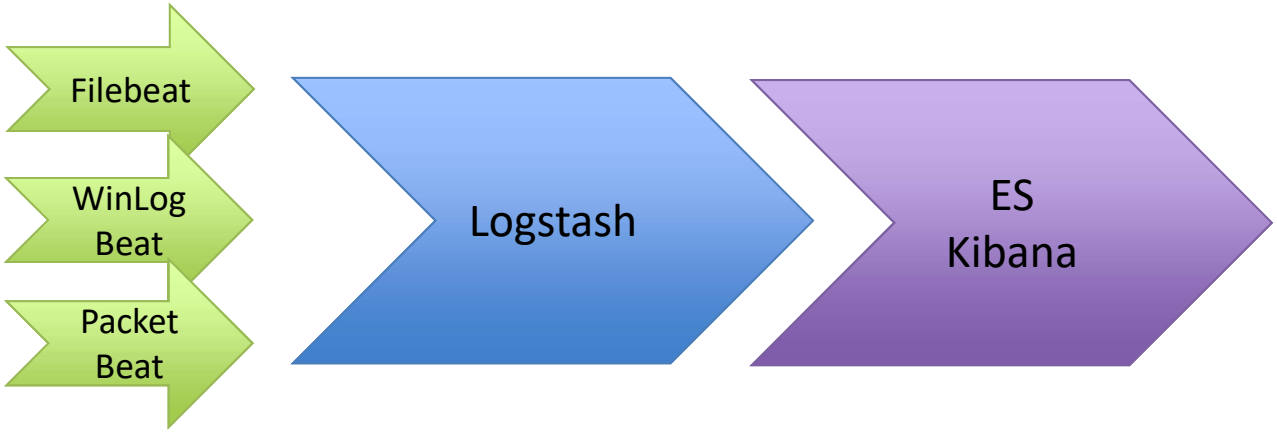
Elastic Stack

OVERVIEW

But I have a Raspberry Pi Budget….

- 3 Tier System
  - ElasticSeach + Kibana Node
  - Logstash for centralized ingestion
  - Beats agent for forwarding to Logstash

- Why this way?
  - Beats agents are multi platform and allow for simple integration
  - Logstash by itself is flexible, connectors for most commercial SIEMs
    - If budget increases, you can switch to $SIEM by changing the Logstash output



#Kitbag : Installing ElasticSearch and Kibana

on Debian9

CHIRON TRAINING

- Elastic has a tutorial
  - https://www.elastic.co/guide/en/elasticsearch/reference/current/setup.html

- TLDR;

sudo apt-get update && sudo apt-get upgrade

sudo apt-get install default-jdk apt-transport-https

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list

sudo apt-get update && sudo apt-get install elasticsearch kibana

sudo sed -i 's/^#network.host.*/network.host : 0.0.0.0/' /etc/elasticsearch/elasticsearch.yml

sudo sed -i 's/^#server.host.*/server.host : 0.0.0.0/' /etc/kibana/kibana.yml

#Kitbag : Installing ElasticSearch and Kibana

on Debian9

CHIRON TRAINING

Continued….

sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
sudo /bin/systemctl enable kibana.service

sudo service elasticsearch start
sudo service kibana start

#Kitbag : Installing Logstash on Debian 9

- Again, Elastic has a great wiki:
  - https://www.elastic.co/guide/en/logstash/6.2/setup-logstash.html

- But, TLDR;

sudo apt-get update && sudo apt-get upgrade

sudo apt-get install default-jdk apt-transport-https

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list

sudo apt-get update && sudo apt-get install logstash

sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable logstash.service

Logstash Config - WinLogBeat

- vi /etc/logstash/conf.d/winlogbeat.conf

```
input {
  beats {
    port => 5044
  }
}
output {
  elasticsearch {
    hosts => ["http://192.168.75.253:9200"]
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
  }
}
```

