



HAYABUSA

[English] | [日本語]

Stable Version v1.8.1 GitHub Downloads 18k GitHub Stars 927 Contributors 13

Black Hat Arsenal Asia 2022 CODE BLUE Bluebox 2022 Maintenance Level Actively Developed rs report A+

[Follow @SecurityYamato](#) 933

Hayabusaについて

Hayabusaは、日本の[Yamato Security](#)グループによって作られたWindowsイベントログのファストフォレンジックタイムライン作成および脅威ハンティングツールです。 Hayabusaは日本語で「ハヤブサ」を意味し、ハヤブサが世界で最も速く、狩猟(hunting)に優れ、とても訓練しやすい動物であることから選ばれました。 Rustで開発され、マルチスレッドに対応し、可能な限り高速に動作するよう配慮されています。 SigmaルールをHayabusaルール形式に変換するツールも提供しています。 Hayabusaの検知ルールもSigmaと同様にYML形式であり、カスタマイズ性や拡張性に優れます。稼働中のシステムで実行してライブ調査することも、複数のシステムからログを収集してオフライン調査することも可能です。また、[Velociraptor](#)と[Hayabusa artifact](#)を用いることで企業向けの広範囲なスレットハンティングとインシデントレスポンスにも活用できます。出力は一つのCSVタイムラインにまとめられ、Excel、Timeline Explorer、Elastic Stack、Timesketch等で簡単に分析できるようになります。

関連プロジェクト

- [EnableWindowsLogSettings](#) - Sigmaベースの脅威ハンティングと、Windowsイベントログのファストフォレンジックタイムライン生成ツール。
- [Hayabusa Rules](#) - Hayabusaのための検知ルール。
- [Hayabusa Sample EVTXs](#) - Hayabusa/Sigma検出ルールをテストするためのサンプルevtxファイル。
- [Takajo](#) - Hayabusa結果の解析ツール。
- [WELA \(Windows Event Log Analyzer\)](#) - PowerShellで書かれたWindowsイベントログの解析ツール。

目次

- [Hayabusaについて](#)

- 関連プロジェクト
 - 目次
 - 主な目的
 - スレット(脅威)ハンティングと企業向けの広範囲なDFIR
 - フォレンジックタイムラインの高速生成
- スクリーンショット
 - 起動画面
 - ターミナル出力画面
 - イベント頻度タイムライン出力画面 (`-T`オプション)
 - 結果サマリ画面
 - HTMLの結果サマリ (`-H`オプション)
 - Excelでの解析
 - Timeline Explorerでの解析
 - Criticalアラートのフィルタリングとコンピュータごとのグルーピング
 - Elastic Stackダッシュボードでの解析
 - Timesketchでの解析
- タイムラインのサンプル結果
- 特徴&機能
- ダウンロード
- Gitクローン
- アドバンス: ソースコードからのコンパイル (任意)
 - Rustパッケージの更新
 - 32ビットWindowsバイナリのクロスコンパイル
 - macOSでのコンパイルの注意点
 - Linuxでのコンパイルの注意点
 - LinuxのMUSLバイナリのクロスコンパイル
- Hayabusaの実行
 - 注意: アンチウィルス/EDRの誤検知と遅い初回実行
 - Windows
 - Linux
 - macOS
- 主なコマンド
- 使用方法
 - デフォルトのヘルプメニュー
 - `csv-timeline`コマンド
 - `json-timeline`コマンド
 - `logon-summary`コマンド
 - `metrics`コマンド
- `pivot-keywords-list`コマンド
 - `update-rules`コマンド
 - `level-tuning`コマンド
 - `set-default-profile`コマンド
 - `list-profiles`コマンド
 - 使用例
 - ピボットキーワードの作成
 - ログオン情報の要約

- サンプルevtxファイルでHayabusaをテストする
- Hayabusaの出力
 - プロファイル
 - 1. `minimal`プロファイルの出力
 - 2. `standard`プロファイルの出力
 - 3. `verbose`プロファイルの出力
 - 4. `all-field-info`プロファイルの出力
 - 5. `all-field-info-verbose`プロファイルの出力
 - 6. `super-verbose`プロファイルの出力
 - 7. `timesketch-minimal`プロファイルの出力
 - 8. `timesketch-verbose`プロファイルの出力
 - プロファイルの比較
 - Profile Field Aliases
- Levelの省略
- MITRE ATT&CK戦術の省略
- Channel情報の省略
- その他の省略
 - プログレスバー
 - 標準出力へのカラー設定
 - 結果のサマリ
 - イベント頻度タイムライン
- Hayabusaルール
 - Hayabusa v.s. 変換されたSigmaルール
 - 検知ルールのチューニング
 - 検知レベルのlevelチューニング
 - イベントIDフィルタリング
- その他のWindowsイベントログ解析ツールおよび関連リソース
- Windowsイベントログ設定のススメ
- Sysmon関係のプロジェクト
- コミュニティによるドキュメンテーション
 - 英語
 - 日本語
- 貢献
- バグの報告
- ライセンス
- Twitter

主な目的

スレット(脅威)ハンティングと企業向けの広範囲なDFIR

Hayabusaには現在、3000以上のSigmaルールと約150のHayabusa検知ルールがあり、定期的にルールが追加されています。 [Velociraptor](#)の[Hayabusa artifact](#)を用いることで企業向けの広範囲なスレットハンティングだけでなくDFIR(デジタルフォレンジックとインシデントレスポンス)にも無料で利用することができます。この2つのオープンソースを組み合わせることで、SIEMが設定されていない環境でも実質的に遡及してSIEMを再現することができます。具体的な方法は[Eric Capuano](#)の[こちら](#)の動画で学ぶことができます。最終的な目標はインシデントレスポンス

