SANS

Untapped Potential

Justin Henderson (GSE # 108) and John Hubbard

@SecurityMapper @SecHubb

About Us

Justin Henderson

- Author - SEC555
- Co-Author - SEC455, SEC530
- GSE #108 / Cyber Guardian Blue + Red / 61 certs
- Owner of H & A Security Solutions
- **Twitter**: @SecurityMapper

John Hubbard

- Author - SEC450
  - ***New course!*** *Blue Team Fundamentals – Security Operations & Analysis*
- Co-author SEC455 , Instructor - SEC555 / SEC511
- Owner of Blueprint Cyber Security
- **Twitter**: @SecHubb

Welcome!

A copy of this talk is available at:
https://github.com/HASecuritySolutions/presentations

More free stuff:
https://github.com/HASecuritySolutions

Untapped Potential

Writing without vowels may still be understandable

- But it is widely inefficient. Same true for SOC/SIEM

✓ **A - Automation**

✓ **E - Enrichment**

✓ **I - Identify**

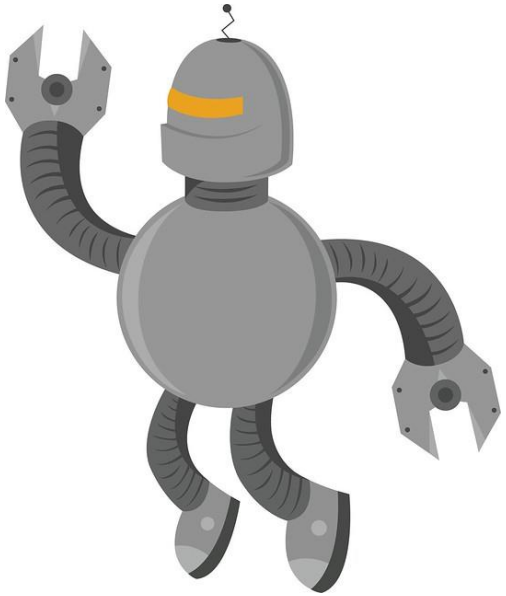
✓ **O - Orchestration**

✓ **U - Universalize**

Which are all enabled by **Y** ---> **You**

A is for

Automation



NXLog AutoConfig

Created to overcome log agent deficiencies and as a functional proof of concept:

https://github.com/SMAPPER/NXLog-AutoConfig

Checks systems each day looking for components (IIS, etc.)

- If found, automatically configures for consistency
  - Or initial configuration…
- Then, sets up an agent to start shipping logs

Largest deployment maintained > 12 K systems

Custom Logging with PowerShell

PowerShell makes writing custom logs easy!

- Create new log sources
- Push logs to custom Windows event channel

Example: Want to log Autoruns items?

```
PS > New-EventLog -LogName "My-Autoruns" -Source "Autoruns"
```

```
PS > autorunsc.exe -c  > autoruns_out.csv
```

```
PS > $items = Import-Csv -Header "Time","Entry
Location","Entry","Enabled","Category","Profile","Description","Compan
y","Image Path","Version","Launch String" .\autoruns_out.csv
```

```
PS > $items | ForEach-Object {Write-EventLog -LogName "my-autoruns" -
source "Autoruns" -EventId 1 -EntryType Information -Message $_}
```

Autoruns in Windows Event Log

Event Viewer

File Action View Help

Event Viewer (Local)
- Custom Views
- Windows Logs
- Applications and Services Logs
  - Hardware Events
  - Internet Explorer
  - Key Management Service
  - Microsoft
  - My-Autoruns
  - ThinPrint Diagnostics
  - Windows PowerShell
- Saved Logs
- Subscriptions

My-Autoruns Number of events: 112

Level	Date and Time	Source	Event ID	Task Category
Information	11/6/2018 1:29:37 PM	Autoruns	1	(1)
Information	11/6/2018 1:29:37 PM	Autoruns	1	(1)
Information	11/6/2018 1:29:37 PM	Autoruns	1	(1)
Information	11/6/2018 1:29:37 PM	Autoruns	1	(1)
Information	11/6/2018 1:28:47 PM	Autoruns	1	(1)
Information	11/6/2018 1:28:47 PM	Autoruns	1	(1)
Information	11/6/2018 1:28:47 PM	Autoruns	1	(1)
Information	11/6/2018 1:28:47 PM	Autoruns	1	(1)
Information	11/6/2018 1:28:47 PM	Autoruns	1	(1)
Information	11/6/2018 1:28:47 PM	Autoruns	1	(1)
Information	11/6/2018 1:28:47 PM	Autoruns	1	(1)

Event 1, Autoruns

General Details

```
@{Time=3/28/2017 1:19 AM; Entry Location=HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run;
Entry=OneDrive; Enabled=enabled; Category=Logon; Profile=SEC511\student; Description=Microsoft OneDrive;
Company=Microsoft Corporation; Image Path=c:\users\student\appdata\local\microsoft\onedrive\onedrive.exe;
Version=17.3.6799.327; Launch String="C:\Users\student\AppData\Local\Microsoft\OneDrive
\OneDrive.exe" /background}
```

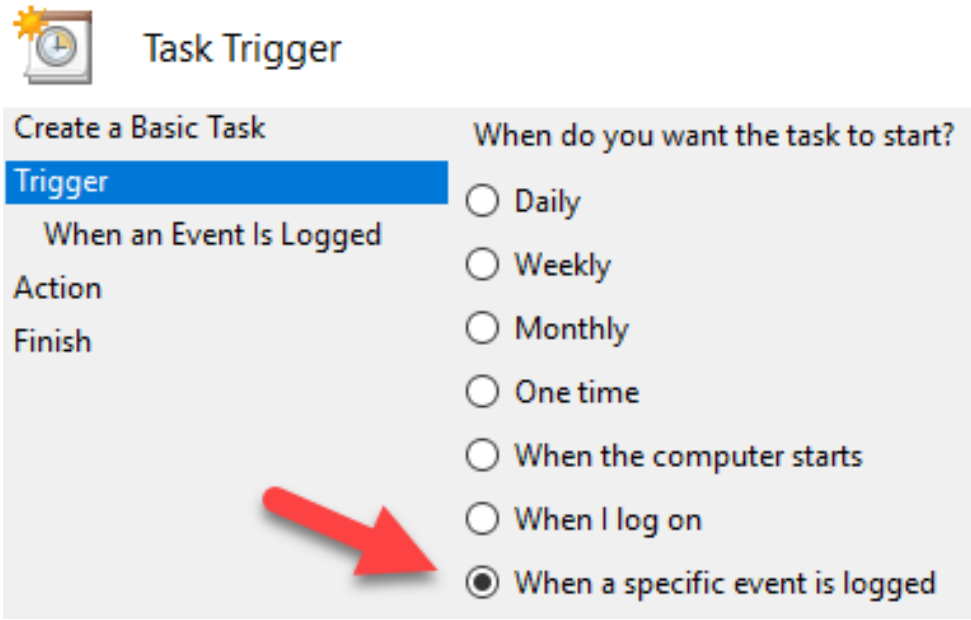
What if a log lacks context… ? Build a better one

Windows Task Scheduler + PowerShell

- Reads native block log
- Runs custom checks against blocked binary
- (Optional) Sends binary to sandbox
- Generates new Windows log

Possible to use new log within SIEM:

- Auto update GPO and notify user



E is for…

Enrichment

Enrichment Example

Which would you rather investigate?