- sudo service logstash restart

WinLogBeat – Install and Configure

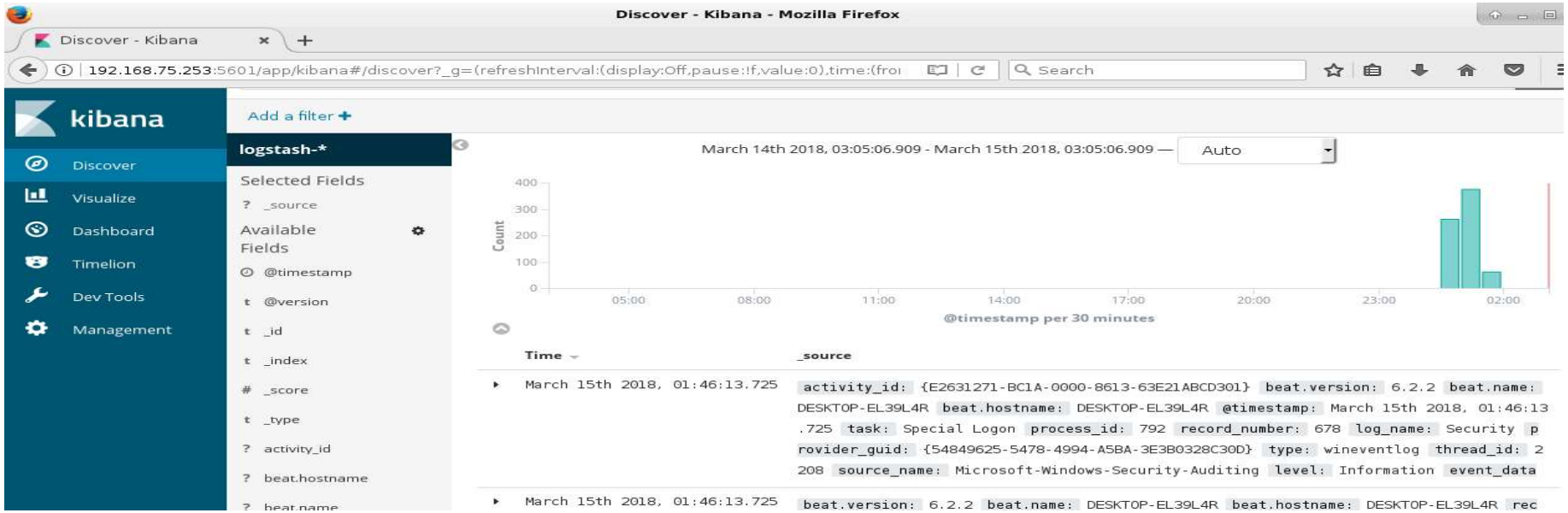
CHIRON TRAINING

- Elastic Wiki
  - https://www.elastic.co/guide/en/beats/winlogbeat/current/winlogbeat-configuration.html
- TLDR;

1. Download and extract the winlogbeat zip file from Elastic
   - https://artifacts.elastic.co/downloads/beats/winlogbeat/winlogbeat-6.2.2-windows-x86_64.zip
2. Edit ./winlogbeat/winlogbeat.yml
   - Comment out all sections relating to ElasticSearch and Kibana
   - Uncomment output.logstash section and fill in the host field with your logstash IP address
3. Re-compress the folder, transfer to client, extract and run “install-service-winlogbeat.ps1”
4. Start-service winlogbeat

Final Step : Configure ElasticSearch Index

- Browse to http://elastic.search.ip:5351
- Click “Configure Index”
- Enter “logstash-*”
- Select “@timestamp” for timestamp
- Profit



Blue Team Sprint

OVERVIEW

The Concept

Network Baselines (Bropy3)

Application Baselines (AppLocker)

ElasticStack

Super Demo

About Us

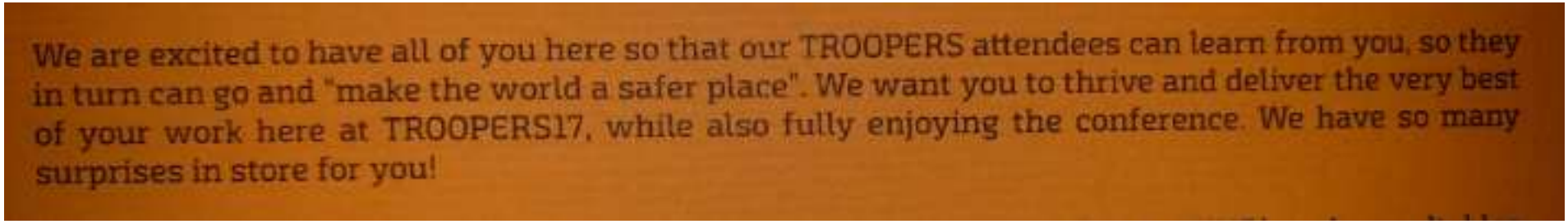
Jordan Salyer

- Beard Enthusiast
- Former:
  - Carpenter
  - Gold Prospector
  - Cyber Network Operator
- Infosec Instructor
- Hiking/Outdoors

Matt Domko

- Beard Enthusiast
- Former:
  - Parachutist
  - Enterprise Admin
  - “Cyber Network Defender”
- Security Engineer at $DayJob
- Brakesec Slack
  https://brakesec.signup.team
- @hashtagcyber

Why We're Here



"Make the world a safer place"
{by sharing information}

Blue Team Sprints

NOT THIS KIND OF SPRINT!

