Fast Forensics and Threat Hunting with Yamato Security Tools

大和セキュリティ

SANS DFIR Summit 2023

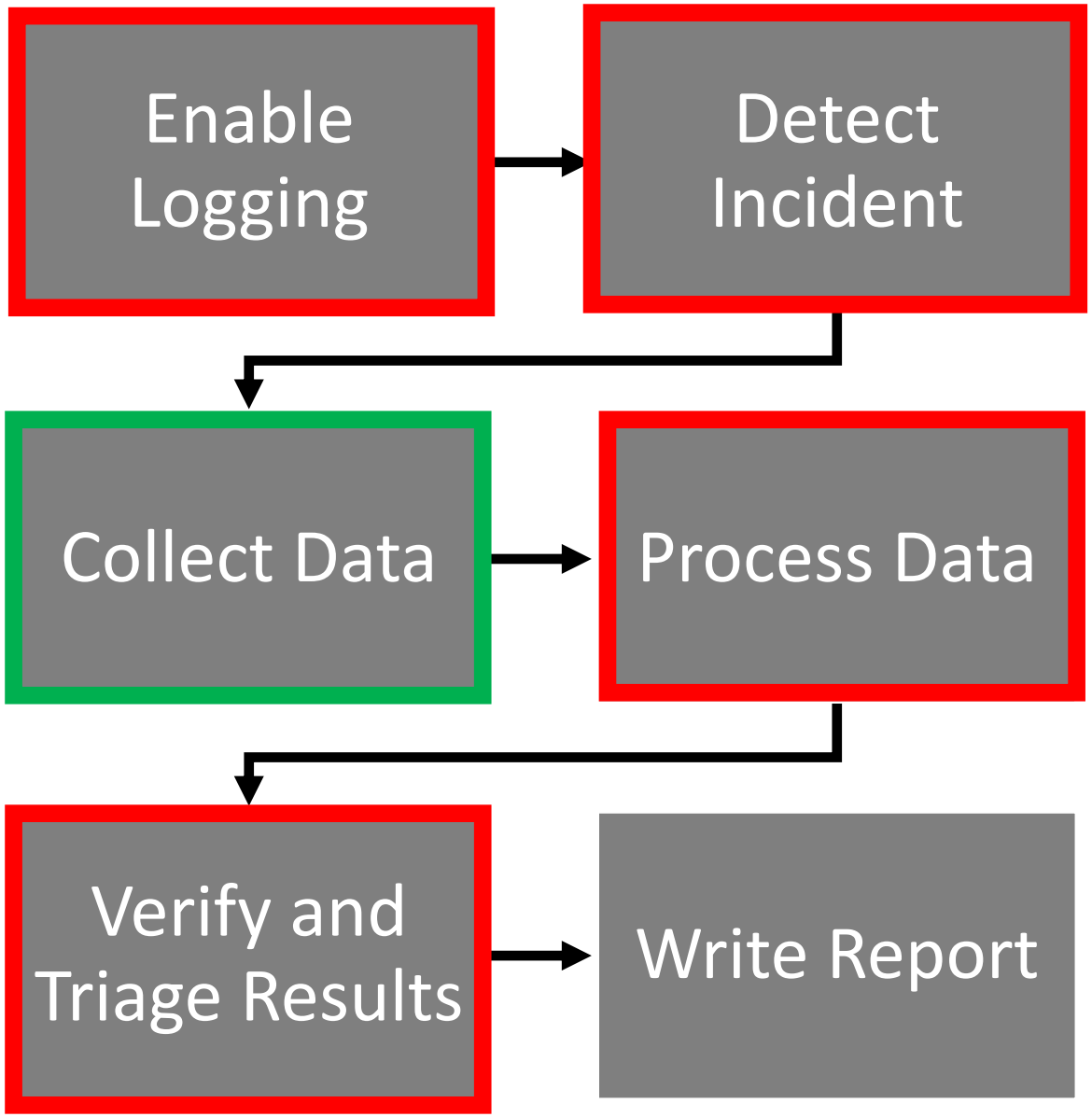
Zach Mathis (@yamatosecurity)

whoami

自己紹介


NORTH
KOREA
SEA
OF
JAPAN
JAPAN
SOUTH
KOREA
Tokyo
Kobe

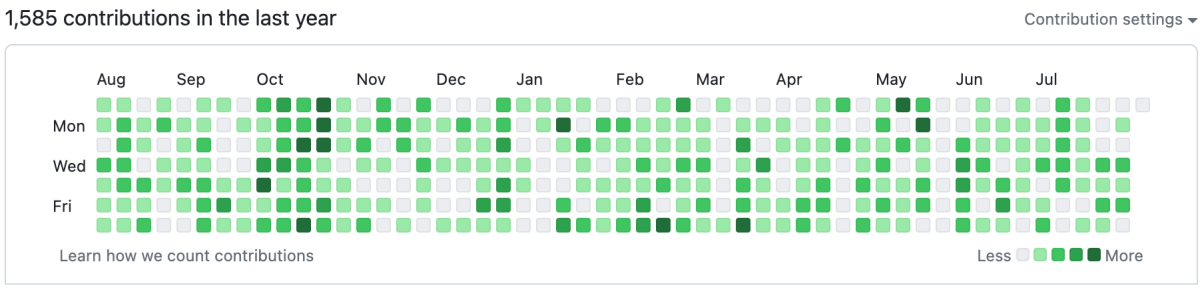
Automating "Fast Forensics"



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.
- We have been releasing free OSS DFIR tools since 2021.