IDS Alert # **1**
or
IDS Alert # **2**

#1

Signature: ET POLICY PE EXE or DLL Windows file download
SID: 2000419
Classification: Potential Corporate Privacy Violation
Source IP: 74.125.159.56
Source Port: 80
Destination IP: 192.168.2.40

#2

Signature: ET POLICY PE EXE or DLL Windows file download
SID: 2000419
Classification: Potential Corporate Privacy Violation
Source IP: 74.125.159.56
Source Port: 80
Destination IP: 192.168.2.40

Geo: US
ASN: Google Inc.
DNS: dl.google.com
Process: iexplore.exe
User: jhenderson
File: ChromeSetup.exe

Domain Parsing

Simple breakdown of fields can yield MANY detection opportunities!

`http://myaccount.google.com-securitysettingspage.tk`

Subdomain field

Parent domain field

TLD

Lots of opportunity for detection!

github.com/HASecuritySolutions/tld_pattern_calculator

tld_pattern_calculator

This project exists to help genarate regex patterns for SIEMs to break up a domain into proper subcomponents such as **subdomain**, **parent_domain**, and various **tlds**. This aids in analyst searching as well as the application of enrichment techniques.

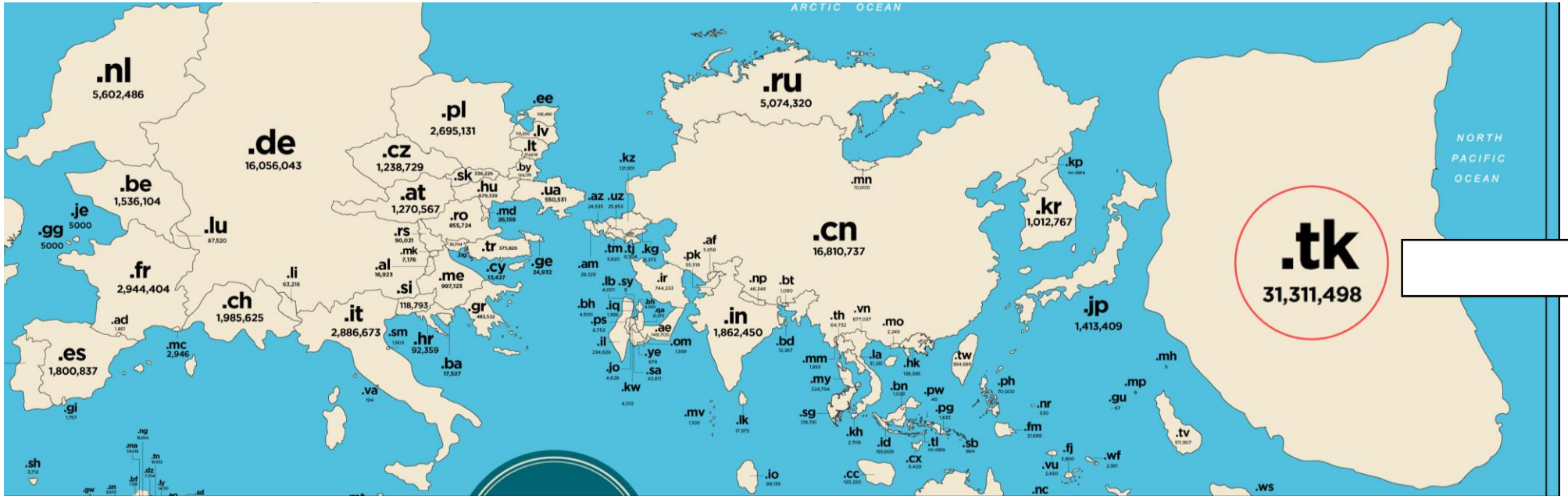
This command runs the **generate_tld_regex.ps1** script which attempts to pull down online lists of TLDs and directly builds a regex pattern file in a file called **tld_patterns.txt**.

```
powershell.exe -File C:\users\user\Downloads\tld_pattern_calculator\generate_tld_regex.ps1 -ExecutionPolicy Bypass
```

```
GTLD aaa|aarp|abarth|abb|abbott|abbvie|abc|able|abogado|abudhabi|academy|accenture|accountant|accountants|aco|acti
CCTLD ac|ad|ae|af|ag|ai|al|am|an|ao|aq|ar|as|at|au|aw|ax|az|ba|bb|bd|be|bf|bg|bh|bi|bj|bl|bm|bn|bo|bq|br|bs|bt|bv|
STLD aero|asia|cat|coop|edu|gov|int|jobs|mil|museum|post|tel|travel|xxx
GRTLD biz|name|pro
REGISTEREDDOMAIN %{WORD:parent_domain}\.(%{GTLD:gtld}|%{GRTLD:grtld}|%{STLD:stld})(\.%{CCTLD:cctld})?$
```

Detection Based on TLD

Some TLDs are just more evil….dashboard it!



https://www.nominet.uk/mapping-the-online-world/

Detection Based on Subdomain

Subdomain field:

1. Does it contain the word Google, or your org name?
2. Long / random, changing and NOT a CDN?
3. 1000's of different subdomains per parent? Tunneling!

