Performing DFIR and Threat Hunting with Yamato Security OSS Tools and Community-Driven Knowledge

Akira Nishikawa and Fukusuke Takahashi
アキラ

Thank You for Your Understanding: Non-Native English Speakers

We are going to try to make this a fun presentation anyway!

Agenda

- Self-Introduction
- About Yamato Security and tools and resources
- Hayabusa
- Sigma
- Takajo
- Future Plans

Self-Introduction

Akira Nishikawa

- First core developer of Hayabusa
- 2007~ Freelance engineer
- 2021~ SaaS product security
- Now working at Kaminashi
- AWS Community Builder

Fukusuke Takahashi

- Latest core developer of Hayabusa
- DFIR, OSINT, SOAR at NTT-DATA CERT
- I fix bugs in open-source projects and bug hunt for vulnerabilities in my free time

About Yamato Security

- “Yamato” (大和) = “Japan”
- First created by Zach Mathis in 2012 to create a security community in Western Japan.
- Free/low-cost high-quality security training around the country
- Now over 2000 registered members
- Developing various open-source DFIR tools and resources since 2020.

Yamato Security tools and resources

- Hayabusa: DFIR timeline generator using native Sigma rules for Windows event logs
- Takajo: Hayabusa results analyzer
- Yamato Security's Windows Event Log Configuration Guide For DFIR And Threat Hunting
- Curation of Sigma Rules for Windows Event Logs
- Deprecated: WELA (Windows Event Log Analyzer)

Hayabusa?

- Who here has used or knows about Hayabusa?

HAYABUSA

About Hayabusa

- https://github.com/Yamato-Security/hayabusa
- Fast forensics and Threat Hunting CLI tool for Windows event logs
- Developed in Rust so it is very fast, cross-platform and safe from anti-forensics memory corruption exploits
- Many features: logon summaries, keyword searches, keyword extractions, sigma-based DFIR timeline generation
- Detects thousands of Windows attacks with 4000+ native Sigma and Hayabusa rules
- Input .evtx event logs and outputs to CSV or JSON/L for easy analysis

Hayabusa Features

Input

.evtx
.json
.jsonl

HAYABUSA

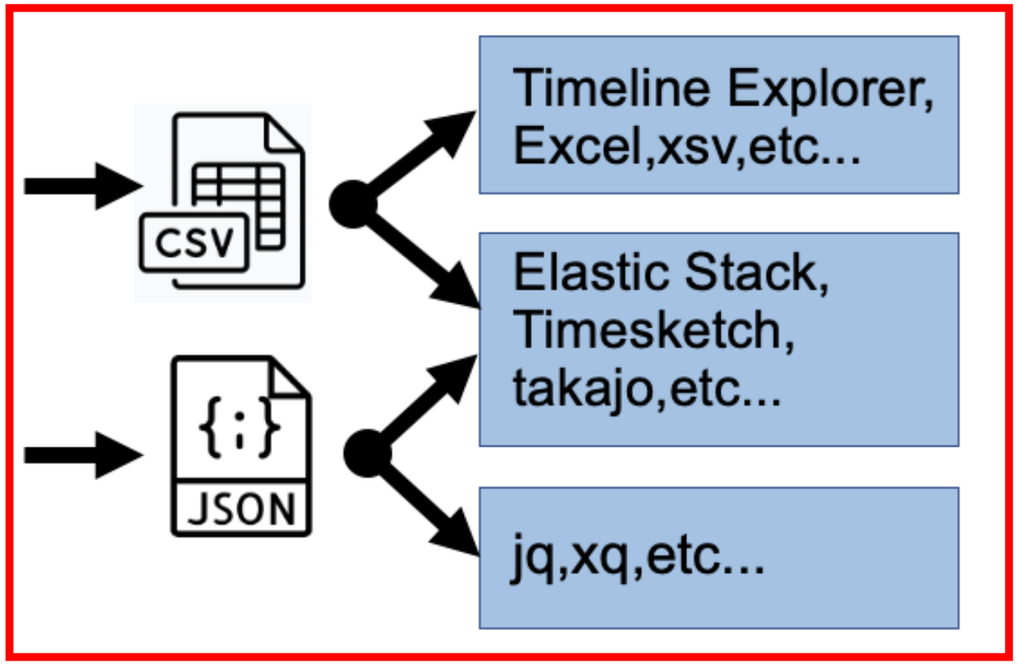
- DFIR Timeline Creation
- Event Metrics
- Logon Summary
- Pivot Keywords Generator
- Keyword Search

Hayabusa's DFIR Timeline Creation

HTML Report
(Results Summary)

HAYABUSA

- 4000+ 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

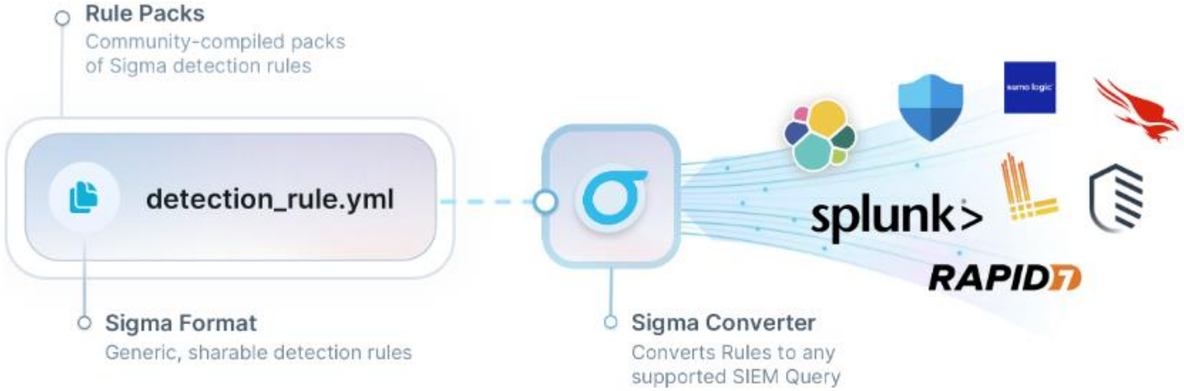


A	C	D	E	G	H	
Timestamp	Chann	Even	Lev	RuleTitle	RuleAuthor	Details
2019-08-05 18:39:	Sec	4624	med	Pass the Hash Activity 2	Dave Kennedy Jeff Warren (method) David Vassallo (rule)	Type: 9 TgtUser: IEUser SrcComp: - SrcIP: ::1 LID: 0x38f87e
2019-08-05 18:39:	Sec	4624	high	Successful Overpass the Hash Attempt	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} -Embeddir 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} -Embeddir 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} -Embeddir LID: 0x29126 PID: 8180

About Sigma

Sigma
SIEM Detection Format

- https://github.com/SigmaHQ/sigma
- Community-driven, generic, open-standard for detection rules for logs
- Actively developed and new rules added frequently
- Easy to write YAML format
- Can convert any Sigma rule into any SIEM query backend (Splunk, Elastic Stack, Qradar, Sentinel, etc…)