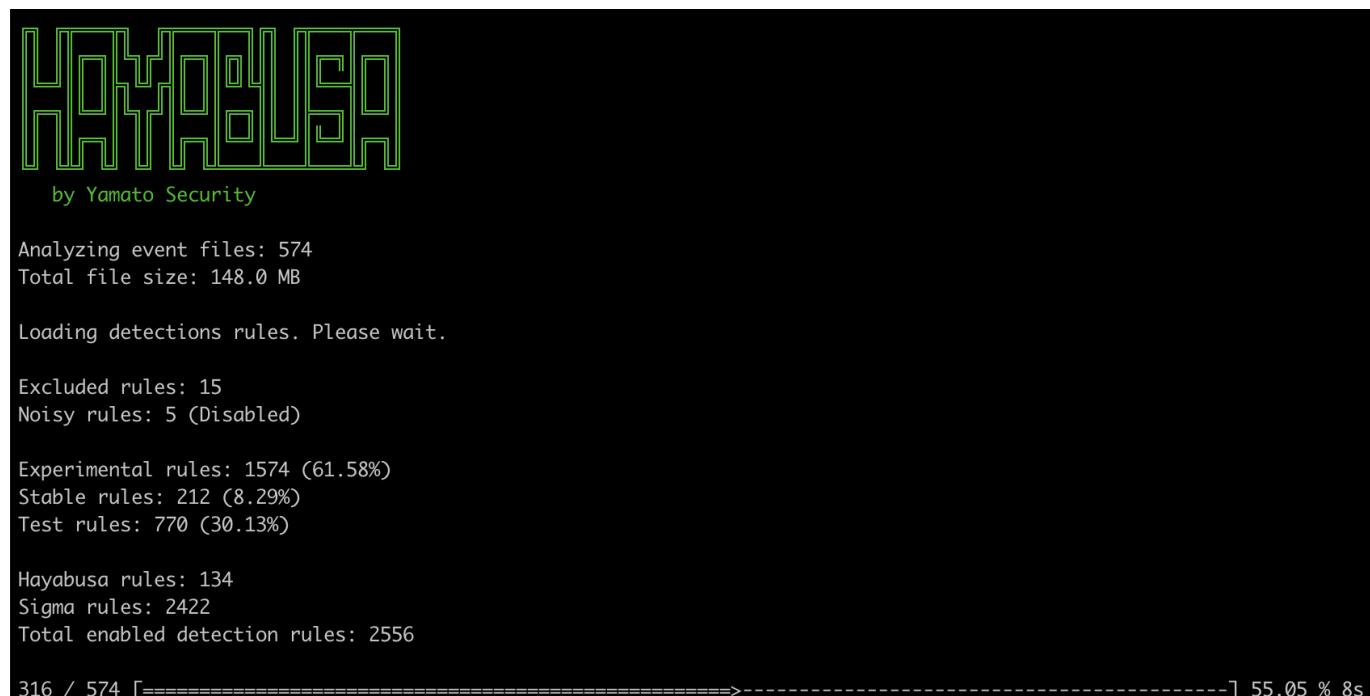
や定期的なスレットハンティングのために、HayabusaエージェントをすべてのWindows端末にインストールして、中央サーバーにアラートを返す仕組みを作ることです。

フォレンジックタイムラインの高速生成

Windowsのイベントログは、1) 解析が困難なデータ形式であること 2) データの大半がノイズであり調査に有用でないことから、従来は非常に長い時間と手間がかかる解析作業となっていました。 Hayabusaは、有用なデータのみを抽出し、専門的なトレーニングを受けた分析者だけでなく、Windowsのシステム管理者であれば誰でも利用できる読みやすい形式で提示することを主な目的としています。 Hayabusaは従来のWindowsイベントログ分析解析と比較して、分析者が20%の時間で80%の作業を行えるようにすることを目指しています。

スクリーンショット

起動画面



The screenshot shows the Hayabusa application's main interface. At the top, there is a large, stylized logo for "HAYABUSA" in green. Below the logo, it says "by Yamato Security". The main area displays the following text:

```
Analyzing event files: 574  
Total file size: 148.0 MB  
  
Loading detections rules. Please wait.  
  
Excluded rules: 15  
Noisy rules: 5 (Disabled)  
  
Experimental rules: 1574 (61.58%)  
Stable rules: 212 (8.29%)  
Test rules: 770 (30.13%)  
  
Hayabusa rules: 134  
Sigma rules: 2422  
Total enabled detection rules: 2556  
  
316 / 574 [=====>-----] 55.05 % 8s
```

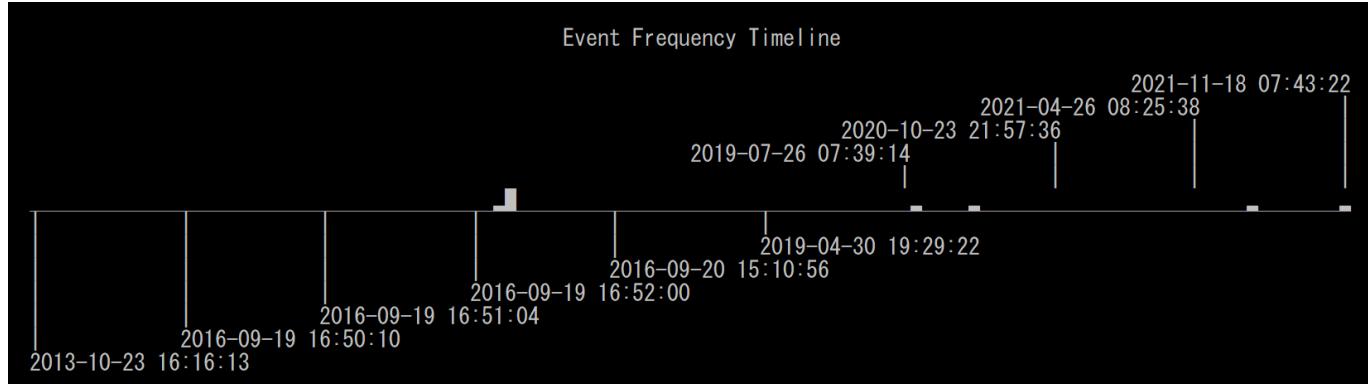
ターミナル出力画面

```

2021-12-13 17:21:30.845 +09:00 || fs03vuln.offsec.lan || Sec || 5145 || info || 1176260 || NetShare File Access || User: admmig | ShareName: \\*\C$ | SharePath: \??\C:\ | Path: Windows\System32\Drivers\etc | IP-Addr: 10.23.23.9 | LID: 0x8ca6e1d
2021-12-13 21:55:45.250 +09:00 || rootdc1.offsec.lan || Sys || 7045 || info || 1467331 || Svc Installed || Svc: BT0BTO | Path: %COMSPEC% /Q /c e
cho cd ^> \\127.0.0.1\%C$\_output 2^>^&1 > %TEMP%\execute.bat & %COMSPEC% /Q /c %TEMP%\execute.bat & del %TEMP%\execute.bat | Acct: Local
System | StartType: demand start
2021-12-13 21:55:45.250 +09:00 || - - - || low || Rare Service Installations || [condition] count() by ServiceName < 5 in timeframe [result] count:1 ServiceName:BT0BTO timeframe:7d
2021-12-13 21:55:45.250 +09:00 || rootdc1.offsec.lan || Sec || 4697 || info || 236864754 || Svc Installed || Svc: BT0BTO | Path: %COMSPEC% /Q /c
echo cd ^> \\127.0.0.1\%C$\_output 2^>^&1 > %TEMP%\execute.bat & %COMSPEC% /Q /c %TEMP%\execute.bat & del %TEMP%\execute.bat | User: adm
 mig | SvcAccount: LocalSystem | SvcType: 0x10 | SvcStartType: 3 | LID: 0x2cff42b44
2021-12-14 23:42:48.182 +09:00 || rootdc1.offsec.lan || Sec || 4776 || info || 237294513 || NTLM Logon To Local Account || User: hack1 | Comp: a
ttacker | Status: 0x0
2021-12-14 23:42:48.182 +09:00 || rootdc1.offsec.lan || Sec || 4624 || info || 237294514 || Logon (Type 3 Network) || User: hack1 | Comp: attack
er | IP-Addr: 10.23.123.11 | LID: 0x308fabb0c
2021-12-14 23:42:48.182 +09:00 || rootdc1.offsec.lan || Sec || 4624 || med || 237294514 || Pass the Hash Activity 2 || User: hack1 | Comp: att
acker | IP-Addr: 10.23.123.11 | Proc: - | LID: 0x308fabb0c
2021-12-14 23:42:48.690 +09:00 || rootdc1.offsec.lan || Sec || 4776 || info || 237294516 || NTLM Logon To Local Account || User: hack1 | Comp:
| Status: 0x0
2021-12-14 23:42:48.690 +09:00 || rootdc1.offsec.lan || Sec || 4624 || info || 237294517 || Logon (Type 3 Network) || User: hack1 | Comp: - | IP
-Addr: 10.23.123.11 | LID: 0x308fb82ad
2021-12-14 23:42:48.693 +09:00 || rootdc1.offsec.lan || Sec || 5140 || info || 237294519 || NetShare Access || User: hack1 | ShareName: \\*\IPC$ | SharePath: | IP-Addr: 10.23.123.11 | LID: 0x308fb82ad
2021-12-14 23:42:48.908 +09:00 || rootdc1.offsec.lan || Sec || 4781 || high || 237294532 || Suspicious Computer Account Name Change CVE-2021-42
287 || OldTgtUser: compnay-88$ | NewTgtUser: rootdc1 | TgtSID: S-1-5-21-4230534742-2542757381-3142984815-1296 | User: hack1 | SID: S-1-5-2
1-4230534742-2542757381-3142984815-1234 | Domain: OFFSEC | TgtDomain: OFFSEC | PrivList: - | LID: 0x308fabb0c
2021-12-14 23:42:48.908 +09:00 || rootdc1.offsec.lan || Sec || 4781 || high || 237294532 || Suspicious Computer Account Name Change CVE-2021-42
287 || OldTgtUser: compnay-88$ | NewTgtUser: rootdc1 | TgtSID: S-1-5-21-4230534742-2542757381-3142984815-1296 | User: hack1 | SID: S-1-5-2
1-4230534742-2542757381-3142984815-1234 | Domain: OFFSEC | TgtDomain: OFFSEC | PrivList: - | LID: 0x308fabb0c
2021-12-14 23:42:49.222 +09:00 || rootdc1.offsec.lan || Sec || 4768 || info || 237294534 || Kerberos TGT Requested || User: rootdc1 | Svc: krbtg
t | IP-Addr: ::ffff:10.23.123.11 | Status: 0x0 | PreAuthType: 2