Yamato Security Tools



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

HAYABUSA

"Windows Event Log Refurbishing Tool"

HAYABUSA

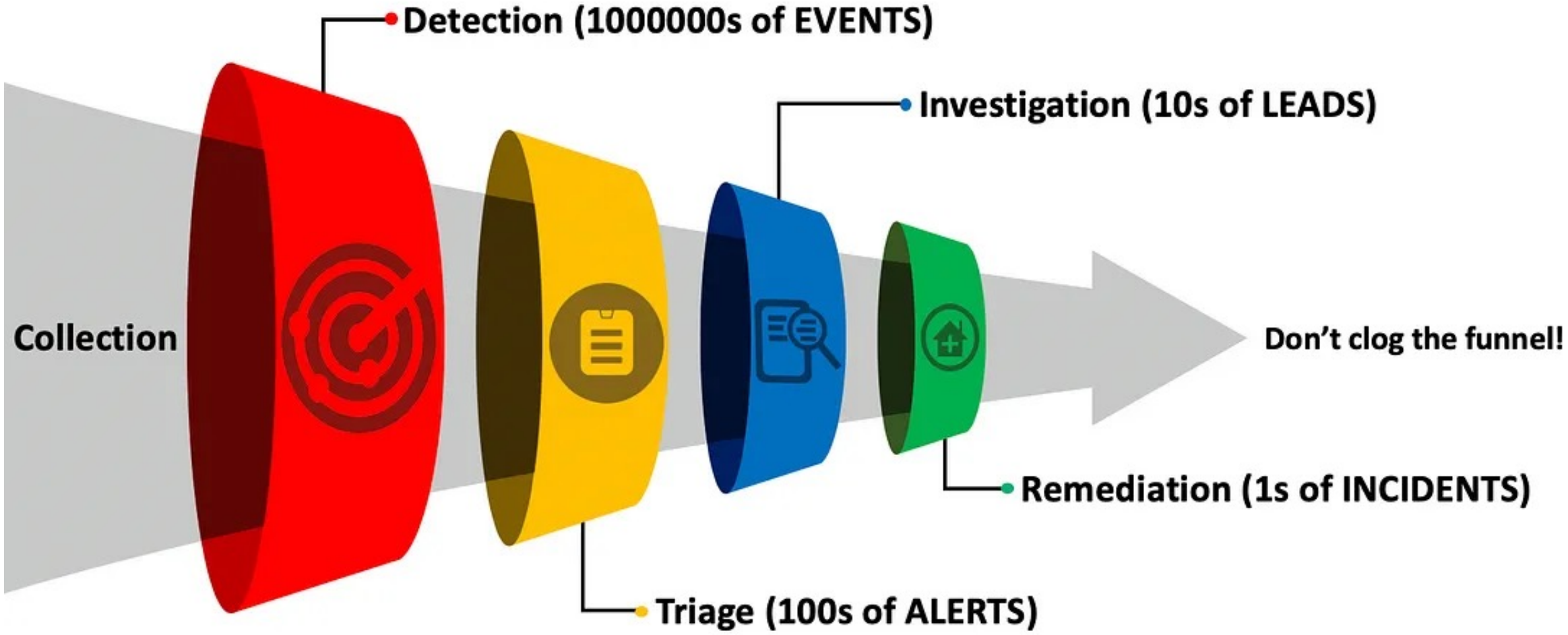
"Fast forensics timeline generator and threat hunting tool"

Threat Hunting with Velociraptor

+

(Thanks to Whitney Champion and Eric Capuano!)

Automate the "Funnel of Fidelity"



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

Hayabusa Stats

- Project started: Late 2020
- Initial release: Christmas 2021
- Contributors: 14
- Releases: 35+
- Closed issues: 472+
- Pull requests merged: 625+
- GitHub stars: ~15k
- Talks: 8? (Black Hat, CODE BLUE, SECCON, etc...)
- Downloads: ~40k

Hayabusa

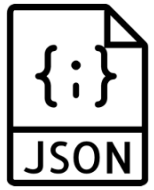
- Written in Rust so fast and memory safe.
- Multi-platform. (Win, Lin, Mac, etc...)
- Standalone binary.
- Rules and config files based on sigma and YML so easy to customize and extend.
- New version 2.7.0 release today!

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



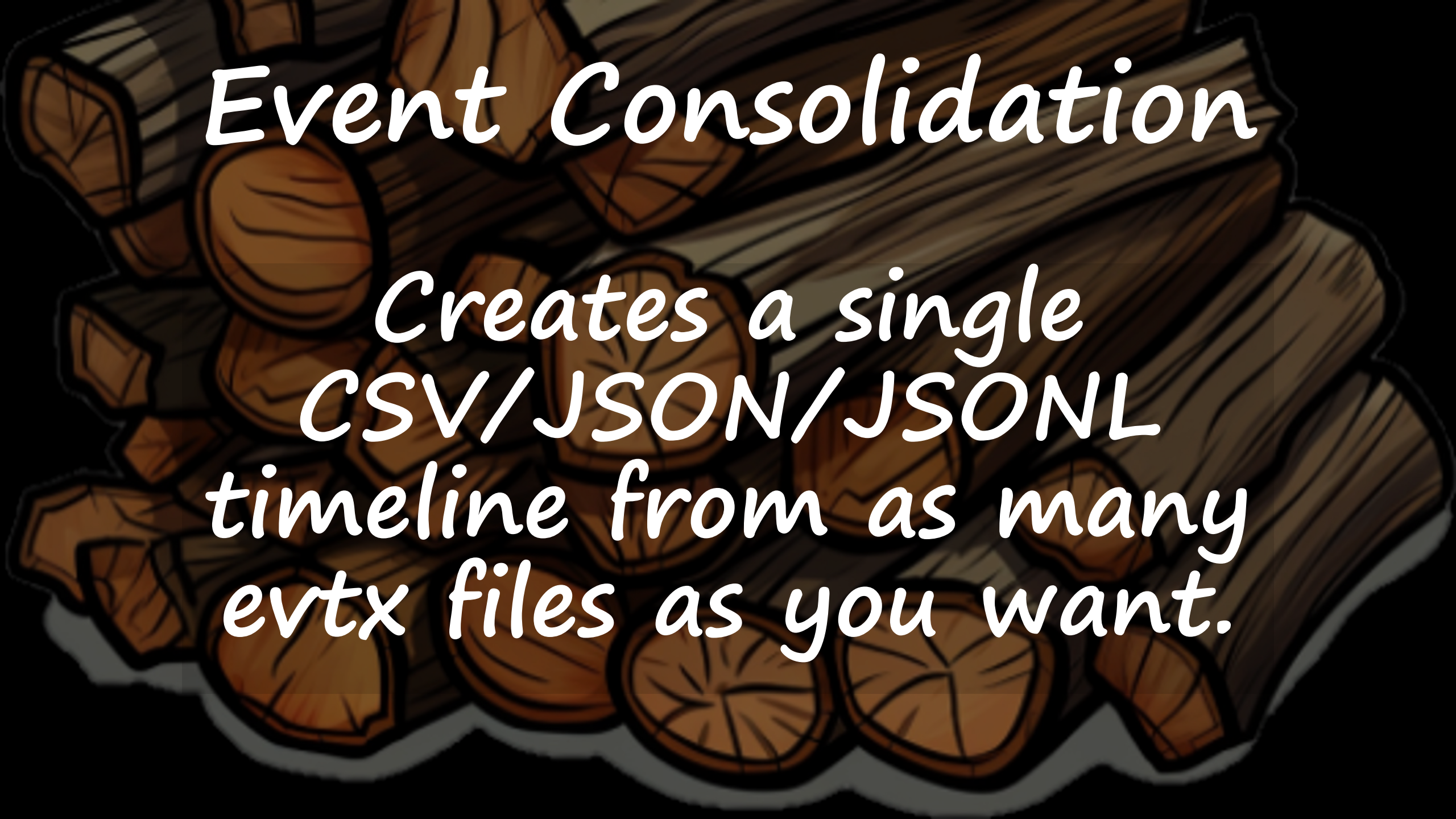
Timeline Explorer,
Excel, xsv, etc...