Sigma Rule Example

```
detection:
    selection_img:
        - Description|contains: '7-Zip'
        - Image|endswith:
            - '\7z.exe'
            - '\7zr.exe'
            - '\7za.exe'
        - OriginalFileName:
            - '7z.exe'
            - '7za.exe'
    selection_extension:
        CommandLine|contains:
            - '.dmp'
            - '.dump'
            - '.hdmp'
    condition: all of selection_*
```

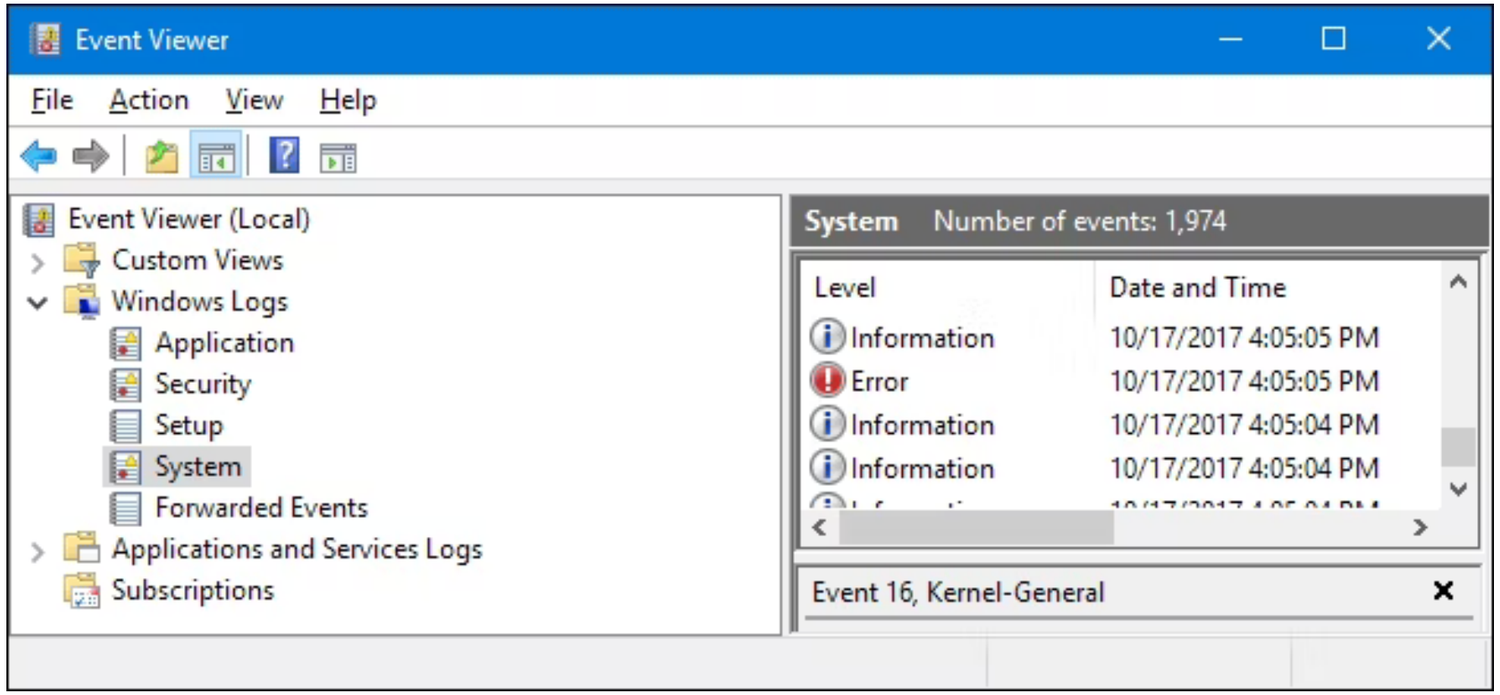
- condition statement at the bottom says that selection_img and selection_extension conditions have to all be true
- Note: Hyphens (-) express OR statements
- ”selection_img” says that the Description field needs to contain ‘7-Zip’ OR the Image field needs to end with various ‘7z.exe’ values OR the OriginalFileName field needs to be either ‘7z.exe’ OR ‘7za.exe’
- selection_extension says that the CommandLine field needs to contain one of the values: ‘.dmp’, etc…

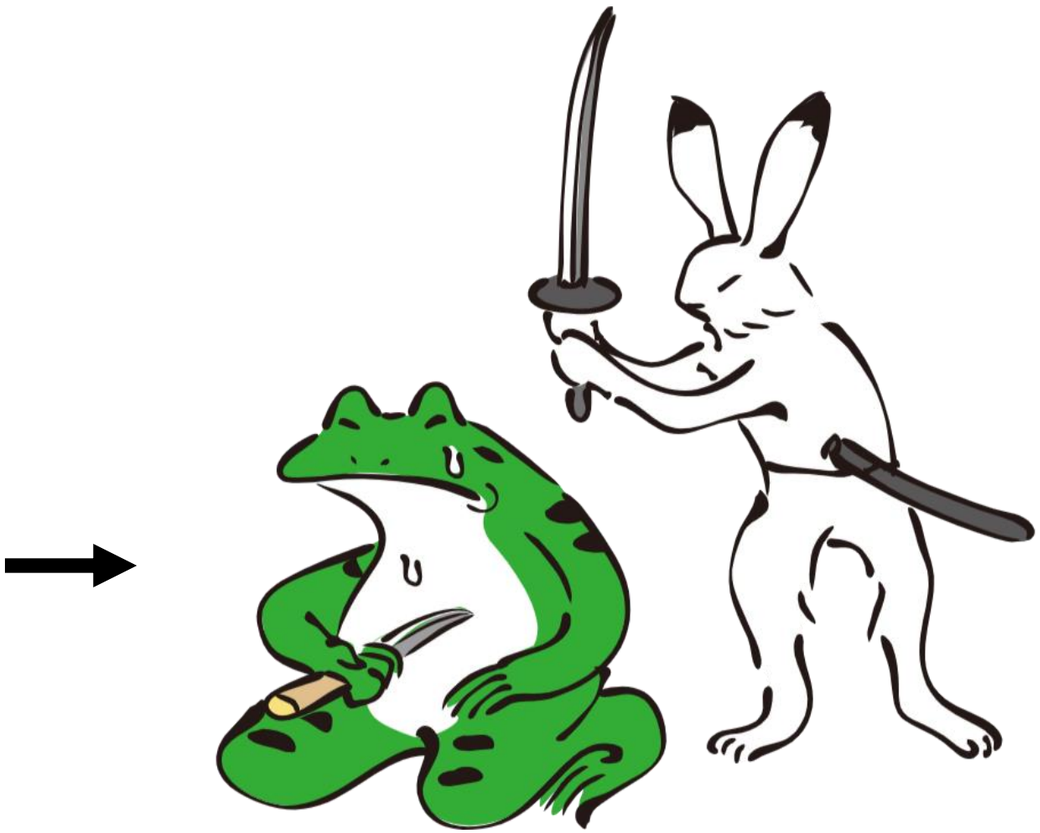
SECTOR BRIEFINGS

Rule converted to KQL

```
DeviceProcessEvents
| where (ProcessVersionInfoFileDescription contains "7-Zip" or (FolderPath endswith "\\7z.exe" or
FolderPath endswith "\\7zr.exe" or FolderPath endswith "\\7za.exe") or
(ProcessVersionInfoOriginalFileName in~ ("7z.exe", "7za.exe"))) and (ProcessCommandLine contains
".dmp" or ProcessCommandLine contains ".dump" or ProcessCommandLine contains ".hdmp")
```

The difficulties with Windows event log analysis





The difficulties with Windows event log analysis

- The default settings are completely insufficient for a proper investigation.
- Around 80% of the logs you want won’t get recorded.
- The most important process creation event is not enabled by default.
- 70%+ of events are just noise.
- Default log size is 1~20MB so evidence will quickly get overwritten.
- Logs are separated into 300+ logs.
- Field names are not consistent.
- Field values can be unintelligible.