```

イベント頻度タイムライン出力画面 (-Tオプション)



結果サマリ画面

Results Summary:

Events with hits / Total events: 19,545 / 76,967 (Data reduction: 57,422 events (74.61%))

Total | Unique detections: 32,684 | 554
 Total | Unique critical detections: 46 (0.14%) | 18 (3.25%)
 Total | Unique high detections: 6,141 (18.79%) | 250 (45.13%)
 Total | Unique medium detections: 1,472 (4.50%) | 156 (28.16%)
 Total | Unique low detections: 6,771 (20.72%) | 76 (13.72%)
 Total | Unique informational detections: 18,254 (55.85%) | 54 (9.75%)

Dates with most total detections:

critical: 2019-07-19 (15), **high:** 2016-09-20 (3,656), **medium:** 2019-05-19 (165), **low:** 2016-09-20 (3,780), **informational:** 2016-08-19 (2,105)

Top 5 computers with most unique detections:

critical: MSEdgeWIN10 (6), IEWIN7 (3), FS03.offsec.lan (2), rootdc1.offsec.lan (2), srvdefender01.offsec.lan (2)
high: MSEdgeWIN10 (109), IEWIN7 (70), FS03.offsec.lan (31), fs03vuln.offsec.lan (27), IE10Win7 (23)
medium: MSEdgeWIN10 (62), IEWIN7 (38), FS03.offsec.lan (16), IE10Win7 (15), PC01.example.corp (14)
low: MSEdgeWIN10 (35), IEWIN7 (18), FS03.offsec.lan (16), fs03vuln.offsec.lan (13), IE10Win7 (11)
informational: MSEdgeWIN10 (18), IEWIN7 (17), fs01.offsec.lan (16), PC01.example.corp (13), IE8Win7 (12)

Top critical alerts:

Sticky Key Like Backdoor Usage (10)
 Meterpreter or Cobalt Strike Getsystem Service Installation (6)
 Active Directory Replication from Non Machine Account (6)
 Windows Defender Alert (4)
 WannaCry Ransomware (4)

Top high alerts:

Metasploit SMB Authentication (3,562)
 Malicious Svc Possibly Installed (271)
 Susp Svc Installed (257)
 PowerShell Scripts Installed as Services (253)
 Suspicious Service Installation Script (250)

Top medium alerts:

Potentially Malicious PwSh (235)
 Proc Injection (104)
 Reg Key Value Set_Sysmon Alert (103)
 Suspicious Remote Thread Target (93)
 Cscript Visual Basic Script Execution (60)

Top low alerts:

Logon Failure_Wrong Password (3,564)
 Susp CmdLine (Possible LOLBIN) (1,418)
 Non Interactive PowerShell (325)
 Rare Service Installations (321)
 Windows Processes Suspicious Parent Directory (282)

Top informational alerts:

Proc Exec (11,173)
 NetShare File Access (2,564)
 PwSh Scriptblock (789)
 PwSh Pipeline Exec (680)
 NetShare Access (433)

Explicit Logon (342)
 Svc Installed (331)
 New Non-USB PnP Device (268)
 Logon (Type 3 Network) (228)
 File Created (210)

Elapsed Time: 00:00:28.827

HTMLの結果サマリ (-Hオプション)

General Overview

- Start time: 2022/10/02 03:16
- Excluded rules: 12
- Noisy rules: 5 (Disabled)
- Experimental rules: 2036 (67.24%)
- Stable rules: 213 (7.03%)
- Test rules: 779 (25.73%)
- Hayabusa rules: 138
- Sigma rules: 2890
- Total enabled detection rules: 3028
- Elapsed Time: 00:00:23.844

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)
 - high: 2016-09-20 (3,656)
 - medium: 2019-05-19 (165)
 - low: 2016-09-20 (3,780)
 - informational: 2016-08-19 (2,105)

Computers with most unique critical detections:

- MSEDGEWIN10 (6)
- IEWIN7 (3)
- srvdefender01.offsec.lan (2)
- FS03.offsec.lan (2)
- rootdc1.offsec.lan (2)
- alice.insecurebank.local (1)
- IE10Win7 (1)
- DC1.insecurebank.local (1)
- win10-02.offsec.lan (1)
- fs01.offsec.lan (1)
- fs03vuln.offsec.lan (1)
- DESKTOP-PIU87N6 (1)

Computers with most unique high detections:

- MSEDGEWIN10 (112)
- IEWIN7 (72)

Top critical alerts:

- [Sticky Key Like Backdoor Usage](#) (10)
- [Meterpreter or Cobalt Strike Getsystem Service Installation](#) (6)
- [Active Directory Replication from Non Machine Account](#) (6)
- [Windows Defender Alert](#) (4)
- [WannaCry Ransomware](#) (4)
- [Dumpert Process Dumper](#) (3)
- [SystemNightmare Exploitation Script Execution](#) (2)
- [Mimikatz MemSSP Default Log File Creation](#) (2)
- [Suspicious LSASS Process Clone](#) (2)
- [TrustedPath UAC Bypass Pattern](#) (2)
- [SMB Relay Attack Tools](#) (1)
- [smbexec.py Service Installation](#) (1)
- [Malicious Named Pipe](#) (1)
- [CobaltStrike Service Installations in Registry](#) (1)
- [Lsass Memory Dump via Comsvcs DLL](#) (1)