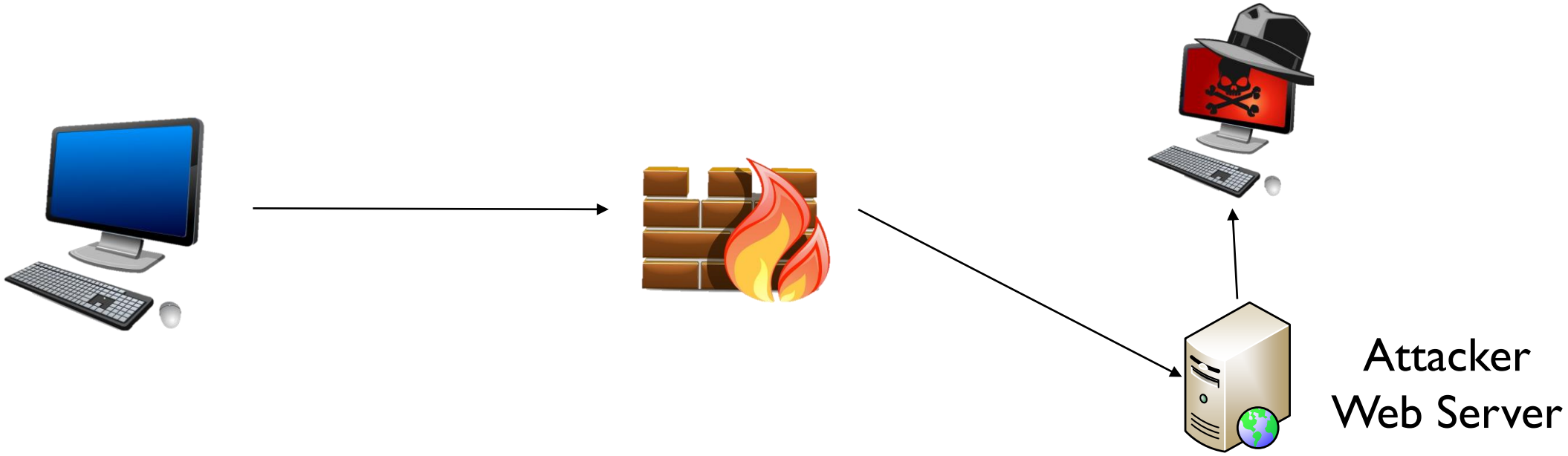
Phishing alert pseudo-logic examples:

- If subdomain = *google*, parent domain != google.com
- If subdomain contains google, ASN != Google
- Group by parent domain, if unique subdomains > 500

Spear Phishing Link

Email Body: Check out this **client's site** before our call

Links To: `http://afecrej6h7cn5sdfhvjg9evmj.com`



freq_server.py

How about natural language processing of select fields?

- Scores the likelihood something is "weird"

Manual testing

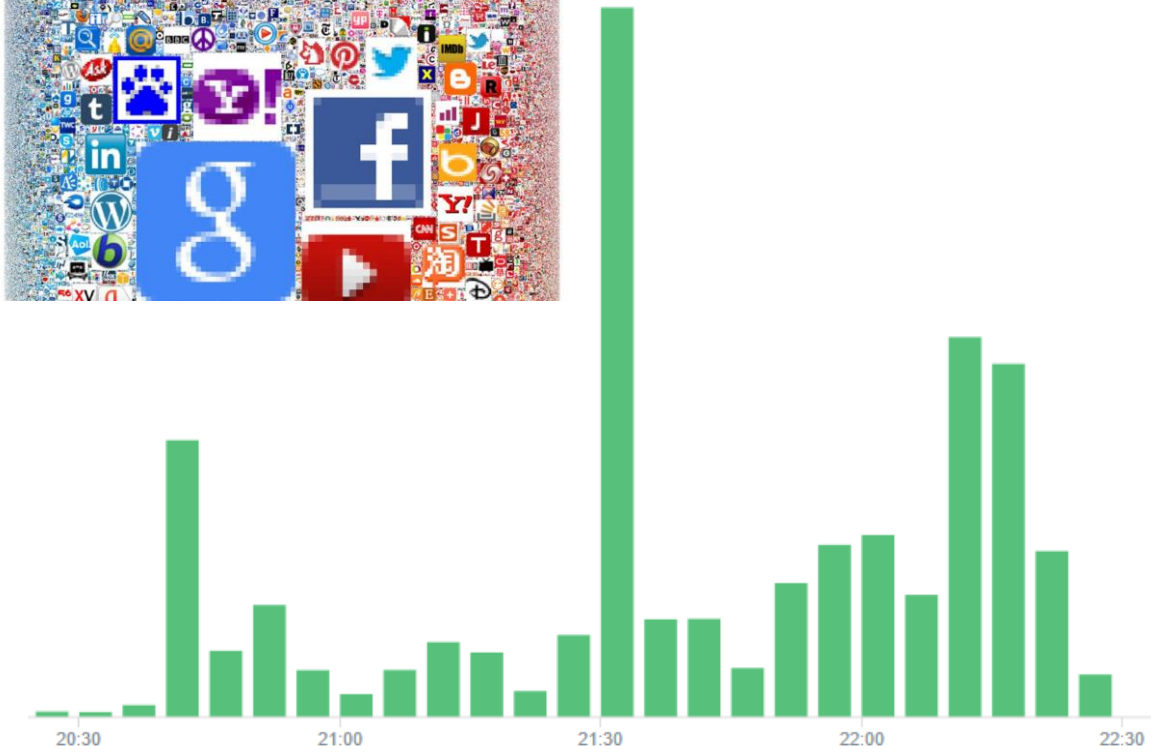
```
curl http://127.0.0.1:10004/measure/google.com
18.2778257342
```

Logstash query

```
rest {
  request => {
    url => "http://localhost:10004/measure/%{highest_registered_domain}"
  }
  sprintf => true
  target => "domain_frequency_score"
}
```

Top I M Filtering

Before



After - approx < 90% logs

Enrichment via SIEM auto-correlation

Signature: Something bad happened
SID: 2000419
Classification: Potential Corporate Privacy Violation
Source IP: 74.125.159.56
Source Port: 80
Destination IP: 192.168.2.40

Does **74.125.159.56** exist in prior DNS logs (answer field)

- Pull back DNS query (**dl.google.com**)

Any logs that have **network socket** to process/user?