The solutions for Windows event log analysis

- Properly configure Windows event log audit settings
- Use automated tools like Hayabusa and Takajo for analysis

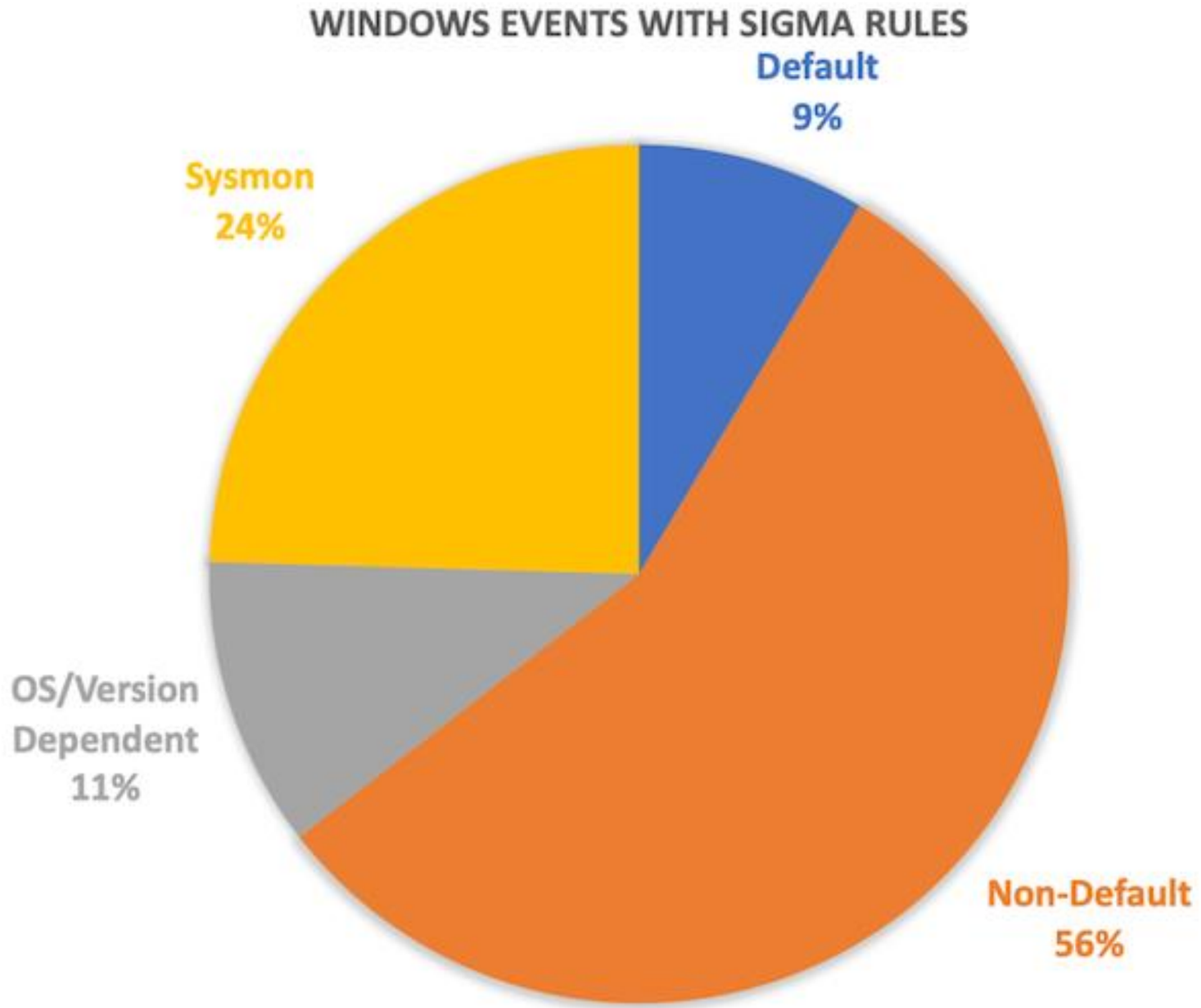
HAYABUSA

鷹匠

Takajō

Properly configuring Windows event logs

- (Yet another) Windows event log audit configuration guide: https://github.com/Yamato-Security/EnableWindowsLogSettings
- Shows the important events needed for detecting attacks
- Shows which events are enabled or not by default
- Shows what attacks you can detect if you enable certain settings
- All based on Sigma detection rules which are based on real attacks making this guide one of the most practical out there
- Includes a script to properly configure everything for you!


WINDOWS EVENTS WITH SIGMA RULES
Default
9%
Sysmon
24%
OS/Version
Dependent
11%
Non-Default
56%

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%

Directory Service Changes

Volume: `High`

Default settings: `No Auditing`

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

Notable Sigma rules:

- `(5136) (High) Powerview Add-DomainObjectAcl DCSync AD Extend Right` : Backdooring domain object to grant the rights associated with DCSync to a regular user or machine account.
- `(5136) (High) Active Directory User Backdoors` : Detects scenarios where one can control another users or computers account without having to use their credentials.
- `(5136) (Med) Possible DC Shadow`
- `(5136) (High) Suspicious LDAP-Attributes Used` : Detects LDAPFragger, a C2 tool that lets attackers route Cobalt Strike beacon data over LDAP attributes.

Event ID	Description	Sigma Rules	Hayabusa Rules	Level	Notes
5136	Directory Service Object Modified	6	Not Yet	Info~Crit	
5137	Directory Service Object Created	0	Not Yet	Info	
5138	Directory Service Object Undeleted	0	Not Yet	Info	
5139	Directory Service Object Moved	0	Not Yet	Info	
5141	Directory Service Object Deleted	0	Not Yet	Info	

Automating Analysis with Hayabusa

- Three use cases:
- Live Response: Run Hayabusa live on computers suspected of being compromised to determine if they are.
- Forensics Investigations: Collect the .evtx files from computers compromised in your environment and run Hayabusa against them on your forensics workstation or run Hayabusa remotely and send the results back to a SIEM, etc…
- Threat Hunting: Periodically run Hayabusa with updated rules as an agent on all the computers in your environment to check for compromise.

SECTOR BRIEFINGS

Hayabusa: Live Response

- Two options:
  - Run Hayabusa from an external USB drive and save results to that if you have physical access.
  - Copy over Hayabusa over the network, run it and save results to the hard drive then download the results.