Top high alerts:

- [Metasploit SMB Authentication](#) (3,562)
- [Malicious Svc Possibly Installed](#) (271)

Excelでの解析

Time	Computername	EventID	Level	Alert	Details
2021-05-03 17:58:38.774 +09:00	webiis01.offsec.lan	4624	informational	Logon Type 3 - Network	User: adminmig : Workstation: - : IP Address: 10.23.23.9 : Port: 62234 : LogonID: 0x258b9ee5
2021-05-03 17:58:38.775 +09:00	webiis01.offsec.lan	4624	informational	Logon Type 3 - Network	User: adminmig : Workstation: - : IP Address: 10.23.23.9 : Port: 62235 : LogonID: 0x258b9ef8
2021-05-03 17:58:38.775 +09:00	webiis01.offsec.lan	4624	informational	Logon Type 3 - Network	User: adminmig : Workstation: - : IP Address: 10.23.23.9 : Port: 62236 : LogonID: 0x258b9efd
2021-05-03 21:06:57.954 +09:00	win10-02.offsec.lan	1	high	Process Creation Sysmon Rule Alert	Rule: technique_id=T1059,technique_name=Command-Line Interface : Command: C:\windows\sysmon.exe
2021-05-03 21:06:57.954 +09:00	win10-02.offsec.lan	1	critical	Sticky Key Like Backdoor Usage	
2021-05-15 05:39:33.214 +09:00	fs01.offsec.lan	1102	high	Security log was cleared	User: adminmig
2021-05-19 06:18:40.607 +09:00	rootdc1.offsec.lan	150	critical	DNS Server Error Failed Loading the ServerLevelPluginDLL	
2021-05-19 06:18:40.607 +09:00	rootdc1.offsec.lan	150	high	Possible CVE-2021-1675 Print Spooler Exploitation	
2021-05-19 06:18:40.607 +09:00	rootdc1.offsec.lan	150	critical	Mimikatz Use	
2021-05-19 06:23:27.038 +09:00	rootdc1.offsec.lan	150	critical	DNS Server Error Failed Loading the ServerLevelPluginDLL	
2021-05-19 06:23:27.038 +09:00	rootdc1.offsec.lan	150	high	Possible CVE-2021-1675 Print Spooler Exploitation	
2021-05-19 06:23:27.038 +09:00	rootdc1.offsec.lan	150	critical	Mimikatz Use	
2021-05-19 06:30:17.318 +09:00	rootdc1.offsec.lan	4688	high	Possible CVE-2021-1675 Print Spooler Exploitation	
2021-05-19 06:30:17.318 +09:00	rootdc1.offsec.lan	4688	critical	Mimikatz Use	
2021-05-19 06:30:17.318 +09:00	rootdc1.offsec.lan	4688	high	Relevant Anti-Virus Event	
2021-05-19 06:33:49.548 +09:00	rootdc1.offsec.lan	770	critical	DNS Server Error Failed Loading the ServerLevelPluginDLL	
2021-05-19 06:33:49.548 +09:00	rootdc1.offsec.lan	770	high	Possible CVE-2021-1675 Print Spooler Exploitation	
2021-05-19 06:33:49.548 +09:00	rootdc1.offsec.lan	770	high	Relevant Anti-Virus Event	
2021-05-19 06:33:49.548 +09:00	rootdc1.offsec.lan	770	critical	Mimikatz Use	
2021-05-20 21:49:31.863 +09:00	fs01.offsec.lan	1102	high	Security log was cleared	User: adminmig
2021-05-20 21:49:46.875 +09:00	fs01.offsec.lan	4648	informational	Explicit Logon	Source User: FS01\$: Target User: sshd_5848 : IP Address: - : Process: C:\Program Files\OpenSSH\OpenSSH-7.4p1-1.1.0.1-1~1\sshd.exe : Port: - : LogonID: 0x3c569ed
2021-05-20 21:49:46.876 +09:00	fs01.offsec.lan	4624	low	Logon Type 5 - Service	User: sshd_5848 : Workstation: - : IP Address: - : Port: - : LogonID: 0x3c569ed
2021-05-20 21:49:46.876 +09:00	fs01.offsec.lan	4672	informational	Admin Logon	User: sshd_5848 : LogonID: 0x3c569ed
2021-05-20 21:49:52.315 +09:00	fs01.offsec.lan	4776	informational	NTLM Logon to Local Account	User: NOUSER : Workstation FS01 : Status: 0xc0000064
2021-05-20 21:49:52.315 +09:00	fs01.offsec.lan	4625	informational	Logon Failure - Username does not exist	User: NOUSER : Type: 8 : Workstation: FS01 : IP Address: - : SubStatus: 0xc0000064 : AuthP