Why YOU are here:

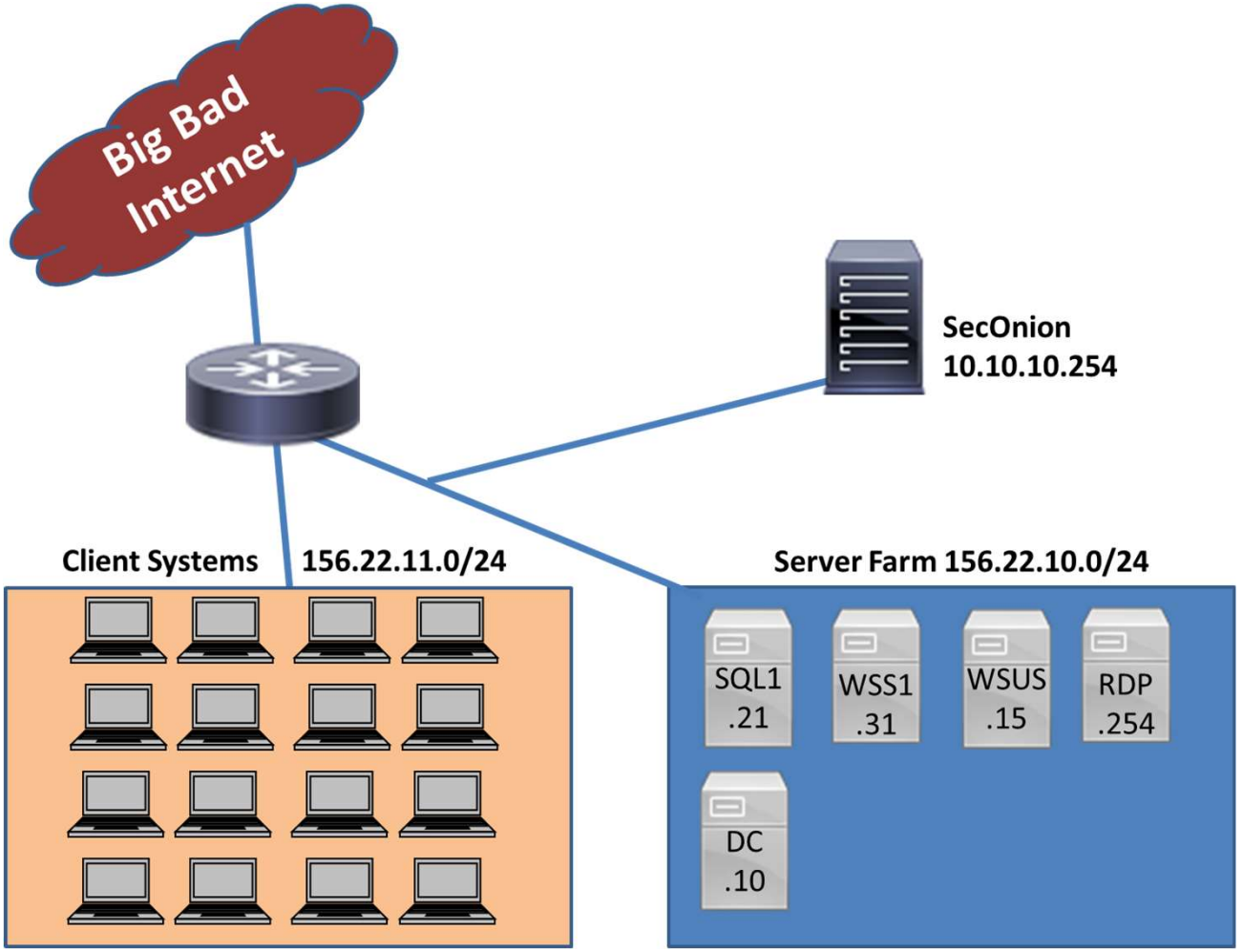
- Not enough time in a day
  - Sorry, can't fix that
- Not enough engineers on your team
  - Sorry, can't fix that
- You want to know more about the packets on your network
  - Bropy3
- You want to spend LESS time resolving skiddy malware
  - Application Whitelisting
- You want a SIEM, but don't have a billion $$$ budget for <redacted>
  - Elastic Stack

The most important thing to me…..