Hayabusa: Live Response

- Things to be careful about:
  - Forensic artifacts may be over-written just like running any live response tool or saving data to the hard drive
  - Windows Defender and other anti-virus may give false positives on Sigma rules

```
detection:
    selection:
        - CommandLine|contains|all:
            - 'System.Management.Automation.AmsiUtils'
            - 'amsiInitFailed'
        - CommandLine|contains|all:
            - '[Ref].Assembly.GetType'
            - 'SetValue($null,$true)'
            - 'NonPublic,Static'
    condition: selection
```

← Microsoft thinks this YAML file is malicious code…

Hayabusa: Live Response

- Both issues mitigated in the latest 2.18.0 “SecTor” release!
- We now also provide release packages with config files embedded inside the Hayabusa binary and all rules and config files are XORed and stored in just two files to bypass any anti-virus signatures and minimize files written to the system.

Hayabusa: Forensics Investigations

- Two options: collect .evtx files or remotely run Hayabusa on all machines and send back results

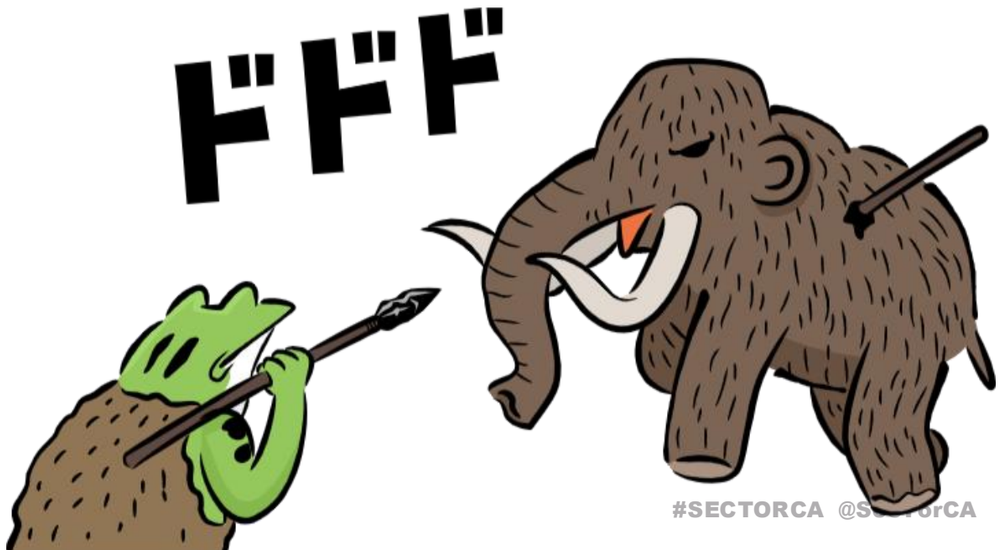
	Gather EVTX Files	Run Remotely
PROS	• Can detect attacks across different machines (ex: detect password spray attacks) • After .evtx files are downloaded, evidence cannot be deleted and is easier to perform different analysis techniques	• Processing is distributed so faster • Less data to transfer • More scalable! • Can re-run scans with the latest logs without having to re-download all the data.
CONS	• Takes time to copy off files (slower process) • May require large storage • May not be feasible to upload all evtx files from clients at the same time (not as scalable) • May take a long time to process the data	• Requires an agent to run on the machine • It might slow down machines • Harder to perform various analysis techniques

Hayabusa: Forensics Investigations

- Generally speaking…
- Small scale investigations: Gather EVTX Files (Copy off via USB, or create a script run by Group Policy, InTune, etc… to upload EVTX files to a file server)
- Large scale investigations: Run Remotely (Run Hayabusa via a Velociraptor agent, download the JSONL results and use that for analyzing with Takajo and/or importing into your favorite SIEM.)
- Note: we are currently looking into the best way to directly import Hayabusa results into a SIEM. Stay tuned!

Hayabusa: Threat Hunting

- Same as running Hayabusa remotely, but this time **before** an incident happens **(before you realize an incident has happened…)**
- Should be done **periodically** with the latest IoCs (Sigma rules)
- You will need a skilled person or team familiar with threat hunting to determine the false positives and perform deeper analysis



Advice on Sigma Rules

- You will get false positives!
- This is by design!
- The more specific a rule is (one designed for no false positives), the easier it will be to bypass and/or not catch new attacks
- You need a mix of specific rules and generic rules that catch abnormal behavior
- A skilled analyst is required to triage the results

Why Hayabusa for Sigma Rules?

- Performs Channel and Rule filtering for the most efficient processing
- Hayabusa has the best native support for Sigma rules
  => Highest number of true positives with lowest number of false positives!
- Note: Many tools now ”Support Sigma” but differ significantly on the which field modifiers they can handle. Hayabusa supports 99% of the rules in use (as well as modifiers in the specification not being used yet).
- Sigma rules are designed to be able to be converted to other backend queries but sometimes things get lost in translation…

Hayabusa's DFIR Timeline Creation

HTML Report
(Results Summary)

HAYABUSA

- 4000+ 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...

Hayabusa: 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'
```

- Hayabusa will convert ‘%%1842’ to ‘YES’, etc… so you do not have to memorize all of these message codes…

Hayabusa: Customizable Output