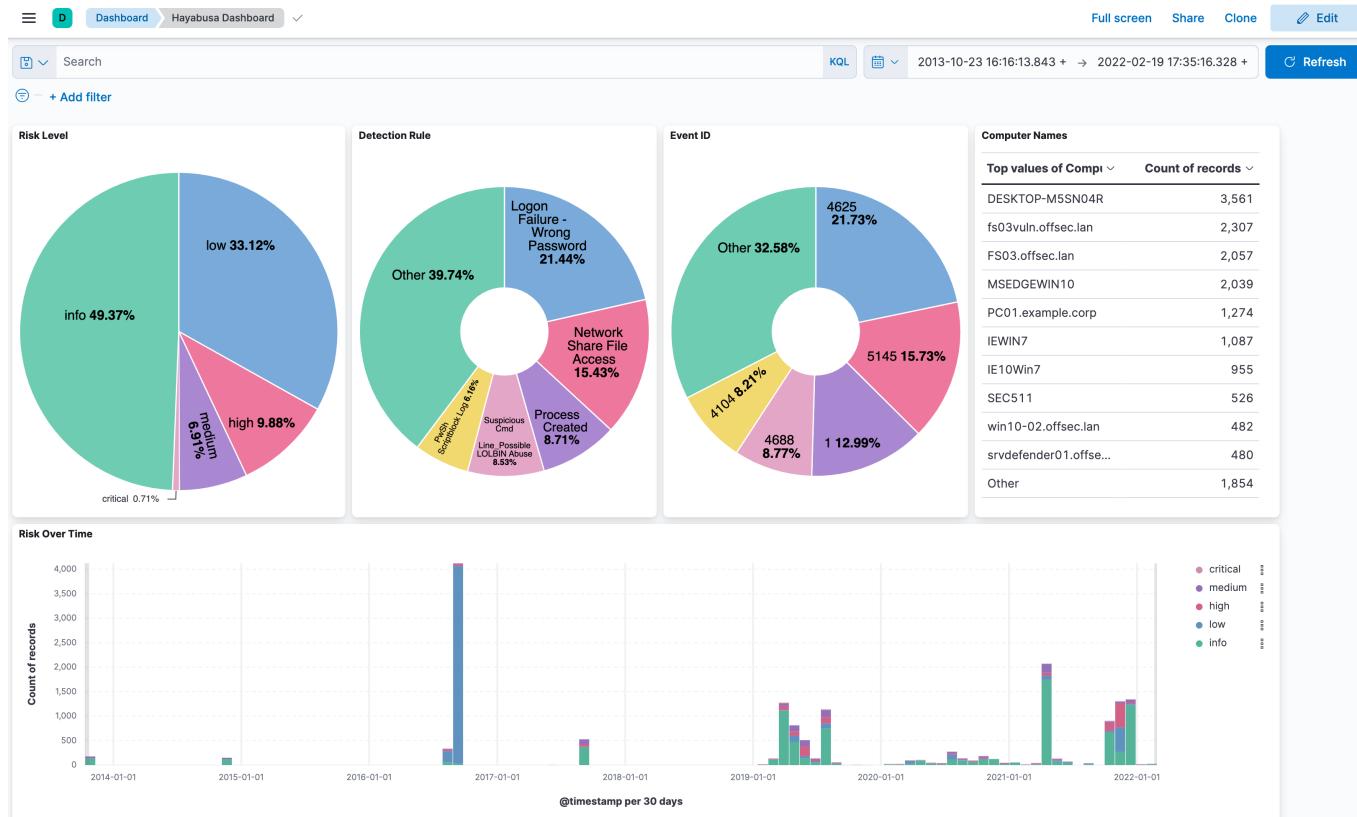
Timeline Explorerでの解析

Time	Computername	Eventid	Level	Alert	Details
2021-05-22 05:43:18.227 +09:00	fs01.offsec.lan	4648	informational	Explicit Logon	Source User: FS01\$: Target User: admmig
2021-05-22 05:43:22.562 +09:00	fs01.offsec.lan	4625	low	Logon Failure - Wrong Password	User: admmig@offsec.lan : Type: 8 : Wor
2021-05-22 05:43:49.345 +09:00	fs01.offsec.lan	4625	low	Logon Failure - Wrong Password	User: admmig@offsec.lan : Type: 8 : Wor
2021-05-22 05:43:50.131 +09:00	fs01.offsec.lan	4625	low	Logon Failure - Wrong Password	User: admmig@offsec.lan : Type: 8 : Wor
2021-05-22 05:43:50.607 +09:00	fs01.offsec.lan	4625	low	Logon Failure - Wrong Password	User: admmig@offsec.lan : Type: 8 : Wor
2021-05-22 05:43:50.866 +09:00	fs01.offsec.lan	4625	low	Logon Failure - Wrong Password	User: admmig@offsec.lan : Type: 8 : Wor
2021-05-23 06:56:57.685 +09:00	fs01.offsec.lan	1102	high	Security log was cleared	User: admmig
2021-05-23 06:57:11.842 +09:00	fs01.offsec.lan	4688	high	Relevant Anti-Virus Event	
2021-05-23 06:57:11.842 +09:00	fs01.offsec.lan	4688	critical	Mimikatz Use	
2021-05-26 22:02:27.149 +09:00	mssql01.offsec.lan	4624	informational	Logon Type 3 - Network	User: admmig : Workstation: - : IP Addr
2021-05-26 22:02:27.155 +09:00	mssql01.offsec.lan	5145	medium	DCERPC SMB Spoolss Named Pipe	
2021-05-26 22:02:27.155 +09:00	mssql01.offsec.lan	5145	critical	CVE-2021-1675 Print Spooler Exploitation IPC Access	
2021-05-26 22:02:29.726 +09:00	mssql01.offsec.lan	4624	informational	Logon Type 3 - Network	User: admmig : Workstation: - : IP Addr
2021-05-26 22:02:29.734 +09:00	mssql01.offsec.lan	5145	medium	DCERPC SMB Spoolss Named Pipe	
2021-05-26 22:02:29.734 +09:00	mssql01.offsec.lan	5145	critical	CVE-2021-1675 Print Spooler Exploitation IPC Access	
2021-05-26 22:02:34.373 +09:00	mssql01.offsec.lan	4624	informational	Logon Type 3 - Network	User: admmig : Workstation: - : IP Addr
2021-05-26 22:02:34.373 +09:00	mssql01.offsec.lan	5145	medium	DCERPC SMB Spoolss Named Pipe	
2021-05-26 22:02:34.379 +09:00	mssql01.offsec.lan	4624	informational	Logon Type 3 - Network	User: admmig : Workstation: - : IP Addr
2021-05-26 22:02:34.379 +09:00	mssql01.offsec.lan	4624	informational	Logon Type 3 - Network	User: admmig : Workstation: - : IP Addr
2021-05-26 22:02:34.380 +09:00	mssql01.offsec.lan	4624	informational	Logon Type 3 - Network	User: admmig : Workstation: - : IP Addr
2021-05-27 05:24:46.570 +09:00	rootdc1.offsec.lan	4768	medium	Possible AS-REP Roasting	Possible AS-REP Roasting
2021-05-27 05:24:46.570 +09:00	rootdc1.offsec.lan	4768	informational	Kerberos TGT was requested	User: admin-test : Service: krbtgt : IP
2021-06-01 23:06:34.542 +09:00	fs01.offsec.lan	4720	medium	Local user account created	User: WADGUtilityAccount : SID:S-1-5-21-1081258321-3780
2021-06-01 23:08:21.225 +09:00	fs01.offsec.lan	4720	medium	Local user account created	User: elie : SID:S-1-5-21-1081258321-3780
2021-06-03 21:17:56.988 +09:00	fs01.offsec.lan	1102	high	Security log was cleared	User: admmig
2021-06-03 21:18:12.941 +09:00	fs01.offsec.lan	4672	informational	Admin Logon	User: admmig : LogonID: 0x322e5b7
2021-06-03 21:18:12.942 +09:00	fs01.offsec.lan	4624	informational	Logon Type 3 - Network	User: admmig : Workstation: - : IP Addr
2021-06-04 03:34:12.672 +09:00	fs01.offsec.lan	4104	high	Windows Firewall Profile Disabled	
2021-06-04 04:17:44.873 +09:00	fs01.offsec.lan	1102	high	Security log was cleared	User: admmig

Criticalアラートのフィルタリングとコンピュータごとのグルーピング

Computername ▲					
Line	Tag	Time	Eventid	Level	Alert
▼ =	■	UTC	UTC	= critical	UTC
▶ Computername: 01566s-win16-ir.threebeesco.com (Count: 1)					
▶ Computername: alice.insecurebank.local (Count: 3)					
△ Computername: DC1.insecurebank.local (Count: 18)					
5540	■	2019-03-26 06:28:45.026 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5539	■	2019-03-26 06:28:45.026 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5538	■	2019-03-26 06:28:45.026 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5537	■	2019-03-26 06:28:45.026 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5536	■	2019-03-26 06:28:45.025 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5535	■	2019-03-26 06:28:45.025 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5534	■	2019-03-26 06:28:45.025 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5533	■	2019-03-26 06:28:45.025 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5532	■	2019-03-26 06:28:45.025 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5531	■	2019-03-26 06:28:45.024 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5530	■	2019-03-26 06:28:45.024 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5529	■	2019-03-26 06:28:45.024 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5528	■	2019-03-26 06:28:45.023 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5527	■	2019-03-26 06:28:45.023 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5526	■	2019-03-26 06:28:45.023 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5525	■	2019-03-26 06:28:45.023 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5524	■	2019-03-26 06:28:45.022 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
5523	■	2019-03-26 06:28:45.022 +09:00	5136	critical	Powerview Add-DomainObjectAcl DCSync AD Extend Right
▶ Computername: DESKTOP-PIU87N6 (Count: 1)					