CHIRON TRAINING

WHAT THE HELL IS ON MY NETWORK?

Scenario Network



Network Anomaly Detection : Bropy3

- Start with an empty whitelist
- Apply a policy to log all traffic not in the whitelist
- Use logs to update the whitelist
- Review new logs
  - Investigate new ports/hosts
  - Update whitelist as needed

Network Anomaly Detection: Bropy3

CHIRON TRAINING

- Bropy released at Troopers17
  - https://www.youtube.com/watch?v=VWZ6IggBigE
  - Terrible speaker, checkout the Security Onion Con version instead
    - https://www.youtube.com/watch?v=LzFNOuaYc0g
  - Basically carried off the stage when Enno found out it didn't support IPv6
- Rewrote Bropy in python3, now supporting IPv6
  - It's currently in Alpha, lot's of features still need porting over
- Robin Summer Explains Bro Better @ #TR14 :
  - https://youtu.be/BBl0yaUdq4c

Sample rules

```
#DNS Client
156.22.10.10    53      udp     156.22.10.0/24,156.22.11.0/24
#Kerberos
156.22.10.10    88      tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    88      udp     156.22.10.0/24,156.22.11.0/24
#LDAP
156.22.10.10    389     tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    389     udp     156.22.10.0/24,156.22.11.0/24
#SMB
156.22.10.10    445     tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    445     udp     156.22.10.0/24,156.22.11.0/24
#RPC
156.22.10.10    135     tcp     156.22.10.0/24,156.22.11.0/24
#NetBIOS
156.22.10.10    139     tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    137     udp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    138     udp     156.22.10.0/24,156.22.11.0/24
#Dynamic Ports-NeedToLockItDown
156.22.10.10    49155   tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    49155   udp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    49158   tcp     156.22.10.0/24,156.22.11.0/24
156.22.10.10    49158   udp     156.22.10.0/24,156.22.11.0/24
#Windows Time
156.22.10.10    123     udp     156.22.10.0/24,156.22.11.0/24
```

Use Case

- Generate a list of every port/protocol critical hosts receive connections on
- Receive alerts when non-standard connections are detected
- Baseline data can be used to generate firewall lists

Bropy 3

1. git clone https://github.com/hashtagcyber/bropy3.git
2. cd bropy3
3. vi etc/bropy.cfg
   - Update Protected Network variable
   - Ensure paths are correct for Bro logs/binaries
4. sudo ./bropy3.py
   - Select “Install”
   - Restart Bro
   - Wait a few hours
   - Use the menu to build baseline

My Next Task…..

WHY ARE ALL MY CLIENT SYSTEMS MINING BITCOIN?

Application Baselines & Whitelisting

- Situational Awareness
  - How can you defend your network if you don't know what is there?
    - What services programs do you have in your organization
- Proactive approach to network security
  - You are failing if you are only being reactive
- Defense in Depth
  - This is just one layer

Microsoft Applocker

- What is it?
  - Successor to Software Restriction Policy (can be used concurrently for legacy Windows computers)
  - Part of Microsoft's built-in threat protection products
  - Allows you to control what applications, scripts, and dll's run in your network

- Supported from Windows 7+
  - Full functionality requires Windows 8 Enterprise +

Microsoft Applocker

- Just one layer of defense
- Combined with other solutions can help with the 80%
  - Device Guard or Windows Defender Application Control
  - Antivirus
  - SIEM
  - User Education and Organizational Policies
- Focus your energy on the actual threats