Elastic Stack,
Timesketch,
takajo, etc...

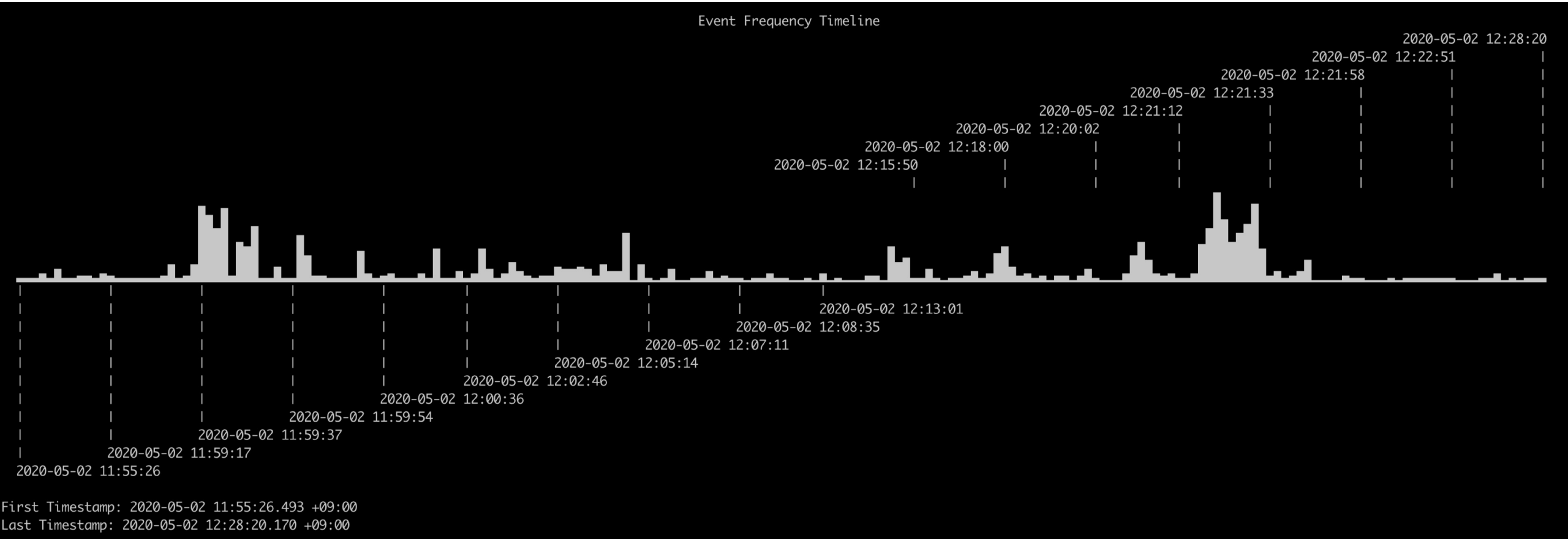
jq, xq, etc...

Sigma Support

- Native sigma support.
- Over 2500 rules!
- Support for deprecated features like “count” to detect password spray/guessing, etc… until sigma correlations is finalized.
- Added new features to detect more attacks.
- Rules updated daily.


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 requested"
- "Special privileges assigned to new logon" ->
  "Admin logon"
- "A logon was attempted using explicit credentials" ->
  "Explicit logon"

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%)

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…

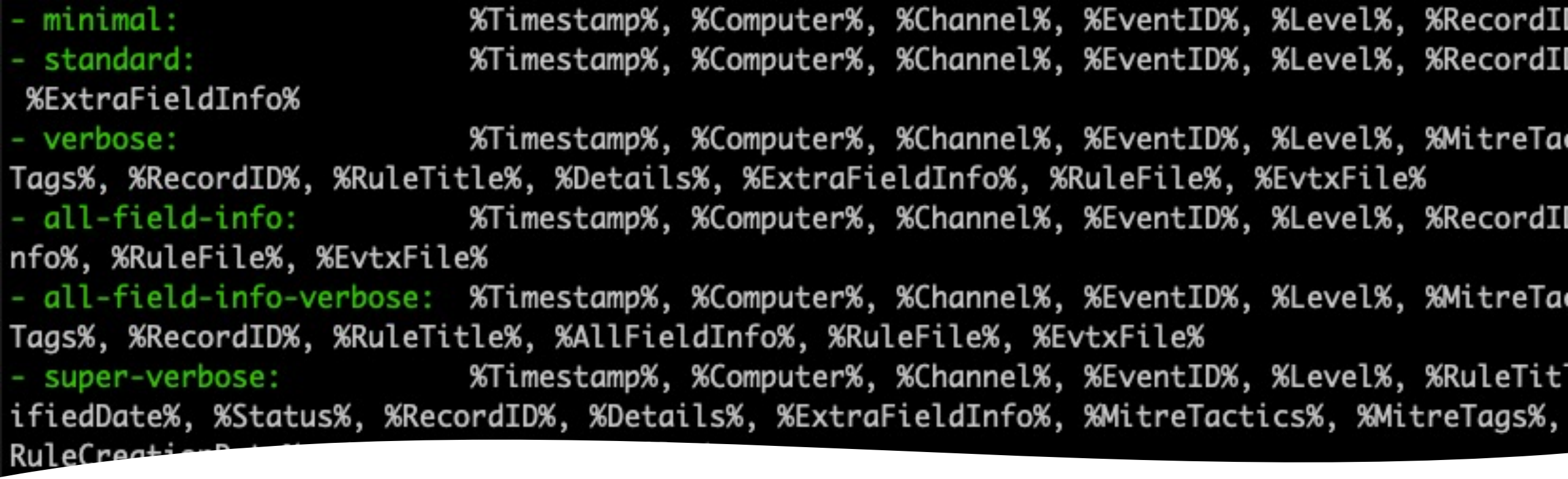
A	C	D	E	G	H	
Timestamp	Chann	Even	Lev	RuleTitle	RuleAuthor	Details
2019-08-05 18:39:	Sec	4624	med	Pass the Hash Activity 2	Dave Kennedy Jeff Warren (method) David Vassallo (rule)	Type: 9 TgtUser: IEUser SrcComp: - SrcIP: ::1 LID: 0x38f87e
2019-08-05 18:39:	Sec	4624	high	Successful Overpass the Hash Attempt	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