Elastic Stackダッシュボードでの解析



Top 10 Alerts							Top 10 Critical Alerts							Top 10 High Alerts						
Top values of RuleTitle		info	Cour	low	Coun	high	Cou	medium	Cou	critical	Cou	Count of records	Top values of RuleTitle	Count of records	Top values of RuleTitle	Count of records				
Network Share File Access		2,564	-	-	-	-	-	-	-	-	-	Mimikatz Use	33	Malicious Service Possibly Inst...	271					
Process Created		1,447	-	-	-	-	-	-	-	-	-	Powerview Add-DomainObjectAcl...	22	Suspicious Service Installed	257					
PwSh Scriptblock Log		1,024	-	-	-	-	-	-	-	-	-	Sticky Key Like Backdoor Usage	8	System Log File Cleared	97					
PwSh Pipeline Execution		680	-	-	-	-	-	-	-	-	-	Active Directory Replication from ...	6	Suspicious Remote Thread Crea...	94					
Network Share Access		433	-	-	-	-	-	-	-	-	-	EfsPotato Named Pipe	6	Accessing WinAPI in PowerShe...	93					
Other		2,058	223	831	594	4:	4:	4:	4:	4:	4:	WannaCry Ransomware	4	Relevant Anti-Virus Event	71					
Logon Failure - Wrong Password		-	3,564	-	-	-	-	-	-	-	-	CobaltStrike Service Installations	3	Security Log Cleared	66					
Suspicious Cmd Line_Possible LOLBIN ...		-	1,418	-	-	-	-	-	-	-	-	DNS Server Error Failed Loading t...	3	Process Created_Sysmon Alert	60					
Process Access		-	154	-	-	-	-	-	-	-	-	Dumpert Process Dumper	3	Disabling Windows Event Audit...	42					
Image Loaded_Sysmon Alert		-	108	-	-	-	-	-	-	-	-	LSASS Access from Non System ...	3	Malicious PowerShell Keywords	30					
Process Start From Suspicious Folder		-	39	-	-	-	-	-	-	-	-	Other	27	Other	562					

Hayabusa Discover: A table showing log entries with 16622 documents.

Time	Computer	EventID	Level	MitreAttack	RuleTitle	Details
2022-02-19 17:35:16.328 +00:00	DESKTOP-TTEQ6PR	7	info	Persis Evas Pr ivEsc	Windows Spooler Service Suspicious Binary Load	-
2022-02-19 17:35:16.301 +00:00	DESKTOP-TTEQ6PR	11	info	-	File Created	Path: C:\Windows\System32\spool\drivers\x64\4\Test.dll Process: C:\Users\win10\Desktop\SpoolPool-main\SpoolPool.exe PID: 1232 PUUID: 0BDAA6306-2A54-6211-0B01-000000001000
2022-02-19 17:35:16.301 +00:00	DESKTOP-TTEQ6PR	11	medium	-	Rename Common File to DLL	-
2022-02-19 17:35:16.207 +00:00	DESKTOP-TTEQ6PR	1	info	-	Process Created	Cmd: "C:\Users\win10\Desktop\SpoolPool-main\SpoolPool.exe" -d11 C:\ProgramData\Test.dll Process: C:\Users\win10\Desktop\SpoolPool-main\SpoolPool.exe User: DESKTOP-TTEQ6PR\win10 Parent Cmd: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -noexit -command Set-Location -LiteralPath C:\Users\win10\Desktop\SpoolPool-main LID: 0x277ef PID: 1232 PUUID: 0BDAA6306-2A54-6211-0B01-000000001000
2022-02-19 17:35:16.207 +00:00	DESKTOP-TTEQ6PR	1	low	Exec	Process Start From Suspicious Folder	-
2022-02-16 19:37:20.934 +00:00	01566s-win16-ir.t hreebeesco.com	5145	info	Collect	Network Share File Access	User: samir Share Name: *\C\$ Share Path: \?\C:\ Path: Users\PSECURITY IP Addr: 172.16.66.36 LID: 0x567758

Timesketchでの解析

2019-05-08T02:10:43	<input type="checkbox"/> Active Directory Replication from Non Machine Account	User: Administrator ObjSvr: DS ObjName: %{c6faf700-bfe4-452a-a766-424f84c29583} OpType: Object Access HID: 0x0 LID: 0x40c6511	4662	DC1.insecurebank.local	Sec	T1003.006
2019-05-08T02:10:43	<input type="checkbox"/> Active Directory Replication from Non Machine Account	User: Administrator ObjSvr: DS ObjName: %{c6faf700-bfe4-452a-a766-424f84c29583} OpType: Object Access HID: 0x0 LID: 0x40c6511	4662	DC1.insecurebank.local	Sec	T1003.006
2019-05-08T02:10:43	<input type="checkbox"/> Active Directory Replication from Non Machine Account	User: Administrator ObjSvr: DS ObjName: %{c6faf700-bfe4-452a-a766-424f84c29583} OpType: Object Access HID: 0x0 LID: 0x40c6511	4662	DC1.insecurebank.local	Sec	T1003.006
4 days						
2019-05-12T12:52:43	<input type="checkbox"/> Meterpreter or Cobalt Strike Getsystem Service Installation	Svc: WinPwnage Path: %COMSPEC% /c ping -n 1 127.0.0.1 >nul && echo 'WinPwnage' > \\.\pipe\WinPwnagePipe Acct: LocalSystem StartType: demand start	7045	IEWIN7	Sys	T1134.001 : T1134.002
39 days						
2019-06-21T07:35:37	<input type="checkbox"/> Dumpert Process Dumper	Path: C:\Windows\Temp\dumpert.dmp Process: C:\Users\administrator\Desktop\x64\O utfleck-Dumpert.exe PID: 3572 PGUID: ECAD0485-88C9-5D0C-0000-0010348C1D00	11	alice.insecurebank.local	Sysmon	T1003.001

タイムラインのサンプル結果

CSVのタイムラインをExcelやTimeline Explorerで分析する方法は[こちら](#)で紹介しています。

CSVのタイムラインをElastic Stackにインポートする方法は[こちら](#)で紹介しています。

CSVのタイムラインをTimesketchにインポートする方法は[こちら](#)で紹介しています。

特徴 & 機能

- クロスプラットフォーム対応: Windows, Linux, macOS。
- Rustで開発され、メモリセーフでハヤブサよりも高速です！
- マルチスレッド対応により、最大5倍のスピードアップを実現。
- フォレンジック調査やインシデントレスポンスのために、分析しやすいCSVタイムラインを作成します。
- 読みやすい/作成/編集可能なYMLベースのHayabusaルールで作成されたIoCシグネチャに基づくスレット。
- SigmaルールをHayabusaルールに変換するためのSigmaルールのサポートがされています。
- 現在、他の類似ツールに比べ最も多くのSigmaルールをサポートしており、カウントルール、新しい機能の`|equalsfield`や`|endswithfield`等にも対応しています。
- イベントログの統計。(どのような種類のイベントがあるのかを把握し、ログ設定のチューニングに有効です。)
- 不良ルールやノイズの多いルールを除外するルールチューニング設定が可能です。
- MITRE ATT&CKとのマッピング (CSVの出力ファイルのみ)。
- ルールレベルのチューニング。
- イベントログから不審なユーザやファイルを素早く特定するためのピボットキーワードの一覧作成。