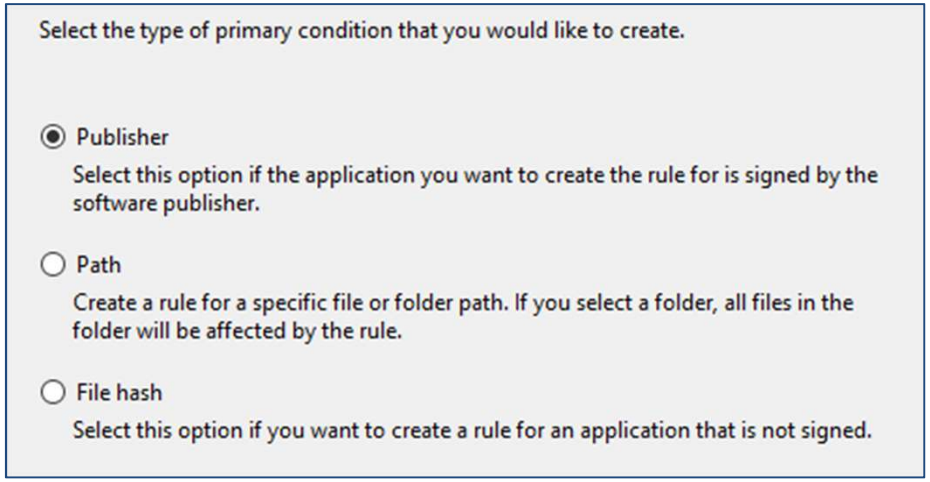
Microsoft Applocker

- Advantages
  - No additional tool cost included with Windows
  - Audit Mode only (more on that in a minute)
  - Manageable through Group Policy Objects
    - Easily import and export GPO's via XML
    - Can be applied to Users and Groups

- Disadvantages
  - Local Event logs only
    - Windows Event Forwarding or SIEM agents like Elastic Beats

Microsoft Applocker

- Rules we can use
  - Publisher
    - Signed Programs
  - Hash
    - Can be difficult to maintain
  - Path
    - Careful with write access



Microsoft Applocker

- Getting started
  - Multiple guides from agency's around the world
    - IAD, NCSC, ASD
- Why reinvent the wheel?
  - Focus your time on tailoring the policy to your needs
    - IAD provides a starter policy

    https://github.com/iadgov/AppLocker-Guidance

    Awesome thing is that it is all xml so it is very easy to verify

Microsoft Applocker

- Understanding what you are whitelisting
  - Use Golden/Trusted Images

- How is your network broken down?
  - Users Groups

Microsoft Applocker

CHIRON TRAINING

- Powershell functionality
  - Get-AppLockerFileInformation
    - Get the file information to create applocker rules or get Applocker event log information
  - Get-ApplockerPolicy
    - Gets the local, effective, or domain applocker policy
  - New-ApplockerPolicy
    - Creates a new applocker policy from file information or Event Log info
    - Generate XML applocker policy

Microsoft Applocker

- Test-ApplockerPolicy
  - Tests file to see if the given policy will affect the execution

- Set-ApplockerPolicy
  - Sets the applocker policy to either a local GPO or Domain GPO if LDAP is specified

Microsoft Applocker

- Application Identity Service
  - Verifies file attributes
    - If service is not running enforcement will no longer be enforced

- Configuring appidsvc to auto-start

```
sc config appidsvc start=auto
```

```
sc stop appidsvc && sc start appidsvc
```

- Apply to Domain with GP Editor

Computer Configuration>Windows Settings>Security Settings>System Services>Application Identity

Microsoft Applocker

- Verify Service is set to Auto-start

```
PS C:\> Get-Service "Application Identity" | Select-Object Status, Name, DisplayName, starttype

 Status      Name       DisplayName            StartType
 ----------  -----------  -----------------     -------------
Running     AppIDSvc   Application Identity   Automatic
```

Microsoft Applocker

- Putting it all together
  - Gather file information and create new policy

```
PS C:\> Get-AppLockerFileInformation -Directory C:\Windows\System32 -Recurse -FileType exe, script, dll |
New-AppLockerPolicy -RuleType Publisher,Hash -User Everyone -IgnoreMissingFileInformation -
RuleNamePrefix System32 -XML | Out-File .\System32.XML
```

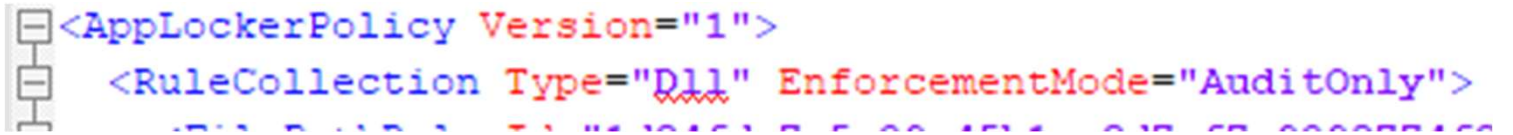
  - Test policy

```
PS C:\> Test-AppLockerPolicy -Path 'C:\Users\Carl.Isdead\Downloads\HxD.exe' -XmlPolicy
'C:\Users\Carl.Isdead\Desktop\System32.xml'

FilePath                                    PolicyDecision MatchingRule
--------                                    -------------- ------------
C:\Users\Carl.Isdead\Downloads\HxD.exe      DeniedByDefault
```