Field Data Mapping

```yaml
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'
```

New in 2.7.0 thanks to Fukusuke Takahashi!



Customizable Output

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

"Windows Event Log Refurbishing Tool"

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

Limitations

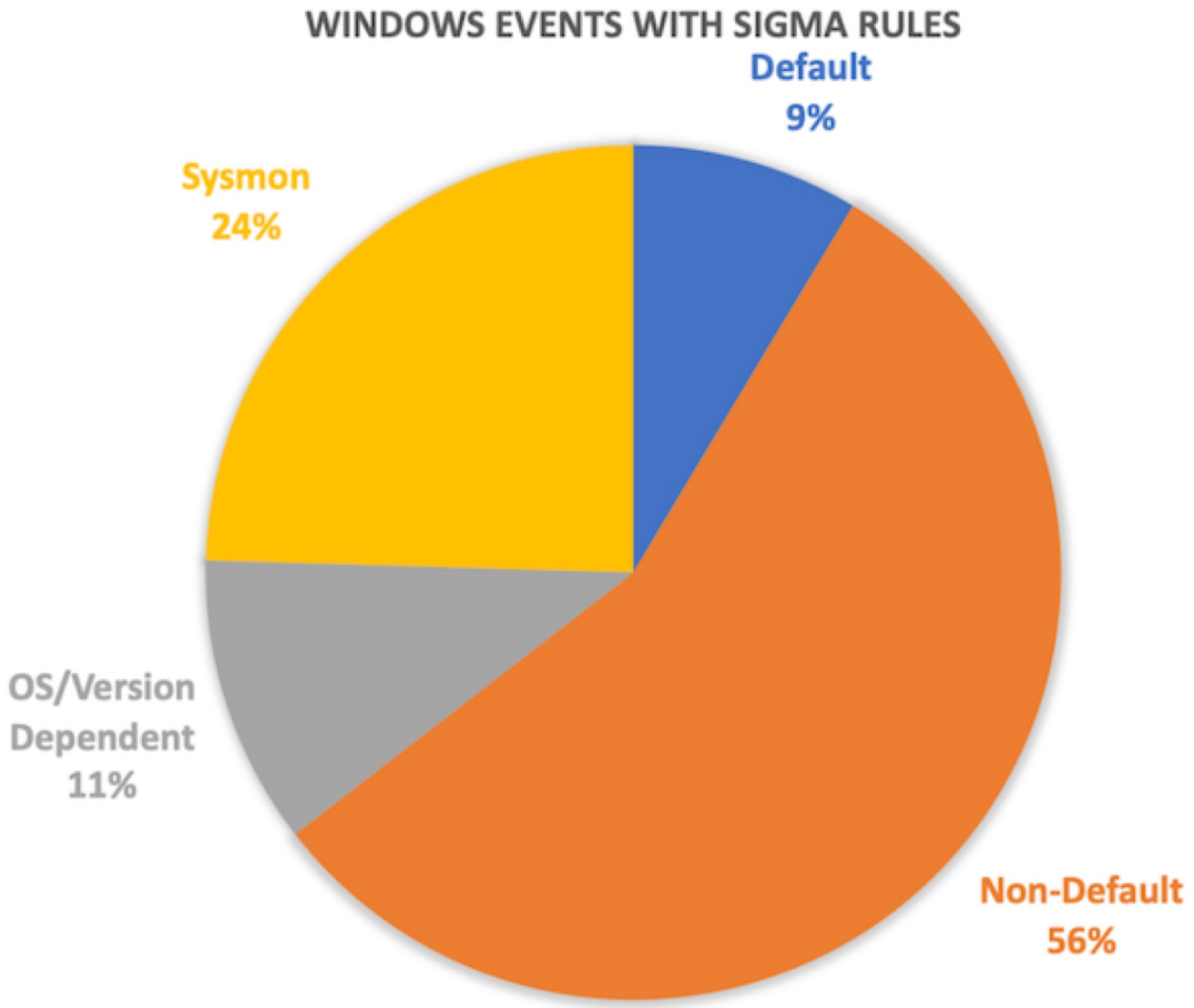
- All tools have limitations so know your tools and what they do in the background.
- Now Hayabusa is meant for fast forensics, triage, etc... not for 100% thorough analysis.
- However, one day we will probably support extracting out all events as well.
- If you want us to add more events in the meantime,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

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

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%

Security Log Events with Sigma Rules

EID	Event	Default Settings	Rules	%
4688	Process Creation	No	695	73.70%
4697	Service installed	Yes - Win 10/2016+	20	2.12%
5145	Access attempt to a network share object	No	18	1.91%
4624	Successful logon	Yes	12	1.27%
4656	Object handle request	No	12	1.27%
4663	Access attempt to object	No	12	1.27%
4625	Failed logon	Server OS only	10	1.06%
4776	NTLM Authentication	Success Only - Servers only	6	0.64%
4662	Operation performed on object	Server OS only	6	0.64%
5136	Directory service object was modified	No	6	0.64%
4657	A registry value was modified	No	5	0.53%
5156	Windows Filtering Platform has allowed a connection	No	4	0.42%
4720	User account created	Success only	4	0.42%
4738	User account changed	Success Only	4	0.42%

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.