```
- minimal:                    %Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %RecordI
- standard:                   %Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %RecordI
 %ExtraFieldInfo%
- verbose:                    %Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %MitreTa
Tags%, %RecordID%, %RuleTitle%, %Details%, %ExtraFieldInfo%, %RuleFile%, %EvtxFile%
- all-field-info:             %Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %RecordI
nfo%, %RuleFile%, %EvtxFile%
- all-field-info-verbose:     %Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %MitreTa
Tags%, %RecordID%, %RuleTitle%, %AllFieldInfo%, %RuleFile%, %EvtxFile%
- super-verbose:              %Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %RuleTit
ifiedDate%, %Status%, %RecordID%, %Details%, %ExtraFieldInfo%, %MitreTactics%, %MitreTags%,
RuleCreati
```

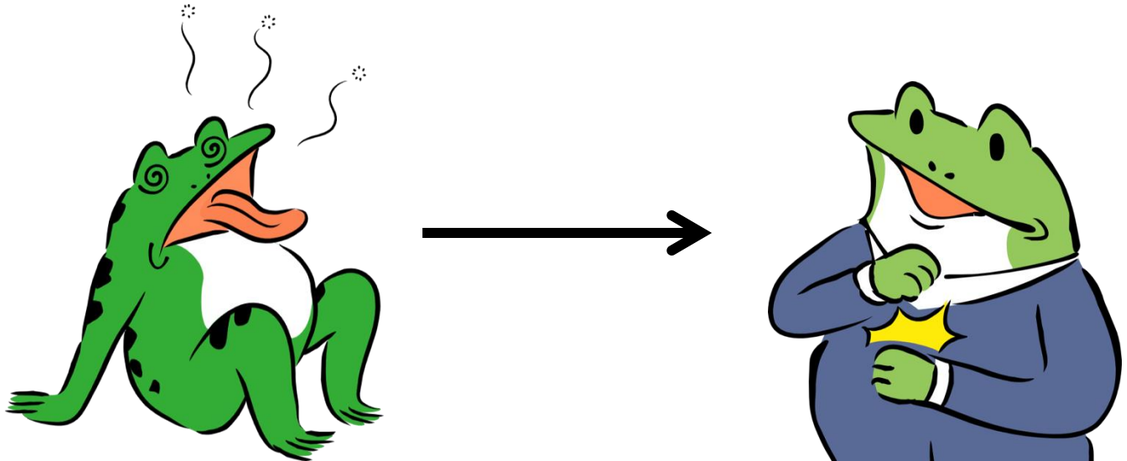
- Can customize as much or as little as you want depending on your preferences and situation.

Hayabusa: Data Reduction

- Unneeded events are ignored (significant noise reduction!)
- Needed data is abbreviated to bare minimum
- All the data can fit on one screen for easier and faster analysis!
- Examples:
  "An account was successfully logged on." → "Logon success"
  “Microsoft-Windows-Windows Firewall With Advanced Security/Firewall” → “Firewall”
- Standard Profile: 128 GB → 28 GB (-80%)
  Minimal Profile: 128 GB → 18 GB (-86%)

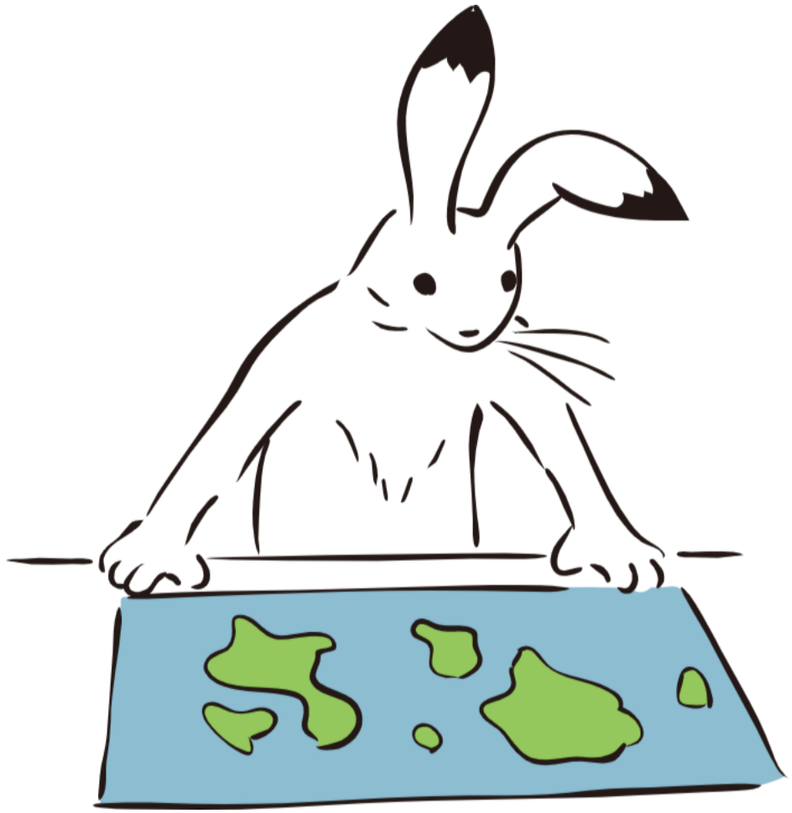
Hayabusa: Field Normalization

- 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 the fields above to the "SrcIP" field so it is easy to analyze with grep, jq, etc..



Hayabusa: Log Enrichment

- You can add IP address geolocation information from the SrcIP (Source IP) and TgtIP (Target IP) fields.
- 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…



Record Recovery and De-Duplication

- You can carve out deleted records in EVTX slack space
- This often results in duplicate events
- However, you can easily de-duplicate the results by adding another command line option
- Many more features! Check the readme!

鷹匠
Takajō

Takajo

- Takajo: “Falconer”
- Analyzes Hayabusa JSONL results and performs various analysis tasks
- Language: Nim (Easy to code script-like language that compiles to C)
- Memory safe, fast, multi-platform
- Over 30 useful analysis commands!
- Advanced HTML summary report!
- Major update release today!

Takajo Commands

- Extract commands: PowerShell ScriptBlock, Plaintext Credentials (new!)
- List commands: Domains, IP addresses, Binary Hashes, etc…
- Split commands: Split large timelines into smaller ones
- Stack commands: “Stack analysis” (xxx) on process command lines, computers, DNS queries, IP addresses, logons, processes, services, tasks, users, etc…
- Sysmon commands: Create process tree displays for malware

Takajo Commands

- Timeline commands: logons, USB usage, suspicious processes, tasks
- VirusTotal commands: lookup hashes, IP addresses, domains, etc… on VirusTotal
- TTP commands: visualize attacker TTPs as well as the kinds of attacks Sigma rules will can detect based on the MITRE ATT&CK framework