Microsoft Applocker

- Edit Enforcement mode



- Enforcement Mode values
  - NotConfigured
    - Policy created only
  - AuditOnly
    - Will on log events, but nothing is blocked (id 8003 is of interest here)
  - Enabled
    - Policy active and will block what was configured

Microsoft Applocker

- Set-Applocker

```
PS C:\> Set-AppLockerPolicy -XMLPolicy C:\System32.xml
```

- Get-GPO

```
Get-GPO -All -Domain zombee.corp | Select-Object DisplayName, Path
```

- Apply to GPO

```
PS C:\> Set-AppLockerPolicy -XMLPolicy C:\System32.xml -LDAP "LDAP://Zom-DC.corp/cn={31B2F340-016D-11D2-945F-00C04FB984F9},cn=policies,cn=system,DC=zombee,DC=corp"
```

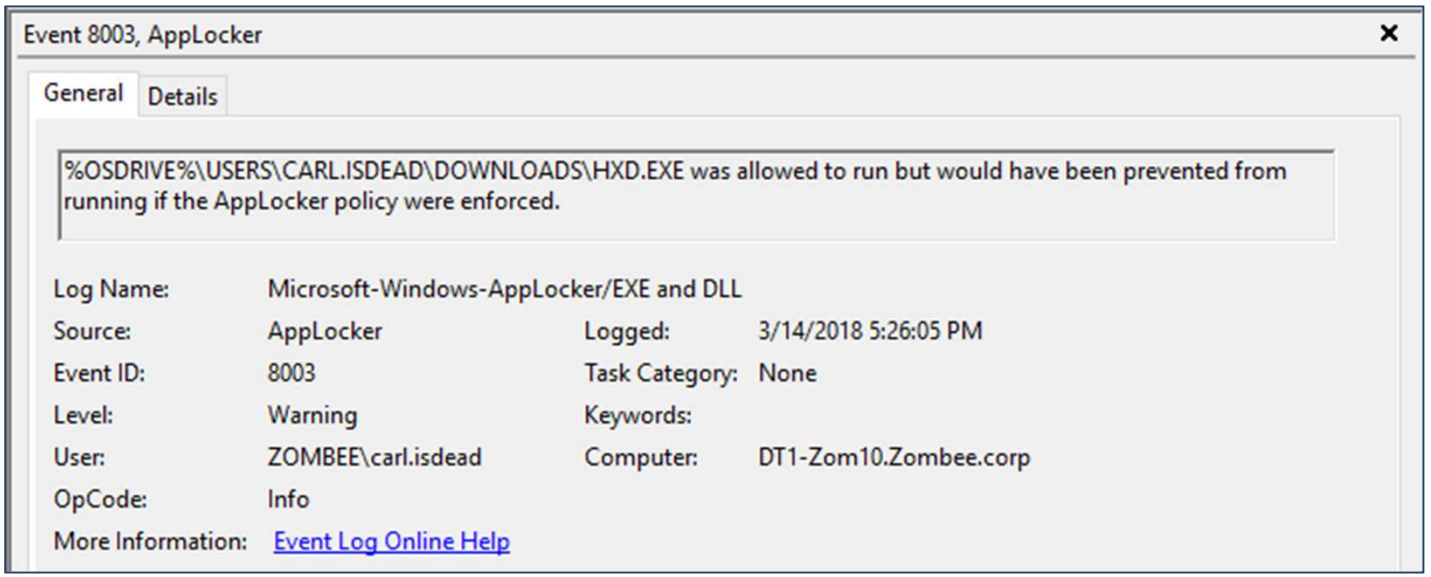
Microsoft Applocker

- Additionally you can create a New-Policy from Audited events

```
C:\PS>Get-AppLockerFileInformation -EventLog -LogPath "Microsoft-Windows-AppLocker/EXE and DLL" -EventType Audited | New-AppLockerPolicy -RuleType Publisher,Hash -User Everyone -IgnoreMissingFileInformation | Set-AppLockerPolicy
```

Microsoft Applocker

- Is just auditing bad?



Event Viewer>Application and Service Logs>Microsoft>Windows>Applocker

And Finally…..