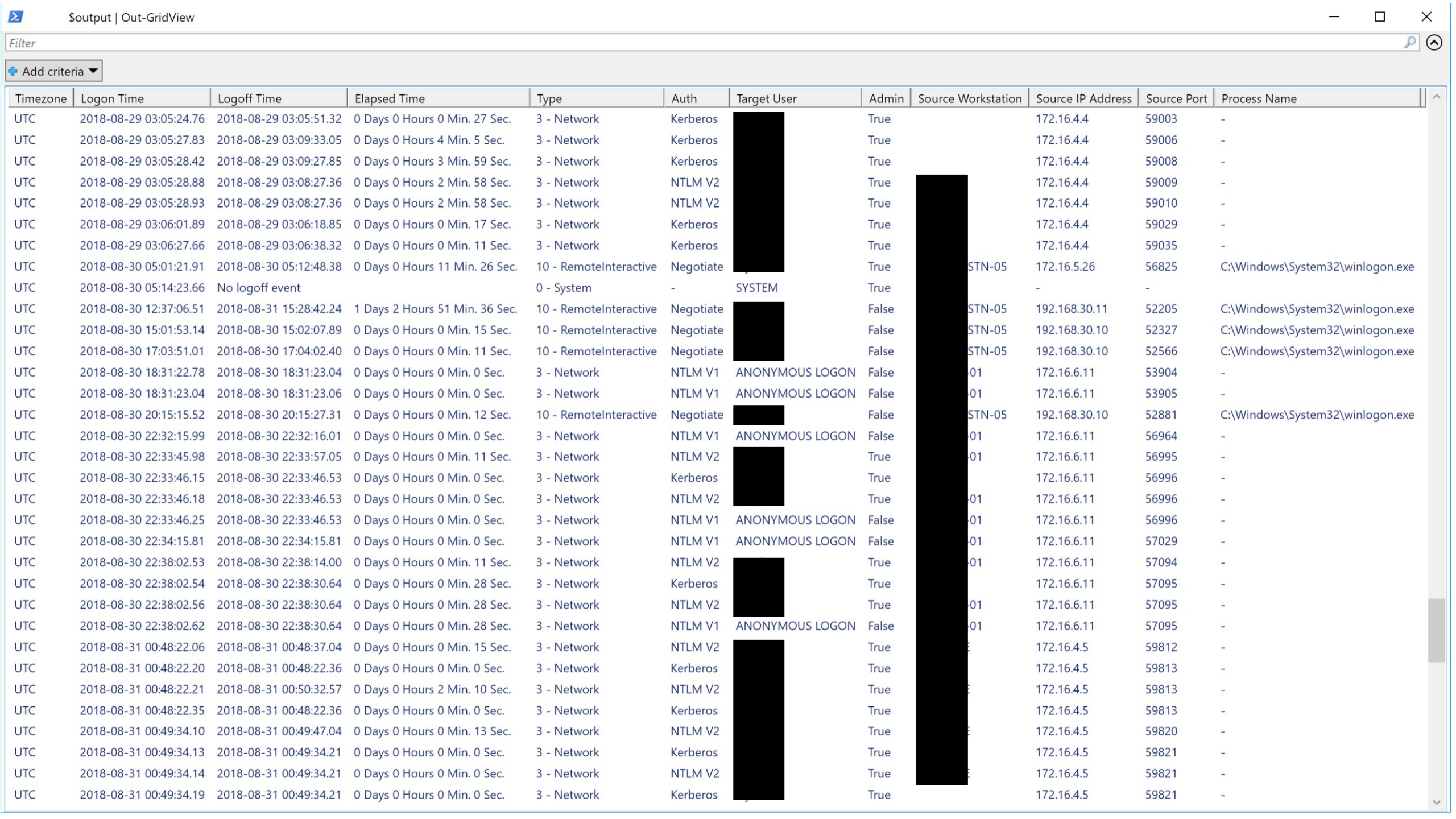
Solution

- YamatoSecurityConfigureWinEventLogs.bat
- or use your favorite baseline from CIS, ACSC, NSA, MS, etc… but use our documentation as a reference.
- but make sure you understand what data is needed for your detection, you have enabled it and tested that your logs are not too noisy…
- (easier said than done…)

WELA

WELA

- "Windows Event Log Analyzer"
- Language: Windows PowerShell
- Does Event ID metrics, NTLM usage analysis, etc... but the main feature was the fast forensics logon timeline generator
- Consolidates only the useful information in multiple logon log entries (4624, 4634, 4647, 4672, 4776) into single events, perform data reduction by ignoring around 90% of the noise, and will convert any hard to read data (such as hex status codes) into human readable format.