- **Sysmon** event ID 3 (pulls in **jhenderson** and **iexplore.exe**)

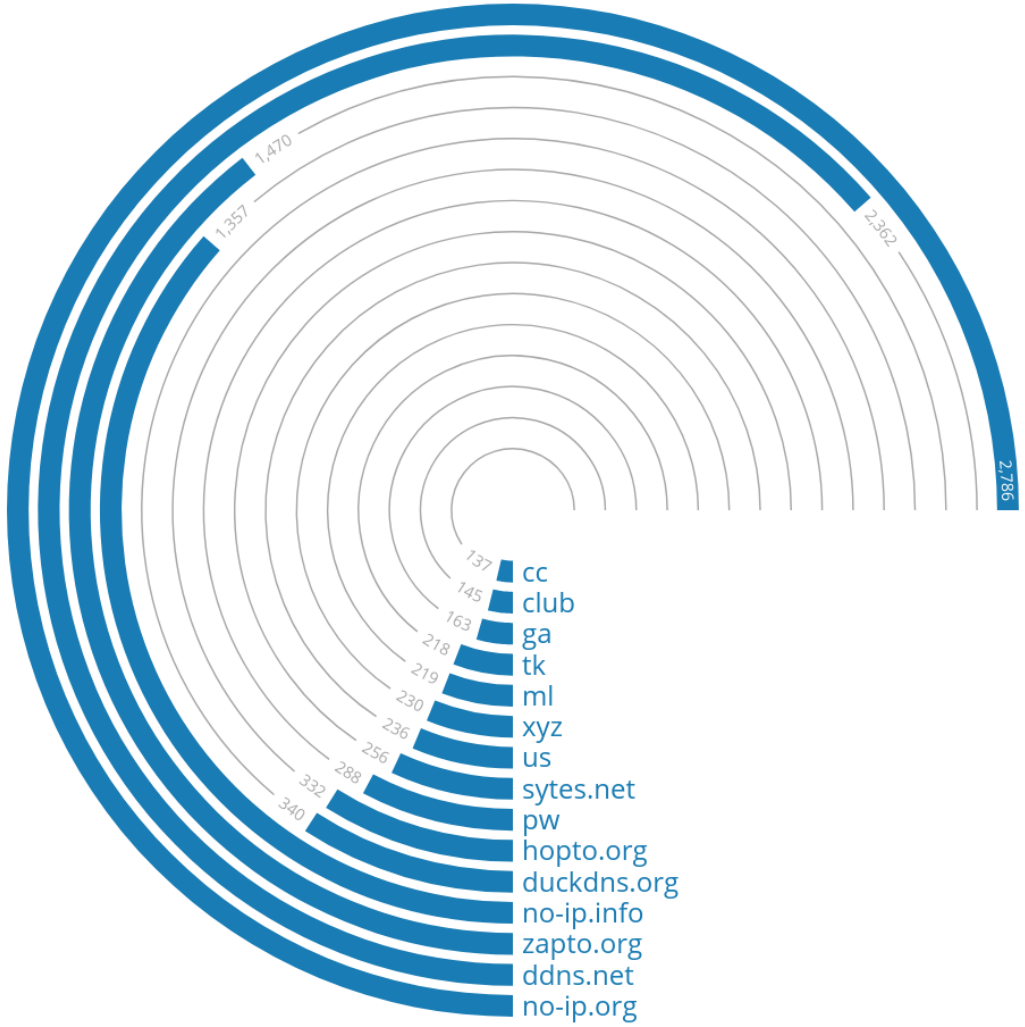
Dynamic DNS Domains

Dynamic DNS Services

- VERY often used for malware
- VERY unlikely to be legitimate business site
- Often used for policy violation

Detect via blacklist!

- Or just block access via DNS RPZ block based on nameserver



https://medium.com/alphasoc/a-deeper-look-at-dangerous-tlds-19f9e3e77926

Autonomous System Numbers (ASN)

Attaches an organization name to an IP address

- Makes geolocation data *better*
- Gives context on downloads

Use case: Which is more suspicious?

User downloads file `chromesetup.exe` from ASN…

1. **Google LLC**
2. **No.31/Jin-rong Street**

SPAMHAUS

The 10 Worst Botnet ASNs

As of 30 September 2019 the world's worst botnet infected Autonomous System Numbers are:

1	**AS4134** No.31/Jin-rong Street	Number of Bots: 1146630

I is for

Identify

Identify All the Things

Easy right? What do you look for?

One Good Rule

Step # 3 - Credential Theft and Reuse

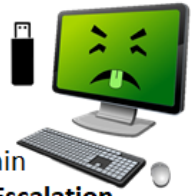
Stolen credentials and reuse
Security - 4624 (successful login)
Security - 4625 (failed login)

Step # 2 - Persistence

Backdoor installed
System - 7045 (new service)
System - 4698 (scheduled task)
System - 4657 (registry audit)

Step # 4 - Backdoor local account

New local user added then added to Administrators group
Security - 4720 (new user)
Security - 4732 (added to local group)



Step # 1 - USB Compromise

PowerShell C2 launch from virtual keyboard
USB device

Pass-the-hash
Local account not your domain
Service Account / Privilege Escalation
Source address invalid for account
Ransomware
Too many logins

Detection Source: Windows Event Logs
PowerShell - 4104
DriverFrameworks-UserMode - 2003
System - Possibly 7045 (new service)

Step # 5 - Logs cleared

Logs cleared from both systems
Security - 1102
System - 104

Advanced attack versions of scenario such as using Mimikatz to clear logs requires alternative methods such as:
Sysmon - 1 process creation monitoring
Sysmon - 7 DLL monitoring
PowerShell Logging - 400, 4103, 4104

MITRE ATT&CK[1]

Adversarial **T**actics, **T**echniques, **and C**ommon **K**nowledge

- Focus is on actionable detection techniques
- Given common adversarial methodologies

Framework is high-level enough to report on and adapt

- Yet specific enough to provide actual items to look for
- Source of specific detection rules

MITRE ATT&CK Matrix

Initial Access	Execution	Persistence	Privilege Escalation	Defe
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Acc Ma
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	B
Hardware Additions	Command-Line Interface	Account Manipulation	AppCert DLLs	Bina
Replication Through Removable Media	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass
Spearphishing Attachment	Control Panel Items	AppInit DLLs	Application Shimming	
Spearphishing Link	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	Clear Co
Spearphishing via Service	Execution through API	Authentication Package	DLL Search Order Hijacking	Co
Supply Chain Compromise	Execution through Module Load	BITS Jobs	Dylib Hijacking	Compi
Trusted Relationship	Exploitation for Client Execution	Bootkit	Exploitation for Privilege Escalation	Compo
Valid Accounts	Graphical User Interface	Browser Extensions	Extra Window Memory Injection	Component Object Model Hijacking	Input Capture	Permission Groups Discovery

Detection

There are many ways to perform UAC bypasses when a user is in the local administrator group on a system, so it may be difficult to target detection on all variations. Efforts should likely be placed on mitigation and collecting enough information on process launches and actions that could be performed before and after a UAC bypass is performed. Monitor process API calls for behavior that may be indicative of Process Injection and unusual loaded DLLs through DLL Search Order Hijacking, which indicate attempts to gain access to higher privileged processes.