CHIRON TRAINING

DO I REALLY NEED TO USE AWK?

Log Ingestion { on the cheap }

- Elastic Stack:
  - Elastic Search (Does the indexing)
  - Logstash (Normalizes log data)
  - Kibana (Makes pretty charts)

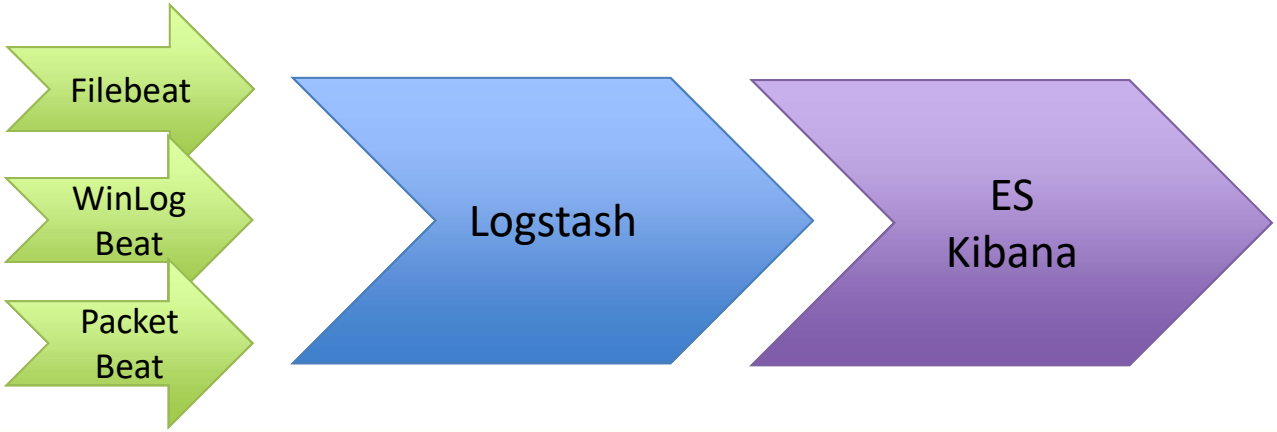
- Lot's of SaaS options:
  - https://www.elastic.co/cloud
  - <Cloud Company> /elasticsearch
  - https://searchly.com
  - https://qbox.io

- A little different, but compatible
  - https://humio.com

But I have a Raspberry Pi Budget….

- 3 Tier System
  - ElasticSeach + Kibana Node
  - Logstash for centralized ingestion
  - Beats agent for forwarding to Logstash

- Why this way?
  - Beats agents are multi platform and allow for simple integration
  - Logstash by itself is flexible, connectors for most commercial SIEMs
    - If budget increases, you can switch to $SIEM by changing the Logstash output



#Kitbag : Installing ElasticSearch and Kibana

on Debian9

CHIRON TRAINING

- Elastic has a tutorial
  - https://www.elastic.co/guide/en/elasticsearch/reference/current/setup.html

- TLDR;

sudo apt-get update && sudo apt-get upgrade

sudo apt-get install default-jdk apt-transport-https

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list

sudo apt-get update && sudo apt-get install elasticsearch kibana

sudo sed -i 's/^#network.host.*/network.host : 0.0.0.0/' /etc/elasticsearch/elasticsearch.yml

sudo sed -i 's/^#server.host.*/server.host : 0.0.0.0/' /etc/kibana/kibana.yml

#Kitbag : Installing ElasticSearch and Kibana

on Debian9

CHIRON TRAINING

Continued….

sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
sudo /bin/systemctl enable kibana.service

sudo service elasticsearch start
sudo service kibana start

#Kitbag : Installing Logstash on Debian 9

- Again, Elastic has a great wiki:
  - https://www.elastic.co/guide/en/logstash/6.2/setup-logstash.html

- But, TLDR;

sudo apt-get update && sudo apt-get upgrade

sudo apt-get install default-jdk apt-transport-https

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list

sudo apt-get update && sudo apt-get install logstash

sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable logstash.service

Ok… But Now What?

- Two(ish) steps remain:
  - Generate Logstash configuration files
    - These tell Logstash what protocols to listen for, and where to send the log data
    - Samples:
      - https://www.elastic.co/guide/en/beats/winlogbeat/master/logstash-output.html
  - Install and configure Beats on endpoints
    - Which logs should be monitored
    - Where is Logstash?