SECTOR BRIEFINGS

Reconnaissance
10 techniques

- Active Scanning (0/3)
- Gather Victim Host Information (0/4)
- Gather Victim Identity Information (0/3)
- Gather Victim Network Information (0/6)
- Gather Victim Org Information (0/4)
- Phishing for Information (0/4)
- Search Closed Sources (0/2)
- Search Open Technical Databases (0/5)
- Search Open Websites/Domains (0/3)
- Search Victim-Owned Websites

Resource Development
8 techniques

- Acquire Access
- Acquire Infrastructure (0/8)
- Compromise Accounts (0/3)
- Compromise Infrastructure (0/7)
- Develop Capabilities (1/4)
- Establish Accounts (0/3)
- Obtain Capabilities (1/6)
- Stage Capabilities (0/6)

Initial Access
10 techniques

- Content Injection
- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing (1/4)
- Replication Through Removable Media
- Supply Chain Compromise (0/3)
- Trusted Relationship
- Valid Accounts (0/4)

Execution
14 techniques

- Cloud Administration Command
- Command and Scripting Interpreter (5/9)
- Container Administration Command
- Deploy Container
- Exploitation for Client Execution
- Inter-Process Communication (1/3)
- Native API
- Scheduled Task/Job (2/5)
- Serverless Execution
- Shared Modules
- Software Deployment Tools
- System Services (1/2)
- User Execution (1/3)
- Windows Management Instrumentation

Persistence
20 techniques

- Account Manipulation (0/6)
- BITS Jobs
- Boot or Logon Autostart Execution (4/14)
- Boot or Logon Initialization Scripts (1/5)
- Browser Extensions
- Compromise Client Software Binary
- Create Account (1/3)
- Create or Modify System Process (1/4)
- Event Triggered Execution (8/16)
- External Remote Services
- Hijack Execution Flow (2/12)
- Implant Internal Image
- Modify Authentication Process (0/8)
- Office Application Startup (0/6)
- Power Settings
- Pre-OS Boot (0/5)
- Scheduled

Privilege Escalation
14 techniques

- Abuse Elevation Control Mechanism (1/5)
- Access Token Manipulation (3/5)
- Account Manipulation (0/6)
- Boot or Logon Autostart Execution (4/14)
- Boot or Logon Initialization Scripts (1/5)
- Create or Modify System Process (1/4)
- Domain Policy Modification (0/2)
- Escape to Host
- Event Triggered Execution (8/16)
- Exploitation for Privilege Escalation
- Hijack Execution Flow (2/12)
- Process Injection (3/12)
- Scheduled Task/Job (2/5)
- Valid

Defense Evasion
43 techniques

- Abuse Elevation Control Mechanism (1/5)
- Access Token Manipulation (3/5)
- BITS Jobs
- Build Image on Host
- Debugger Evasion
- Deobfuscate/Decode Files or Information
- Deploy Container
- Direct Volume Access
- Domain Policy Modification (0/2)
- Execution Guardrails (0/1)
- Exploitation for Defense Evasion
- File and Directory Permissions Modification (1/2)
- Hide Artifacts (2/11)
- Hijack Execution Flow (2/12)
- Impair Defenses (3/11)
- Impersonation
- Indicator Removal (2/9)
- Indirect Command Execution
- Masquerading (2/9)
- Modify Authentication Process (0/8)
- Modify Cloud Compute Infrastructure (0/5)
- Modify Registry

Credential Access
17 techniques

- Adversary-in-the-Middle (1/3)
- Brute Force (0/4)