Timezone	Logon Time	Logoff Time	Elapsed Time	Type	Auth	Target User	Admin	Source Workstation	Source IP Address	Source Port	Process Name
UTC	2018-08-29 03:05:24.76	2018-08-29 03:05:51.32	0 Days 0 Hours 0 Min. 27 Sec.	3 - Network	Kerberos	[redacted]	True		172.16.4.4	59003	-
UTC	2018-08-29 03:05:27.83	2018-08-29 03:09:33.05	0 Days 0 Hours 4 Min. 5 Sec.	3 - Network	Kerberos		True		172.16.4.4	59006	-
UTC	2018-08-29 03:05:28.42	2018-08-29 03:09:27.85	0 Days 0 Hours 3 Min. 59 Sec.	3 - Network	Kerberos		True		172.16.4.4	59008	-
UTC	2018-08-29 03:05:28.88	2018-08-29 03:08:27.36	0 Days 0 Hours 2 Min. 58 Sec.	3 - Network	NTLM V2		True	[redacted]	172.16.4.4	59009	-
UTC	2018-08-29 03:05:28.93	2018-08-29 03:08:27.36	0 Days 0 Hours 2 Min. 58 Sec.	3 - Network	NTLM V2		True		172.16.4.4	59010	-
UTC	2018-08-29 03:06:01.89	2018-08-29 03:06:18.85	0 Days 0 Hours 0 Min. 17 Sec.	3 - Network	Kerberos		True		172.16.4.4	59029	-
UTC	2018-08-29 03:06:27.66	2018-08-29 03:06:38.32	0 Days 0 Hours 0 Min. 11 Sec.	3 - Network	Kerberos		True		172.16.4.4	59035	-
UTC	2018-08-30 05:01:21.91	2018-08-30 05:12:48.38	0 Days 0 Hours 11 Min. 26 Sec.	10 - RemoteInteractive	Negotiate		True	STN-05	172.16.5.26	56825	C:\Windows\System32\winlogon.exe
UTC	2018-08-30 05:14:23.66	No logoff event		0 - System	-	SYSTEM	True		-	-	
UTC	2018-08-30 12:37:06.51	2018-08-31 15:28:42.24	1 Days 2 Hours 51 Min. 36 Sec.	10 - RemoteInteractive	Negotiate		False	STN-05	192.168.30.11	52205	C:\Windows\System32\winlogon.exe
UTC	2018-08-30 15:01:53.14	2018-08-30 15:02:07.89	0 Days 0 Hours 0 Min. 15 Sec.	10 - RemoteInteractive	Negotiate		False	STN-05	192.168.30.10	52327	C:\Windows\System32\winlogon.exe
UTC	2018-08-30 17:03:51.01	2018-08-30 17:04:02.40	0 Days 0 Hours 0 Min. 11 Sec.	10 - RemoteInteractive	Negotiate		False	STN-05	192.168.30.10	52566	C:\Windows\System32\winlogon.exe
UTC	2018-08-30 18:31:22.78	2018-08-30 18:31:23.04	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	NTLM V1	ANONYMOUS LOGON	False	01	172.16.6.11	53904	-
UTC	2018-08-30 18:31:23.04	2018-08-30 18:31:23.06	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	NTLM V1	ANONYMOUS LOGON	False	01	172.16.6.11	53905	-
UTC	2018-08-30 20:15:15.52	2018-08-30 20:15:27.31	0 Days 0 Hours 0 Min. 12 Sec.	10 - RemoteInteractive	Negotiate		False	STN-05	192.168.30.10	52881	C:\Windows\System32\winlogon.exe
UTC	2018-08-30 22:32:15.99	2018-08-30 22:32:16.01	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	NTLM V1	ANONYMOUS LOGON	False	01	172.16.6.11	56964	-
UTC	2018-08-30 22:33:45.98	2018-08-30 22:33:57.05	0 Days 0 Hours 0 Min. 11 Sec.	3 - Network	NTLM V2		True	01	172.16.6.11	56995	-
UTC	2018-08-30 22:33:46.15	2018-08-30 22:33:46.53	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	Kerberos		True		172.16.6.11	56996	-
UTC	2018-08-30 22:33:46.18	2018-08-30 22:33:46.53	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	NTLM V2		True	01	172.16.6.11	56996	-
UTC	2018-08-30 22:33:46.25	2018-08-30 22:33:46.53	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	NTLM V1	ANONYMOUS LOGON	False	01	172.16.6.11	56996	-
UTC	2018-08-30 22:34:15.81	2018-08-30 22:34:15.81	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	NTLM V1	ANONYMOUS LOGON	False	01	172.16.6.11	57029	-
UTC	2018-08-30 22:38:02.53	2018-08-30 22:38:14.00	0 Days 0 Hours 0 Min. 11 Sec.	3 - Network	NTLM V2		True	01	172.16.6.11	57094	-
UTC	2018-08-30 22:38:02.54	2018-08-30 22:38:30.64	0 Days 0 Hours 0 Min. 28 Sec.	3 - Network	Kerberos		True		172.16.6.11	57095	-
UTC	2018-08-30 22:38:02.56	2018-08-30 22:38:30.64	0 Days 0 Hours 0 Min. 28 Sec.	3 - Network	NTLM V2		True	01	172.16.6.11	57095	-
UTC	2018-08-30 22:38:02.62	2018-08-30 22:38:30.64	0 Days 0 Hours 0 Min. 28 Sec.	3 - Network	NTLM V1	ANONYMOUS LOGON	False	01	172.16.6.11	57095	-
UTC	2018-08-31 00:48:22.06	2018-08-31 00:48:37.04	0 Days 0 Hours 0 Min. 15 Sec.	3 - Network	NTLM V2		True	E	172.16.4.5	59812	-
UTC	2018-08-31 00:48:22.20	2018-08-31 00:48:22.36	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	Kerberos		True		172.16.4.5	59813	-
UTC	2018-08-31 00:48:22.21	2018-08-31 00:50:32.57	0 Days 0 Hours 2 Min. 10 Sec.	3 - Network	NTLM V2		True	E	172.16.4.5	59813	-
UTC	2018-08-31 00:48:22.35	2018-08-31 00:48:22.36	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	Kerberos		True		172.16.4.5	59813	-
UTC	2018-08-31 00:49:34.10	2018-08-31 00:49:47.04	0 Days 0 Hours 0 Min. 13 Sec.	3 - Network	NTLM V2		True	E	172.16.4.5	59820	-
UTC	2018-08-31 00:49:34.13	2018-08-31 00:49:34.21	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	Kerberos		True		172.16.4.5	59821	-
UTC	2018-08-31 00:49:34.14	2018-08-31 00:49:34.21	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	NTLM V2		True	E	172.16.4.5	59821	-
UTC	2018-08-31 00:49:34.19	2018-08-31 00:49:34.21	0 Days 0 Hours 0 Min. 0 Sec.	3 - Network	Kerberos		True		172.16.4.5	59821	-

WELA is now deprecated as of today in favor of Hayabusa and Takajo!

鷹匠

Takajō

Takajo

- Takajo: “Falconer”
- Language: Nim
- Memory safe, fast, multi-platform
- Extremely easy to code!
- Version 2.0.0 released today!