- WinLogBeat Demo

Logstash Config - WinLogBeat

- vi /etc/logstash/conf.d/winlogbeat.conf

```
input {
  beats {
    port => 5044
  }
}
output {
  elasticsearch {
    hosts => ["http://192.168.75.253:9200"]
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
  }
}
```

- sudo service logstash restart

WinLogBeat – Install and Configure

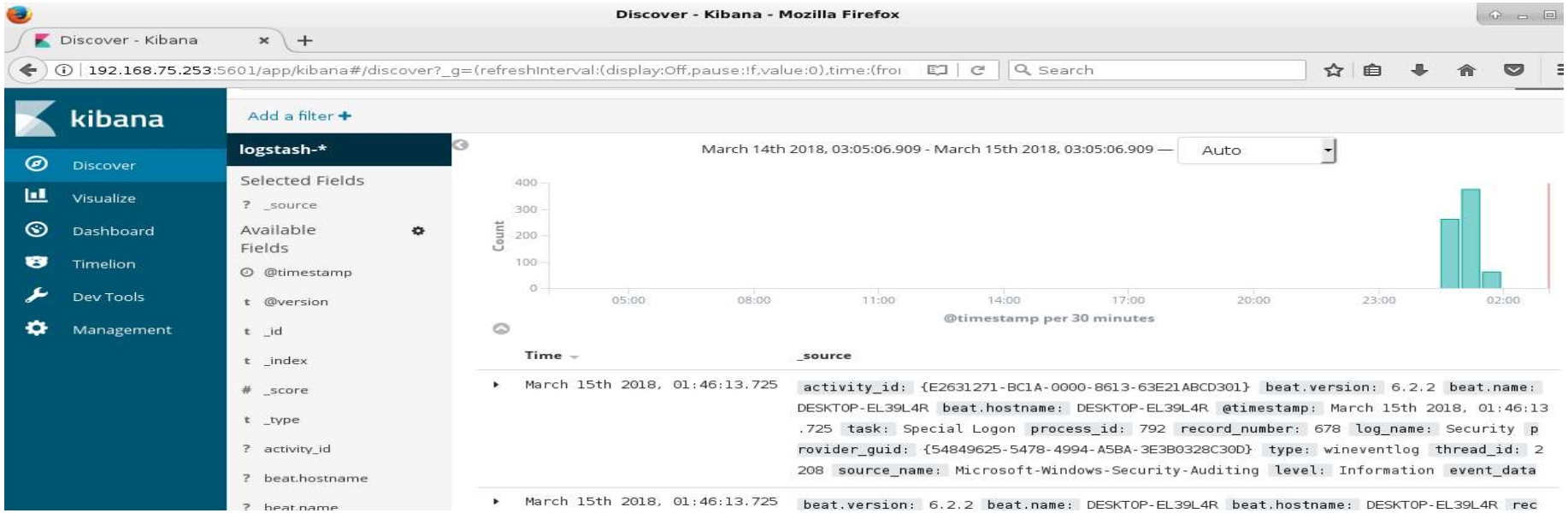
CHIRON TRAINING

- Elastic Wiki
  - https://www.elastic.co/guide/en/beats/winlogbeat/current/winlogbeat-configuration.html
- TLDR;

1. Download and extract the winlogbeat zip file from Elastic
   - https://artifacts.elastic.co/downloads/beats/winlogbeat/winlogbeat-6.2.2-windows-x86_64.zip
2. Edit ./winlogbeat/winlogbeat.yml
   - Comment out all sections relating to ElasticSearch and Kibana
   - Uncomment output.logstash section and fill in the host field with your logstash IP address
3. Re-compress the folder, transfer to client, extract and run “install-service-winlogbeat.ps1”
4. Start-service winlogbeat

Final Step : Configure ElasticSearch Index

- Browse to http://elastic.search.ip:5351
- Click “Configure Index”
- Enter “logstash-*”
- Select “@timestamp” for timestamp
- Profit



One Last Thing….

- SecurityOnion has a version in development that runs ElasticStack instead of ELSA.
  - Everything is configured out of the box
  - Security is built in
  - Pre-built security dashboards
  - You just need to:
    - Use so-allow utility to allow incoming traffic on port 5044
    - Configure WinFileBeat to send traffic to SecurityOnion
    - https://github.com/Security-Onion-Solutions/security-onion/wiki/Beats

Super Awesome Demo Time

- See title

CHIRON TRAINING

Thank you!