- Credentials from Password Stores (1/6)
- Exploitation for Credential Access
- Forced Authentication
- Forge Web Credentials (0/2)
- Input Capture (0/4)
- Modify Authentication Process (0/8)
- Multi-Factor Authentication Interception
- Multi-Factor Authentication Request Generation
- Network Sniffing
- OS Credential Dumping (6/8)
- Steal Application Access Token
- Steal or Forge Authentication Certificates
- Steal or Forge Kerberos Tickets (1/4)

Discovery
32 techniques

- Account Discovery (2/4)
- Application Window Discovery
- Browser Information Discovery
- Cloud Infrastructure Discovery
- Cloud Service Dashboard
- Cloud Service Discovery
- Cloud Storage Object Discovery
- Container and Resource Discovery
- Debugger Evasion
- Device Driver Discovery
- Domain Trust Discovery
- File and Directory Discovery
- Group Policy Discovery
- Log Enumeration
- Network Service Discovery
- Network Share Discovery
- Network Sniffing
- Password Policy Discovery
- Peripheral Device Discovery
- Permission Groups Discovery (2/3)
- Process Discovery

Lateral Movement
9 techniques

- Exploitation of Remote Services
- Internal Spearphishing
- Lateral Tool Transfer
- Remote Service Session Hijacking (0/2)
- Remote Services (4/8)
- Replication Through Removable Media
- Software Deployment Tools
- Taint Shared Content
- Use Alternate Authentication Material (1/4)

Collection
17 techniques

- Adversary-in-the-Middle (1/3)
- Archive Collected Data (0/3)
- Audio Capture
- Automated Collection
- Browser Session Hijacking
- Clipboard Data
- Data from Cloud Storage
- Data from Configuration Repository (0/2)
- Data from Information Repositories (0/3)
- Data from Local System
- Data from Network Shared Drive
- Data from Removable Media
- Data Staged (0/2)
- Email Collection (0/3)
- Input Capture (0/4)
- Screen Capture

Command and Control
17 techniques

- Application Layer Protocol (1/4)
- Communication Through Removable Media
- Content Injection
- Data Encoding (1/2)
- Data Obfuscation (0/3)
- Dynamic Resolution (0/3)
- Encrypted Channel (0/2)
- Fallback Channels
- Ingress Tool Transfer
- Multi-Stage Channels
- Non-Application Layer Protocol
- Non-Standard Port
- Protocol Tunneling
- Proxy (2/4)
- Remote Access Software
- Traffic Signaling (0/2)
- Web Service (0/3)

Exfiltration
9 techniques

- Automated Exfiltration (0/1)
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol (0/3)
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium (0/1)
- Exfiltration Over Physical Medium (0/1)
- Exfiltration Over Web Service (0/4)
- Scheduled Transfer
- Transfer Data to Cloud Account

Impact
14 techniques

- Account Access Removal
- Data Destruction
- Data Encrypted for Impact
- Data Manipulation (0/3)
- Defacement (0/2)
- Disk Wipe (0/2)
- Endpoint Denial of Service (0/4)
- Financial Theft
- Firmware Corruption
- Inhibit System Recovery
- Network Denial of Service (0/2)
- Resource Hijacking
- Service Stop
- System Shutdown/Reboot

Most Important Takajo Commands

- **automagic**: Automates most commands for you!
- **html-report**: Creates HTML summary reports!