New Takajo Commands

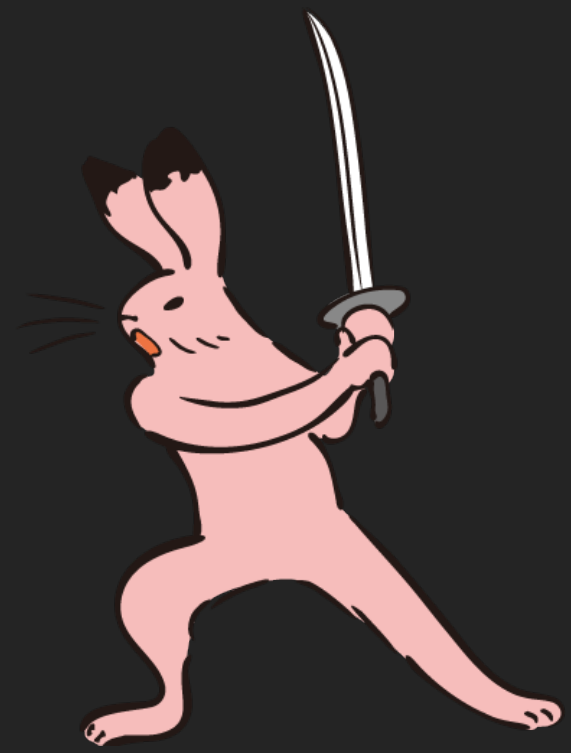
- List commands:
  - `list-domains`: extract out all domain names
  - `list-hashes`: list up process hashes
  - `list-ip-addresses`: extract out all IP addresses
- Split commands:
  - `split-csv-timeline` & `split-json-timeline`: split up large timelines based on computer name
- Stack commands:
  - `stack-logons`: frequency analysis of logons

- Sysmon commands:
  - `sysmon-process-tree`: create a process tree visualization of malicious processes
- Timeline commands:
  - `timeline-logon`: WELA style timeline of logons
  - `timeline-suspicious-processes`: timeline of just suspicious processes
- VirusTotal Commands:
  - `vt-domain-lookup`, `vt-hash-lookup`, `vt-ip-lookup`: lookup domains, IP addresses and hashes on VirusTotal

Hayabusa + Takajo

v.s.

SANS 608 CTF



The Challenge

- Team of 4-5 people.
- 92 GB of evtx data!
- Only 4 hours to process and analyze the data.

Objectives

- Find "patient zero"
- Find out how the breach happened
- Find lateral movement, exfil, etc…
- List up malware, IP addresses, persistence, compromised hosts and users, etc…

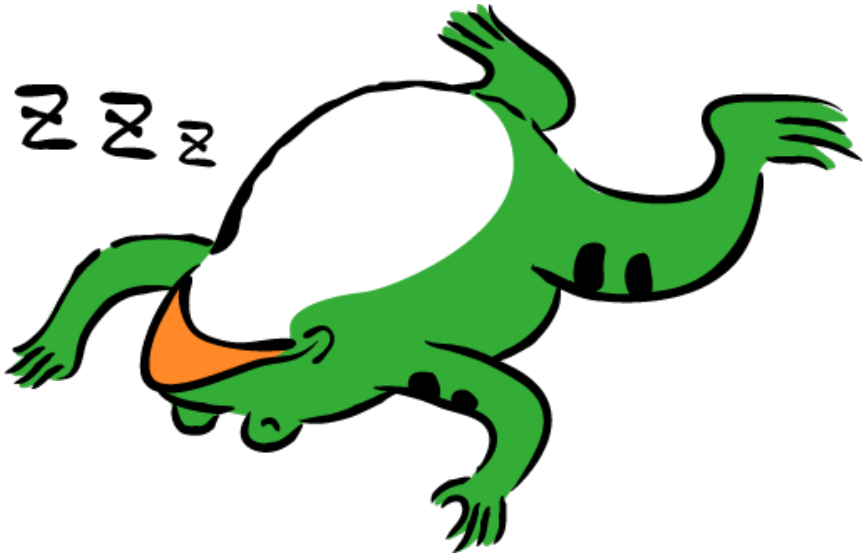
My Hardware

- Lenovo P51 (Windows 11, 2017 Intel Xeon 4 Core @3GHz, 64 GB RAM, SSDs)
- 1.2x faster with 2019 Intel Mac
- 1.9x faster with M2 Mac


First task: Process the data

Attempt #1: Normal Scan

- `--UTC --enable-unsupported-rules --enable-deprecated-rules --remove-duplicate-data`
- 88 million events -> 18 million events (-80%)
- 11 GB CSV
- Takes over 3 hours!



Problem: Too Many Rules!

- 3917 rules used during the tests (※Actually 2577 different sigma rules but process creation rules, etc... are actually 2 rules in 1...)
- The key to fast scanning is to only enable what you need.
- However...

Too Few Rules Is Also Not Ideal

- ※Hayabusa is not a SIEM~
- Scanning time for just 1 rule:
  26 minutes!
- 26 x 3917 = over 70 days!


Solution: Divide and Conquer!

- With large data, break up scans into 2-3 scans.
- Have different analysts review the different results.

New Filtering In 2.7.0!

- Thanks to DustInDark, you can now filter in/out:
- EIDs, Computer Names, Tags, Categories, Status, Levels, Reliability

Computer Filtering

- First find out which computers have the most events. (New in 2.7.0)
- `hayabusa.exe computer-metrics -d D:\logs\608`
- 20 minutes to process...
- Mail server: 27 million events
- DC: 12 million events
- Everything else: Less than 10 million

Computer Filtering

- **One analyst handles the mail server:**
  `--include-computer base-mail.shieldbase.lan`
  Takes 1hr 8 mins. ~3 hours left for analysis.
- **One analyst handles the DC:**
  `--include-computer base-dc.shieldbase.lan`
  Takes 45 mins. +3 hours left for analysis.
- **One analyst handles everything else (at first):**
  `--exclude-computer base-mail.shieldbase.lan, base-dc.shieldbase.lan`
  Takes 1 hour 52 mins... only 2hr left for analysis...

Another Method?

- First only load high+ or critical rules to identify computers with a high possibility of compromise.
- `-m high -H report.html` options: 2 hours
- `-m critical -H report.html` options: 39 mins
- Then only include those computers in more thorough scans.
- Afterwards, check the other computers.

Divide By MITRE ATT&CK Tactics

- **Use** `--include-tag` **or** `--exclude-tag` **on the following:**
- `attack.reconnaissance`
- `attack.resource_development`
- `attack.initial_access`
- `attack.execution`
- `attack.persistence`
- `attack.privilege_escalation`
- `attack.defense_evasion`
- `attack.credential_access`
- `attack.discovery`
- `attack.lateral_movement`
- `attack.collection`
- `attack.command_and_control`
- `attack.exfiltration`
- `attack.impact`

Divide By Sysmon/Non-Sysmon

- ※We automatically add the "sysmon" tag to all sysmon rules on our backend as it is not included in the original sigma rules.
- Use `--include-tags sysmon` or `--exclude-tags sysmon` to include/exclude sysmon rules:
- sysmon: 1 hr 37 min
- non-sysmon: 2 hr

EID Filter

- `-E, --EID-filter`
- Only scan EIDs that are known to have detections.
- List updated periodically.
- Speed increase: : 3 hrs -> 2 hrs 30 mins (~20% faster)

Proven Rules

- New in 2.7.0
- `-P, --proven-rules`
- Only load rules that have been proven to work.
- Currently 950 rules out of 2577 (~35%).
- Updated about once a month.
- Speed increase: 3 hrs -> 1 hr 25 mins (+50% faster!)