Some UAC bypass methods rely on modifying specific, user-accessible Registry settings. For example:

- The `eventvwr.exe` bypass uses the `[HKEY_CURRENT_USER]\Software\Classes\mscfile\shell\open\command` Registry key. [6]
- The `sdclt.exe` bypass uses the `[HKEY_CURRENT_USER]\Software\Microsoft\Windows\CurrentVersion\App Paths\control.exe` and `[HKEY_CURRENT_USER]\Software\Classes\exefile\shell\runas\command\isolatedCommand` Registry keys. [30] [31]

Analysts should monitor these Registry settings for unauthorized changes.

Other Repositories

More sources than just MITRE ATT&CK for detection rules

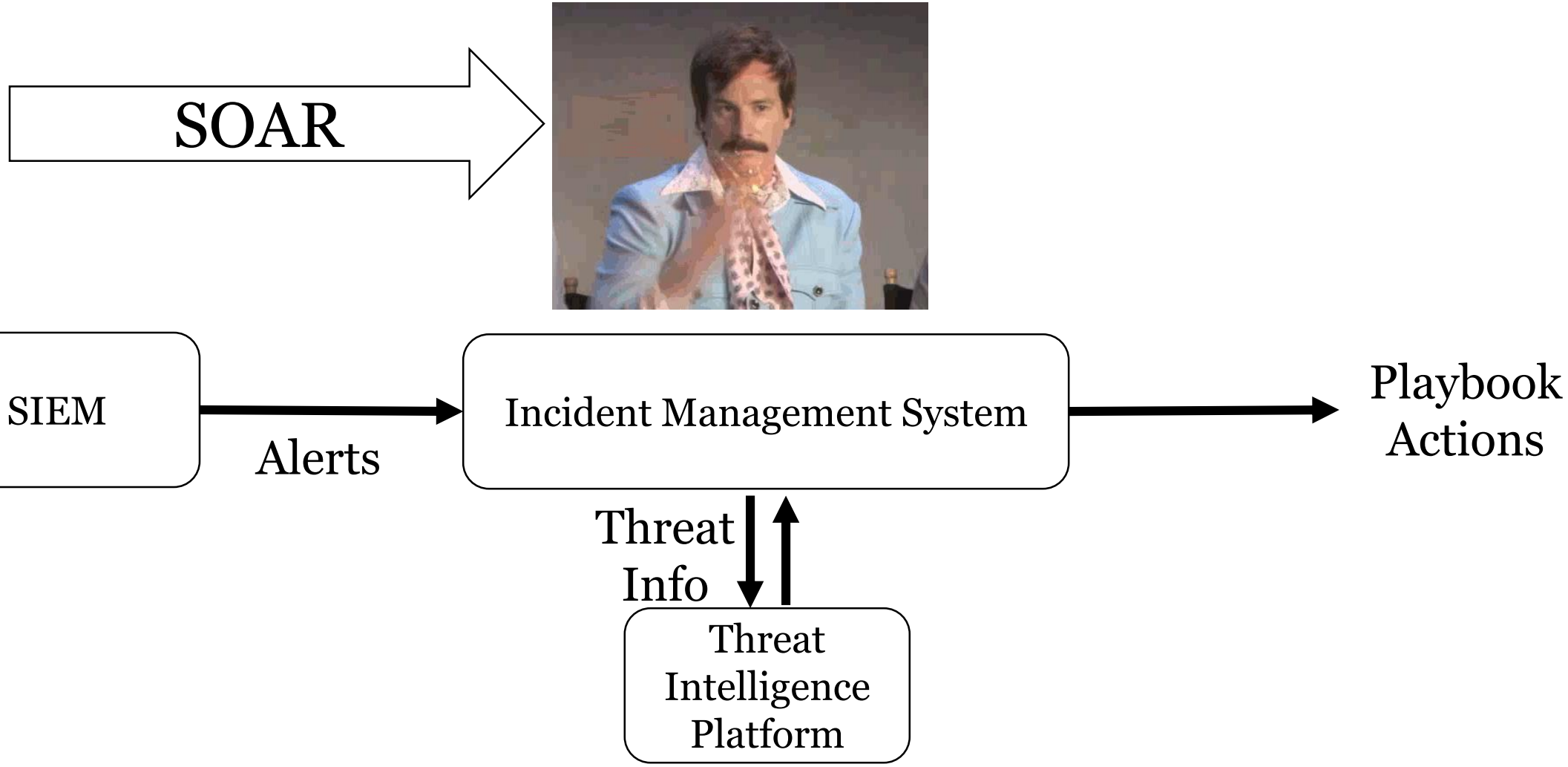
- PDF - **NSA Spotting the Adversary**
- **Sigma** - Open source rule repository (more on this later)
- SOC Prime - **Threat Detection Marketplace**
  - Open source rules
  - And commercial rules

Threat Feeds - MISP, Open Threat Exchange, etc.