- 詳細な調査のために全フィールド情報の出力。
- 成功と失敗したユーザログオンの要約。
- [Velociraptor](#)と組み合わせた企業向けの広範囲なすべてのエンドポイントに対するスレットハンティングとDFIR。
- CSV、JSON、JSONL形式とHTML結果サマリの出力。
- 毎日のSigmaルール更新。

ダウンロード

[Releases](#)ページからHayabusaの安定したバージョンでコンパイルされたバイナリが含まれている最新版もしくはソースコードをダウンロードできます。

Gitクローン

以下の`git clone`コマンドでレポジトリをダウンロードし、ソースコードからコンパイルして使用することも可能です：

```
git clone https://github.com/Yamato-Security/hayabusa.git --recursive
```

注意： mainブランチは開発中のバージョンです。まだ正式にリリースされていない新機能が使えるかもしれないが、バグがある可能性があるので、テスト版だと思って下さい。

* `--recursive`をつけ忘れた場合、サブモジュールとして管理されている`rules`フォルダ内のファイルはダウンロードされません。

`git pull --recurse-submodules`コマンド、もしくは以下のコマンドで`rules`フォルダを同期し、Hayabusaの最新のルールを更新することができます：

```
hayabusa-2.0.0-win-x64.exe update-rules
```

アップデートが失敗した場合は、`rules`フォルダの名前を変更してから、もう一回アップデートしてみて下さい。

注意: アップデートを実行する際に `rules` フォルダは `hayabusa-rules` レポジトリの最新のルールとコンフィグファイルに置き換えられます 既存ファイルへの修正はすべて上書きされますので、アップデート実行前に編集したファイルのバックアップをおすすめします。もし、`level-tuning`を行っているのであれば、アップデート後にルールファイルの再調整をしてください `rules` フォルダ内に新しく追加したルールは、アップデート時に上書きもしくは削除は行われません。

アドバンス: ソースコードからのコンパイル（任意）

Rustがインストールされている場合、以下のコマンドでソースコードからコンパイルすることができます：

注意: hayabusaをコンパイルするためにはRust(rustc)が1.65.0以上であることが必要です。

```
cargo build --release
```

以下のコマンドで定期的にRustをアップデートしてください：

```
rustup update stable
```

コンパイルされたバイナリはtarget/releaseフォルダ配下で作成されます。

Rustパッケージの更新

コンパイル前に最新のRust crateにアップデートすることで、最新のライブラリを利用することができます：

```
cargo update
```

※ アップデート後、何か不具合がありましたらお知らせください。

32ビットWindowsバイナリのクロスコンパイル

以下のコマンドで64ビットのWindows端末で32ビットのバイナリをクロスコンパイルできます：

```
rustup install stable-i686-pc-windows-msvc
rustup target add i686-pc-windows-msvc
rustup run stable-i686-pc-windows-msvc cargo build --release
```

注意: Rust の新しい安定版が出たときには必ずrustup install stable-i686-pc-windows-msvcを実行してください。rustup update stableはクロスコンパイル用のコンパイラを更新しないので、ビルドエラーが発生することがあります。

macOSでのコンパイルの注意点

opensslについてのコンパイルエラーが表示される場合は、[Homebrew](#)をインストールしてから、以下のパッケージをインストールする必要があります：

```
brew install pkg-config
brew install openssl
```

Linuxでのコンパイルの注意点

opensslについてのコンパイルエラーが表示される場合は、以下のパッケージをインストールする必要があります。

Ubuntu系のディストロ:

```
sudo apt install libssl-dev
```

Fedora系のディストロ:

```
sudo yum install openssl-devel
```

LinuxのMUSLバイナリのクロスコンパイル

まず、Linux OSでターゲットをインストールします。

```
rustup install stable-x86_64-unknown-linux-musl
rustup target add x86_64-unknown-linux-musl
```

以下のようにコンパイルします:

```
cargo build --release --target=x86_64-unknown-linux-musl
```

注意: Rust の新しい安定版が出たときには必ず `rustup install stable-x86_64-unknown-linux-musl` を実行してください。 `rustup update stable` はクロスコンパイル用のコンパイラを更新しないので、ビルドエラーが発生することがあります。

MUSLバイナリは `./target/x86_64-unknown-linux-musl/release/` ディレクトリ配下に作成されます。 MUSLバイナリはGNUバイナリより約15%遅いですが、より多くのLinuxバージョンとディストロで実行できます。

Hayabusaの実行

注意: アンチウィルス/EDRの誤検知と遅い初回実行

Hayabusa実行する際や、`.yml` ルールのダウンロードや実行時にルール内で `detection` に不審な PowerShell コマンドや `mimikatz` のようなキーワードが書かれている際に、アンチウィルスや EDR にブロックされる可能性があります。 誤検知のため、セキュリティ対策の製品が Hayabusa を許可するように設定する必要があります。 マルウェア感染が心配であれば、ソースコードを確認した上で、自分でバイナリをコンパイルして下さい。

Windows PC起動後の初回実行時に時間がかかる場合があります。これは Windows Defender のリアルタイムスキャンが行われていることが原因です。 リアルタイムスキャンを無効にするか Hayabusa のディレクトリをアンチウィルススキャンから除外することでこの現象は解消しますが、設定を変える前にセキュリティリスクを十分ご考慮ください。

Windows

コマンドプロンプトや Windows Terminal から 32 ビットもしくは 64 ビットの Windows バイナリを Hayabusa のルートディレクトリから実行します。

例: **hayabusa-2.0.0-win-x64.exe**

Linux

まず、バイナリに実行権限を与える必要があります。

```
chmod +x ./hayabusa-2.0.0-lin-gnu
```

次に、Hayabusaのルートディレクトリから実行します：

```
./hayabusa-2.0.0-lin-gnu
```

macOS

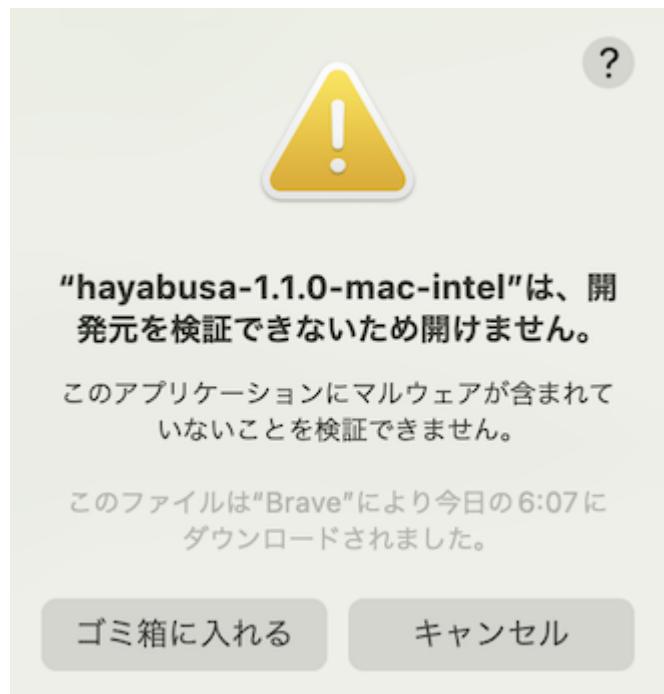
まず、ターミナルやiTerm2からバイナリに実行権限を与える必要があります。

```
chmod +x ./hayabusa-2.0.0-mac-intel
```