Filtering Based On Time

- **Use `timeline-start` and `timeline-end` options to only scan events near the incident.**
- `--timeline-start "2020-01-01 00:00:00 +00:00"`
- **Not only will processing be faster but analysis will also be much faster and easier.**
- **Speed increase: : 3 hrs -> 1 hr 12 mins (60% increase!)**

Best Method (So Far) (CSV)

- **Mail server:**

```
hayabusa.exe csv-timeline --include-computer
base-mail.shieldbase.lan --proven-rules --EID-
filter --timeline-start "2020-01-01 00:00:00
+00:00" --UTC --enable-unsupported-rules --
visualize-timeline --remove-duplicate-data --
GeoIP <MAXMIND-DB-DIR> -o mail.csv --HTML-
report mail-results.html -d <LOG-DIR>
```

Speed increase: 1 hr 8 mins -> 18 mins! (565 MB CSV) (+115% faster)

Best Method (So Far) (CSV)

- DC:

```
hayabusa.exe csv-timeline --include-computer
base-dc.shieldbase.lan --proven-rules --EID-
filter --timeline-start "2020-01-01 00:00:00
+00:00" --UTC --enable-unsupported-rules --
visualize-timeline --remove-duplicate-data --
GeoIP <MAXMIND-DB-DIR> -o DC.csv --HTML-report
DC-results.html -d <LOG-DIR>
```

Speed increase: 45 mins -> 16 mins! (565 MB CSV)

Best Method (So Far) (CSV)

- **Everything else:**
- hayabusa.exe csv-timeline --UTC **--proven-rules --EID-filter --exclude-computer base-mail.shieldbase.lan,base-dc.shieldbase.lan --timeline-start "2020-01-01 00:00:00 +00:00"** --enable-unsupported-rules --visualize-timeline --remove-duplicate-data --GeoIP <MAXMIND-DB-DIR> -o 608-results.csv --HTML-report 608-results.html -d <LOG-DIR>
  Speed increase: 1 hr 52 mins -> 27 mins! (3.6 GB CSV)
- **Split up non-mail and non-DC CSV files:**
  takajo.exe split-csv-timeline -t ..\hayabusa\608-results.csv
  Takes less than 10 mins!
  155 small CSV files (1 KB ~ 500 MB)
  ※Timeline Explorer is much faster analyzing 500 MB than 5 GB!

Best Method (So Far) (JSONL)

- hayabusa.exe **json-timeline** -L **--timeline-start "2020-01-01 00:00:00 +00:00" --proven-rules --EID-filter** --UTC --enable-unsupported-rules --visualize-timeline --GeoIP <MAXMIND-DB-DIR> --HTML-report results.html **-p timesketch-verbose** --RFC-3339

  Takes less than 1 hour

  3+ hours left for timesketch import/analysis.


Second task: Analyze the data

Traditional Methodology

- For a thorough investigation you should check all *unique* alerts.
- Triage by level: Critical -> Low
- 20,713,599 detections
- but only ~300 unique alerts (low+)!
- Still, it will take too long...

Faster Methodology

- Identify and create a timeline of the high layer events:
  - Compromised machines
  - Compromised accounts/credentials
  - Exfiltrated data
  - Lateral movement
- Can be done quickly by pivoting on keywords (malware filenames, backdoor account names, C2 servers, etc...) instead of focusing on individual alerts.
- Perform deep dive investigations and fill in the details.

Suspicious Process Timeline

- **takajo.exe timeline-suspicious-processes** `-t ../hayabusa/DFIR-Report.jsonl -o DFIR-Report-susp-procs.csv`
- **Processing time:** 10 mins.
- Timeline of most of the malicious commands the attackers ran!
- You will need to filter out noisy alerts!
  (By filtering out just 4 rules, I reduced the results from 100,000+ lines to less than 600.)
- In less than 20 mins I was able to find out when the incident happened, what machines were compromised, attacker accounts, exfil, etc...

Pivot Keywords List

- Results from critical alerts:
  - Identified the attacker's accounts, source computer and a suspicious process.
  - Analysis Time: < 1 min.
- Results from high alerts:
  - Identified attacker's IP addresses, lots of malware and LOLBINS, and more compromised accounts.
  - Analysis Time: 10~20 mins.

Logon Summary/Timeline

- Quickly identified:
  - Accounts targeted.
    (268,575 failed logons for a certain user…)
  - Computers the attacker had access to.
  - Lateral movement.
- Also, "uncrackable" plaintext passwords in the username of failed logon events…

In The End…

- Only analyzing Hayabusa results is insufficient… (There are still rules that need to be created and there are also Mac and Linux machines too…)
- But can get 80% of the Windows analysis done in 20% of the time.


Win for Hayabusa + Takajo!

All Our Tools/Resources

https://github.com/Yamato-Security

Future Plans

- Hayabusa results import into SOF-ELK (Thanks to Phil Hagen!)
- Support for other logs besides Windows: Linux, macOS, cloud, etc...
- Much much more!
- Updates announced at @SecurityYamato

Many Thanks

- Nextron Systems and the Sigma community
- Mike Pilkington for the 608 CTF logs
- Hayabusa developers:
  - Akira Nishikawa (@nishikawaakira)
  - DustInDark (@hitenkoku)
  - Fukusuke Takahashi (@fukuseket)
  - Garigariganzy (@garigariganzy31)
  - ItiB (@itiB_S144)
  - James Takai / hachiyone (@hach1yon)
  - Kazuminn (@k47_um1n)

Contributing/Support

- Rust/Nim development
- Submit new rules to the Sigma repo
- Feedback on anything
- Creating logs for non-windows analysis
- Many things to do, just ask if you want to help
- Tweets, blog write-ups, GitHub stars, etc...
  to show your support!

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