O is for…

Orchestration

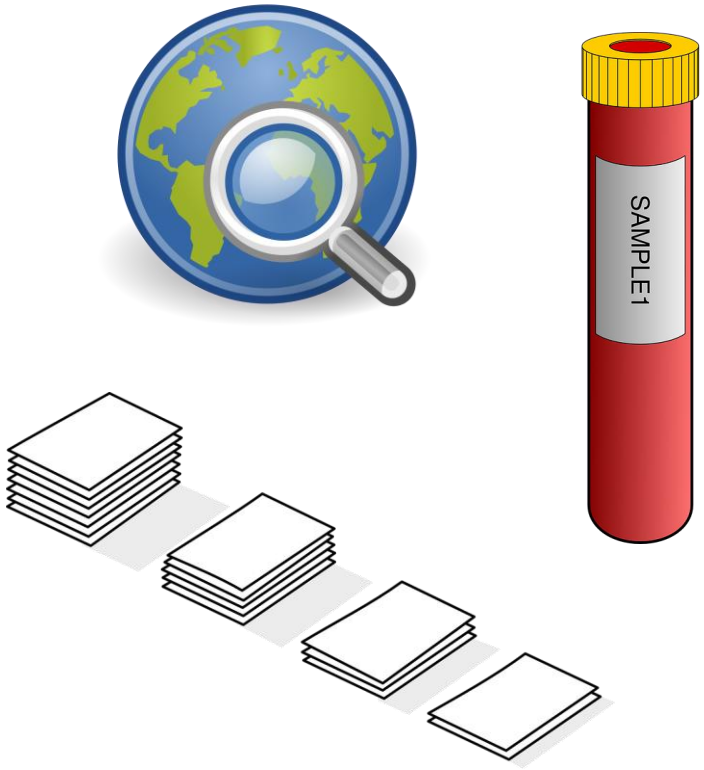
SOAR + SIEM + SOC = Magic



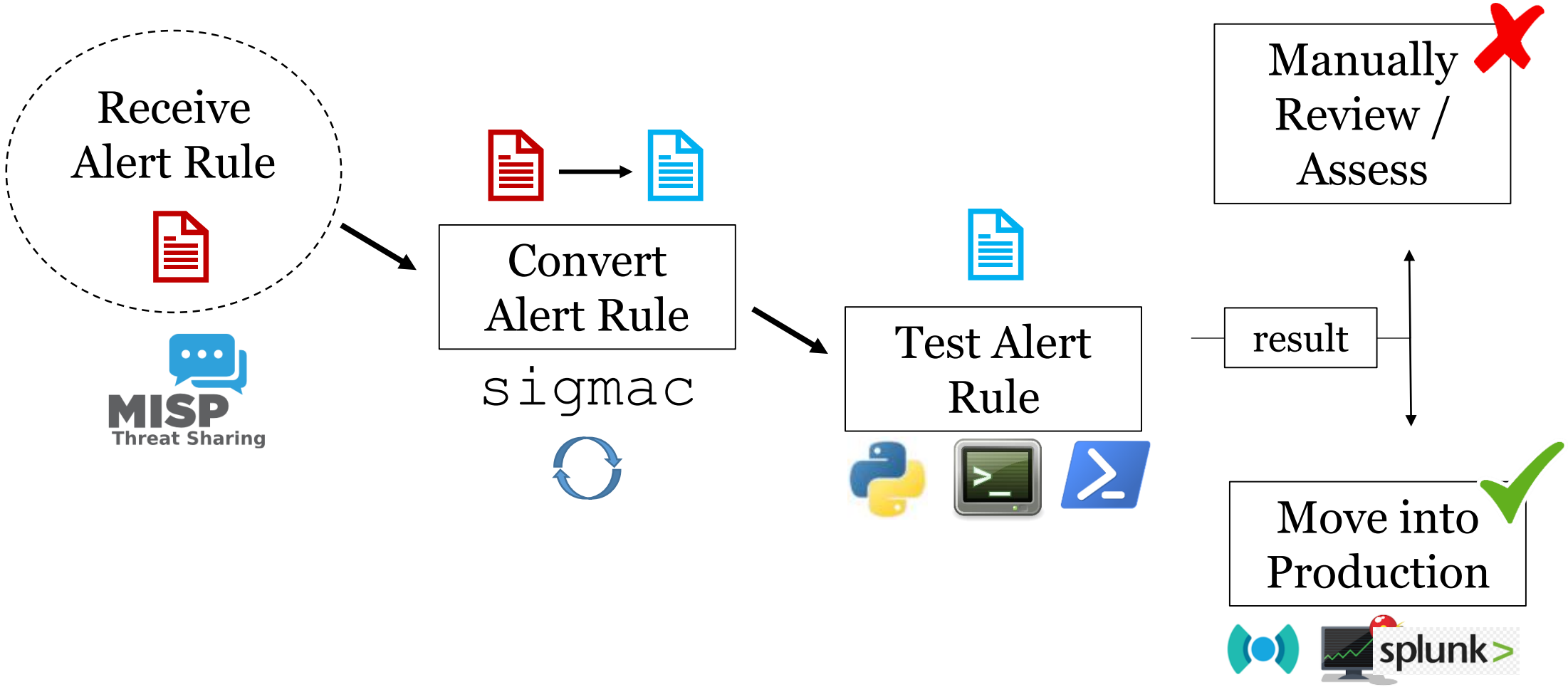
Automation and Orchestration Use Case Categories

Typical categories for SOAR:

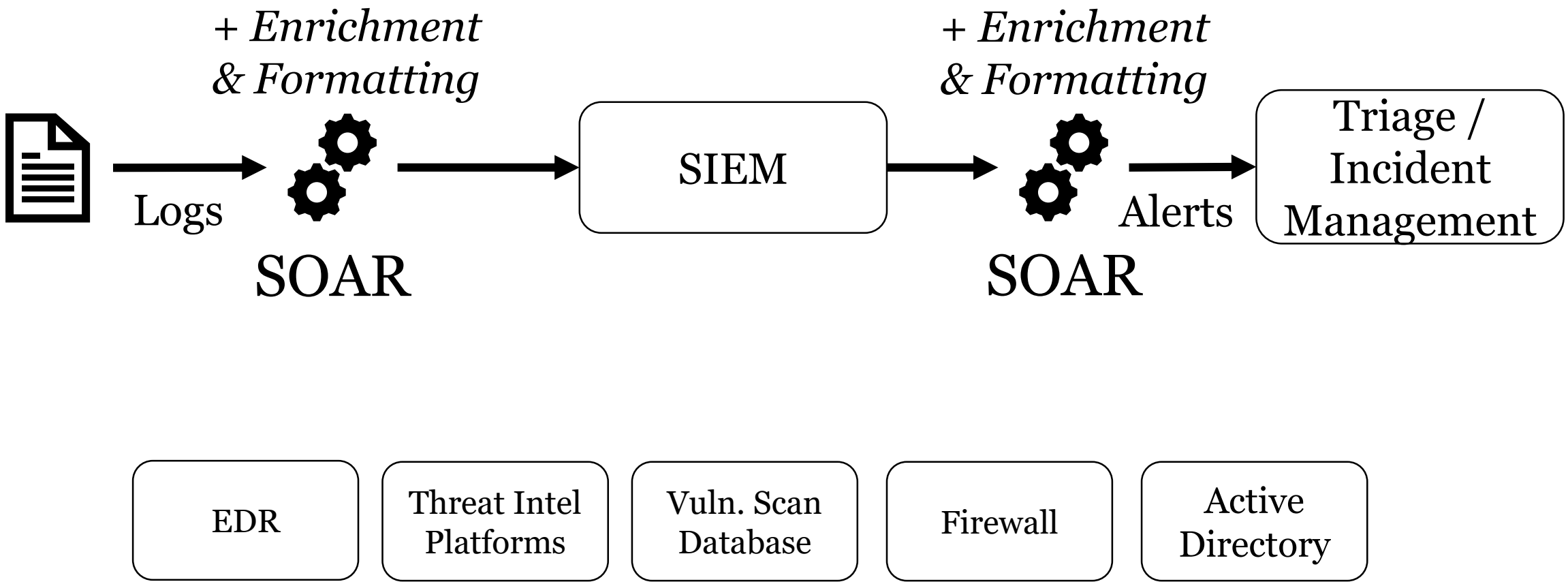
- **Enumeration** and **Enrichment** (IP, Hostname, Hash)
  - Using internal tool APIs
  - On external data
  - Resolved by SOAR framework
- **Incident Response**
  - Blocking actions
  - Sample gathering
  - Cleanup
- **Alert and Case management**



Orchestration



Give Your SIEM a Hand with SOAR



U is for…

Universalize

Universalize

Logs vary from data source to data source and org to org

- Need to universalize for applied logic and analytics

Oct 5 15:06:54 server sshd[2014]: Failed password for invalid user 123 from 212.129.35.106 port 43271 ssh2

Compare ↕

An account failed to log on.