- **html-server**: Creates a more advanced dynamic HTML summary report! (Just released today!)

Summary

Total detections

SEVERITY	NUMBER OF DETECTIONS	DETECTION RATE
critical	357	1.16%
high	200	0.65%
med	784	2.54%
low	26894	87.07%
info	2653	8.59%

Unique detections

SEVERITY	NUMBER OF DETECTIONS	DETECTION RATE
critical	3	3.85%
high	9	11.54%
med	23	29.49%
low	15	19.23%
info	28	35.90%

Dates with most total detections

SEVERITY	NUMBER OF DETECTIONS	DETECTION RATE
critical	2023-10-21	352

Takajo

Summary

Total detections

SEVERITY	NUMBER OF DETECTIONS	DETECTION RATE
critical	357	1.16%
high	200	0.65%
med	784	2.54%
low	26894	87.07%
info	2653	8.59%

Unique detections

SEVERITY	NUMBER OF DETECTIONS	DETECTION RATE
critical	3	3.85%
high	9	11.54%
med	23	29.49%
low	15	19.23%
info	28	35.90%

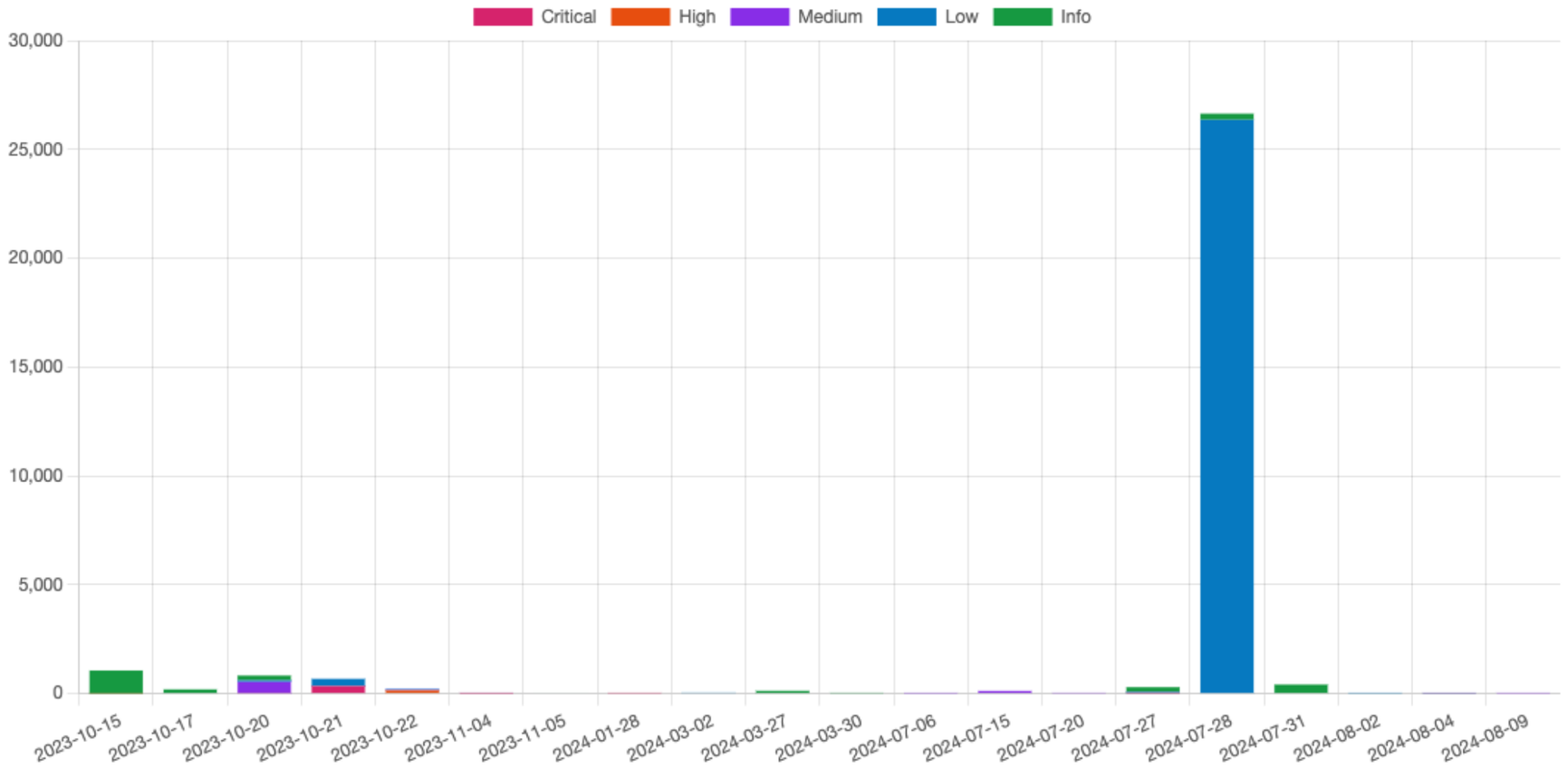
Takajo

Summary

- critical alerts (357)
- high alerts (200)
- med alerts (784)
- low alerts (26894)

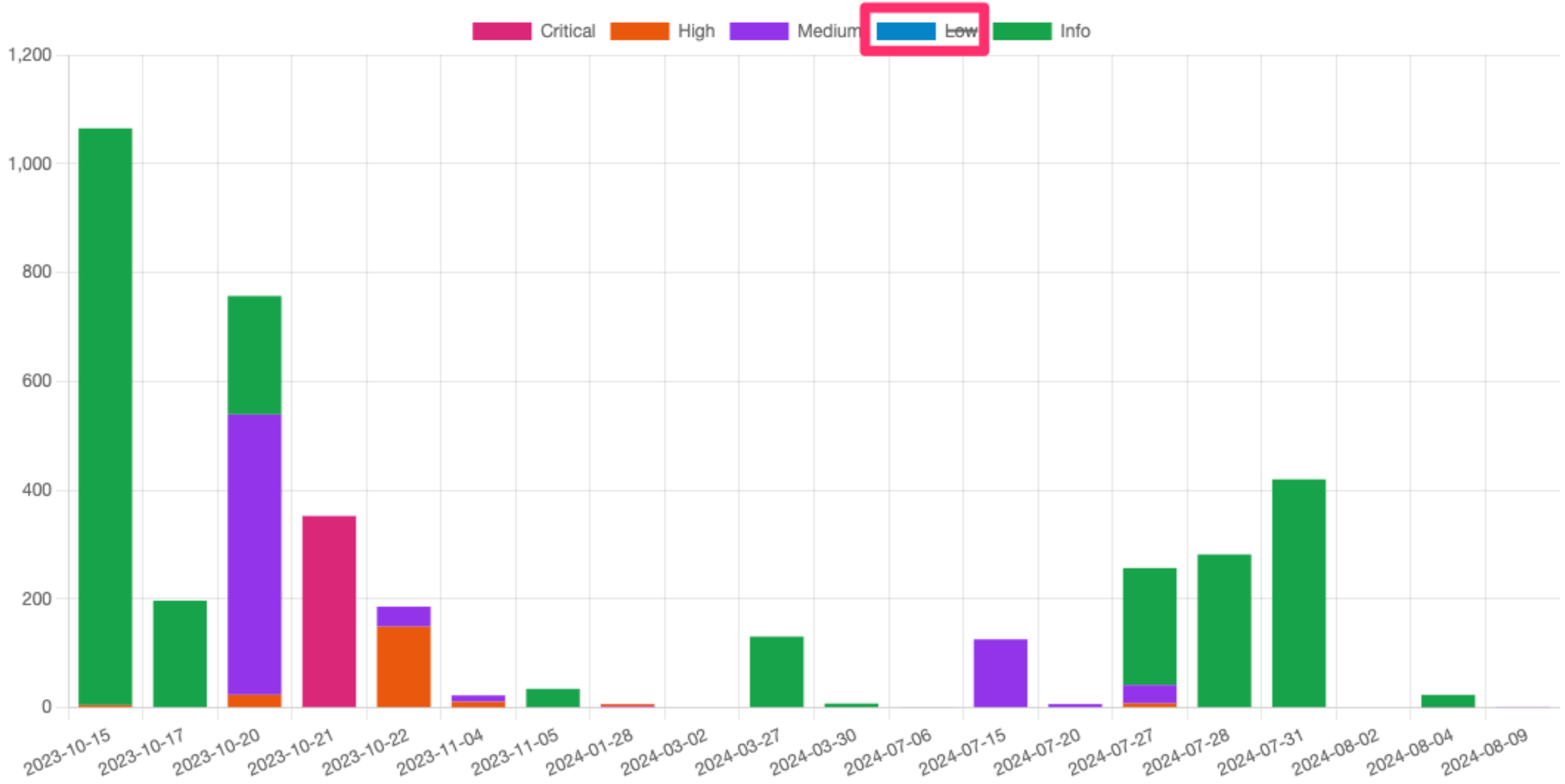
mouse

357	199	736	26893	2589
Critical	High	Medium	Low	Information



mouse

357	199	736	26893	2589
Critical	High	Medium	Low	Information



Detection Rule List

ALERT TITLE	RULE PATH	SEVERITY
Antivirus Exploitation Framework Detection	av_exploiting.yml	critical
Antivirus Password Dumper Detection	av_password_dumper.yml	critical
Defender Alert (Severe)	Defender_1116_Crit_Alert.yml	critical
Antivirus Hacktool Detection	av_hacktool.yml	high
Antivirus Relevant File Paths Alerts	av_relevant_files.yml	high
Defender Alert (High)	Defender_1116_High_Alert.yml	high
Microsoft Defender Blocked from Loading Unsigned DLL	win_security_mitigations_defender_load_unsigned_dll.yml	high
Microsoft Defender Tamper Protection Trigger	win_defender_tamper_protection_trigger.yml	high
Powershell Token Obfuscation - Powershell	posh_ps_token_obfuscation.yml	high

mouse

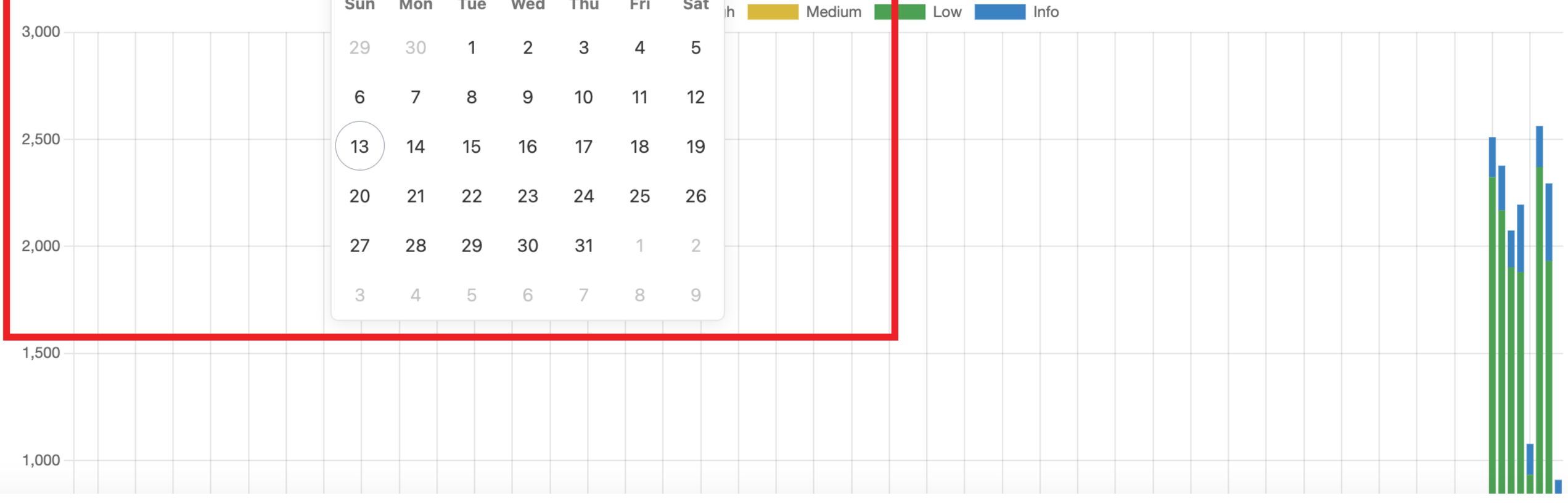
385	222	800	15035	3603
Critical	High	Medium	Low	Information

2024-06-01 ~ to date Search past30days

← New! (Dynamic Web Sever)

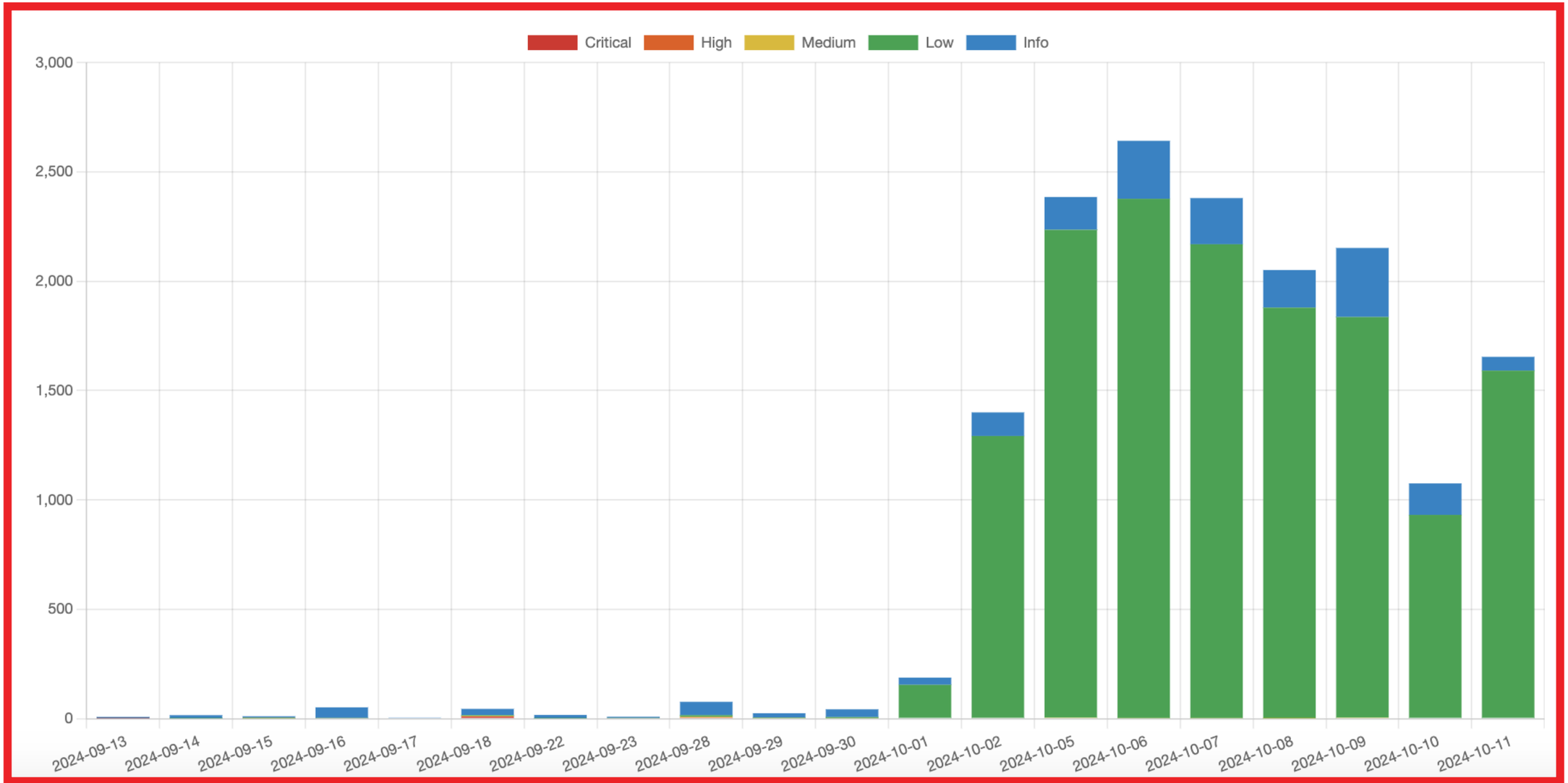
October 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9



13	2	31	14467	1710
Critical	High	Medium	Low	Information

2024-01-01 ~ 2024-08-08 Search past30days ← Past 30 days



Future Plans

Future Plans

Analysis with generative AI

- ”Analyze the Hayabusa and Takajo results and write a 10-page forensics report”
- => “OK! No problem”
- Challenges:
  - Needs to keep all of the data local
  - Needs to be accurate
  - Current generative AI does not produce long reports so we would first have to script things to ask many questions

Analysis with machine learning (UEBA-type detection)

- Example: Find any abnormal logins

Suzaku: A Sigma-based event log analyzer for cloud logs (AWS, Azure, GCP)

Please let us know if you want to help out with any of these!

Thank you so much for listening!

If you like our tools, please consider supporting us with a GitHub star!

Please give us feedback on what we can do better. Please also contact us if you want to help out.

Check out the latest release information on X-Twitter!

Follow:
@SecurityYamato

Thank you for your attention!