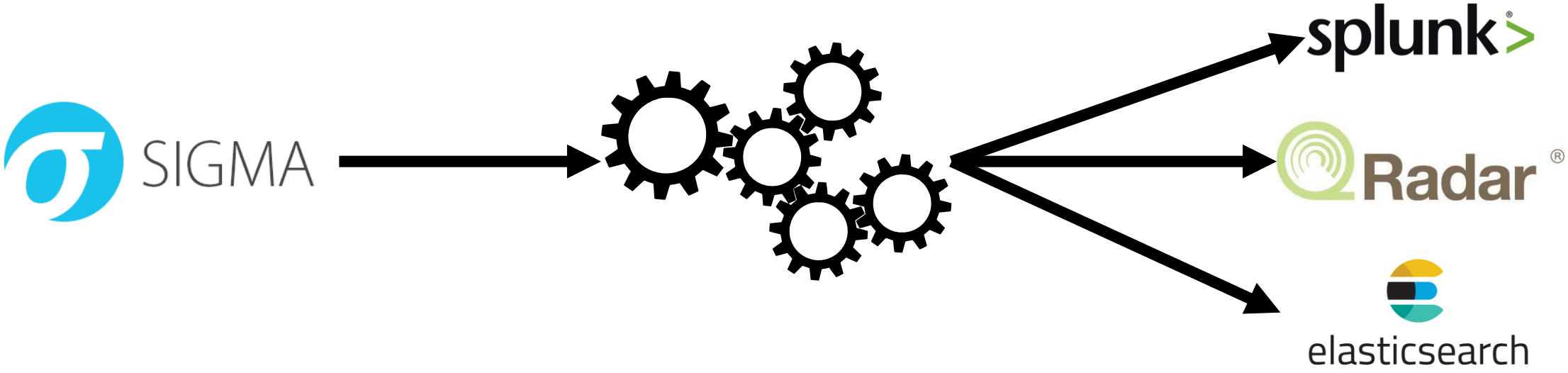
Subject:
Security ID: S-1-5-21-2635542286-2942777934-2742232658-1105
Account Name: jhenderson
Account Domain: SEC555

How to universalize?

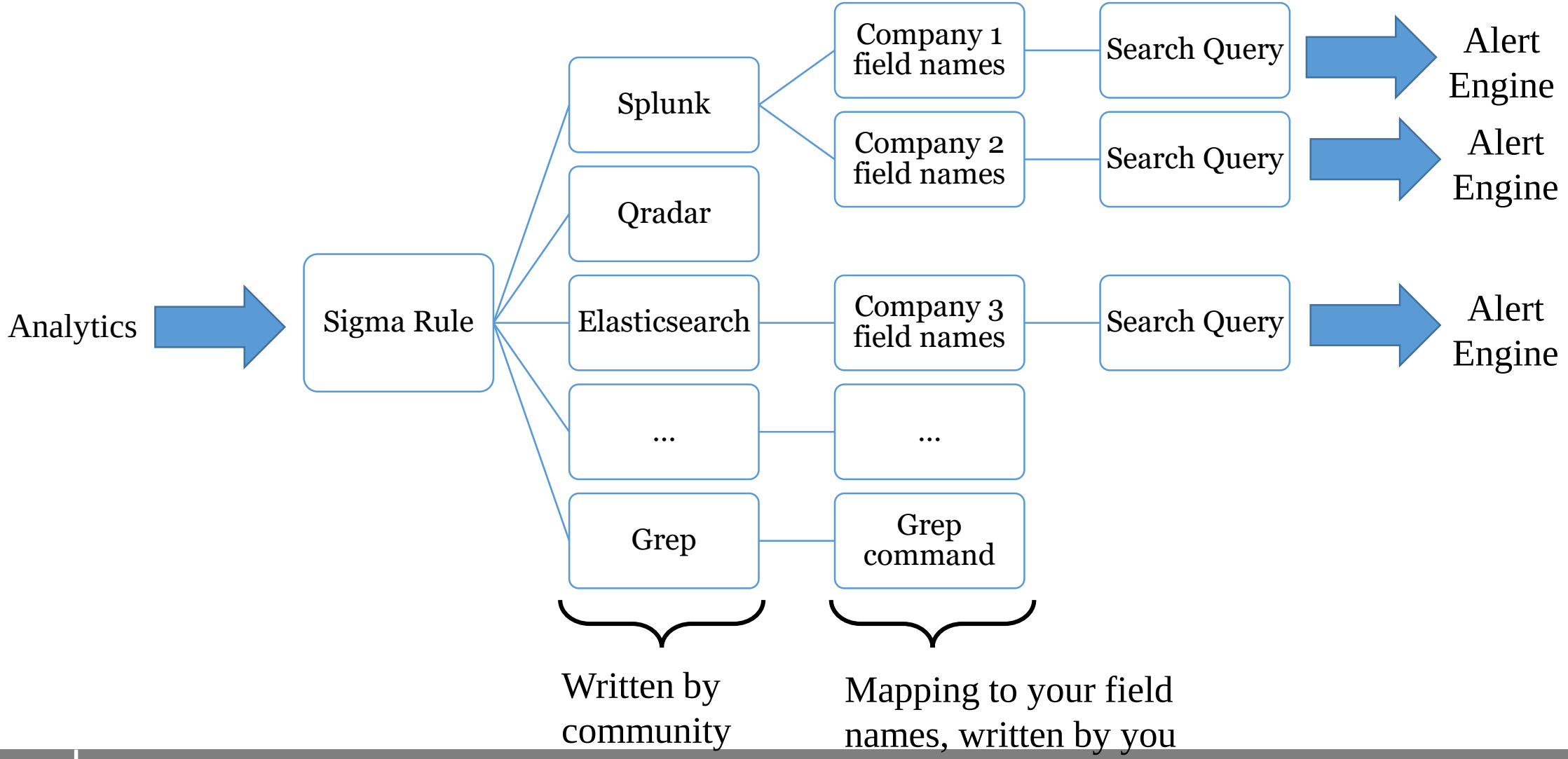
- Parse with **standard field names**
- Add **field aliases** to standard field names
- Utilize **tags** such as logon_failure

Sigma

- Written by **Florian Roth** & **Thomas Patzke**
  - **"To logs, what Snort is to network traffic and YARA is to files"**
- High level **generic language for analytics**
- **Enables analytics re-use and sharing** across orgs



Conversion of Signatures to Alert Queries



One Input – Three Outputs

```
$ ./sigmac --target splunk --config ./config/splunk-windows-index.yml win_pass_the_hash.yml
```

```
(index="windows" (LogonType="3" LogonProcessName="NtLmSsp" WorkstationName="%Workstations%"
ComputerName="%Workstations%" (EventCode="4624" OR EventCode="4625")) NOT
(AccountName="ANONYMOUS LOGON"))
```

```
$ ./sigmac --target qradar --config ./config/qradar.yml win_pass_the_hash.yml
```

```
SELECT UTF8(payload) as search_payload from events where
(LOGSOURCETYPENAME(devicetype)='Microsoft Windows Security Event Log' and (LogonType='3'
and LogonProcessName='NtLmSsp' and WorkstationName='%Workstations%' and
ComputerName='%Workstations%' and ("Event ID Code"='4624' or "Event ID Code"='4625')) and
not (AccountName='ANONYMOUS LOGON'))
```

```
$ ./sigmac --target es-qs --config ./config/winlogbeat.yml win_pass_the_hash.yml
```

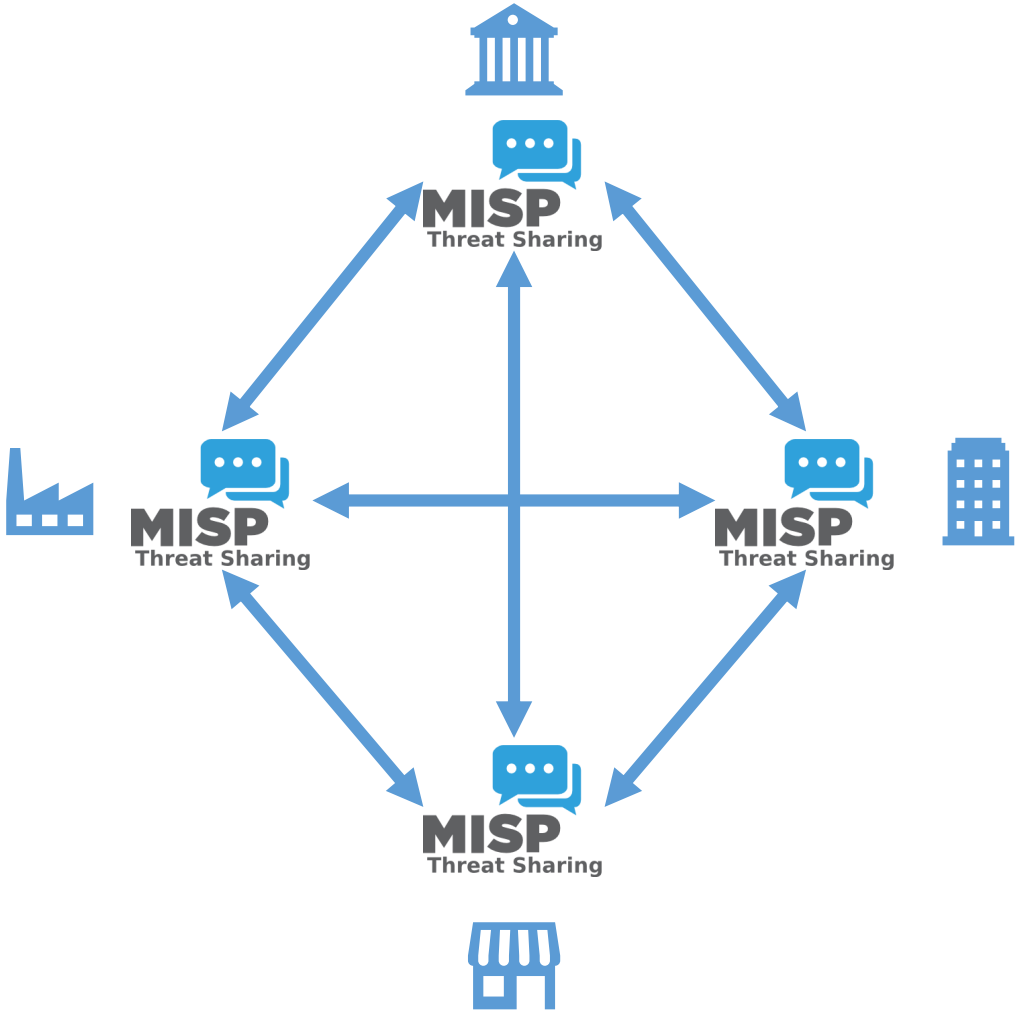
```
(winlog.channel:"Security" AND (winlog.event_data.LogonType:"3" AND
winlog.event_data.LogonProcessName:"NtLmSsp" AND
winlog.event_data.WorkstationName:"%Workstations%" AND winlog.ComputerName:"%Workstations%"
AND (winlog.event_id:"4624" OR winlog.event_id:"4625")) AND (NOT
(winlog.event_data.AccountName:"ANONYMOUS\ LOGON")))
```

Sigma + MISP

- **MISP is one of the best, free Threat Intel Platforms**
- Wide usage in enterprise
  - Integrates well with other tools via open API
- “Event” driven data organization
  - All hashes, IPs, URLs, for incident go into an "event"
- **Meant for sharing**
  - **Supports Sigma rules** as object type
- Tool **sigma2misp** pushes rules to events

Imagine a world…

- Where intelligence **reports come with Sigma rules**
- **Don't have to write the analytics**
- **Don't even have to transcribe them**
  - They came to you through MISP!
- **Analytics automatically appear in Threat Intel Platform**
  - Already associated with threat actors
  - Supporting IOCs included
- Simply convert the rules you want!



Conclusion

Is your SIEM doing everything for you that it can?

Remember…

- ✓ **A - Automation**
- ✓ **E - Enrichment**
- ✓ **I - Identify**
- ✓ **O - Orchestration**
- ✓ **U - Universalize**

Which are all enabled by **Y** ---> **You**

Thank you!