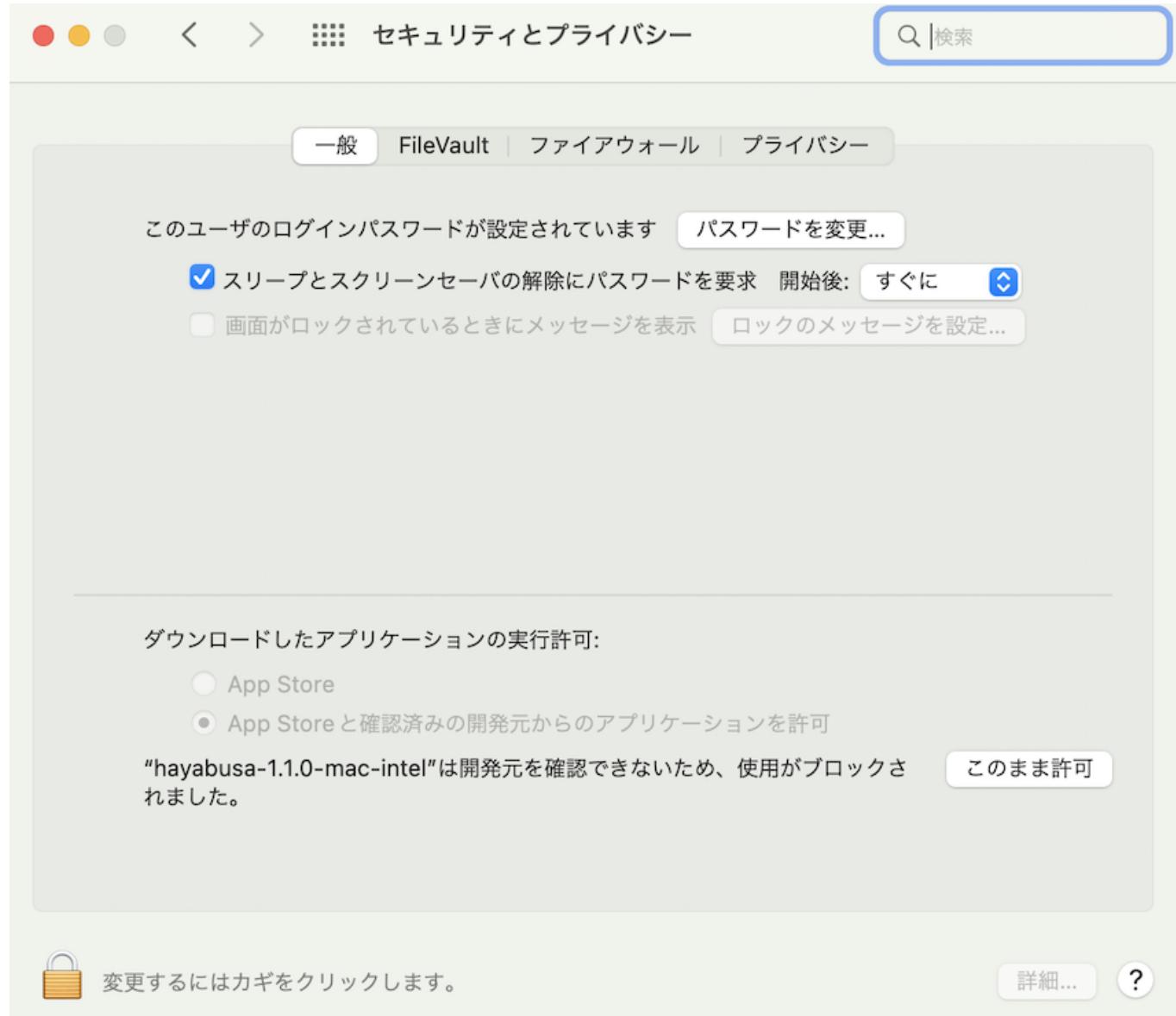
次に、Hayabusaのルートディレクトリから実行してみてください：

```
./hayabusa-2.0.0-mac-intel
```

macOSの最新版では、以下のセキュリティ警告が出る可能性があります：



macOSの環境設定から「セキュリティとプライバシー」を開き、「一般」タブから「このまま許可」ボタンをクリックしてください。



その後、ターミナルからもう一回実行してみてください：

```
./hayabusa-2.0.0-mac-intel
```

以下の警告が出るので、「開く」をクリックしてください。



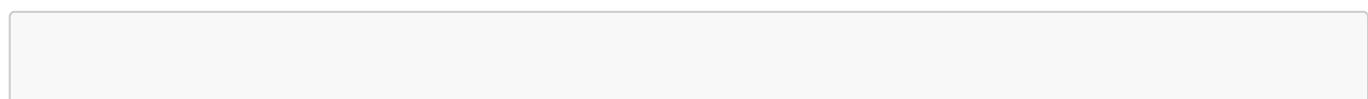
これで実行できるようになります。

主なコマンド

- `csv-timeline`: CSV形式のタイムラインを出力する。
- `json-timeline`: JSON/JSONL形式のタイムラインを出力する。
- `logon-summary`: ログオンイベントのサマリを出力する。
- `metrics`: イベントIDに基づくイベントの合計と割合の集計を出力する。
- `pivot-keywords-list`: ピボットする不審なキーワードのリストを作成する。
- `update-rules`: GitHubの[hayabusa-rules](#)リポジトリにある最新のルールに同期させる。
- `level-tuning`: アラート `level` のカスタムチューニング。
- `list-profiles`: 出力プロファイルの一覧表示。
- `set-default-profile`: デフォルトプロファイルを変更する。

使用方法

デフォルトのヘルプメニュー



Usage:

```
hayabusa.exe help <COMMAND>
hayabusa.exe <COMMAND> [OPTIONS]
```

Commands:

csv-timeline	CSV形式のタイムラインを出力
json-timeline	JSON/JSONNL形式のタイムラインを出力
logon-summary	ログオンイベントのサマリを出力
metrics	イベントIDに基づくイベントの合計と割合の集計を出力
pivot-keywords-list	ピボットキーワードの一覧作成
update-rules	rulesフォルダをhayabusa-rulesのgithubリポジトリの最新版に更新する

level-tuning	ルールlevelのチューニング (デフォルト: ./rules/config/level_tuning.txt)
set-default-profile	デフォルトの出力コンフィグを設定する
list-contributors	コントリビュータの一覧表示
list-profiles	出力プロファイルの一覧表示
help	コマンドに付随するオプションのヘルプを表示する

Options:

--no-color	カラーで出力しない
-q, --quiet	Quietモード: 起動バナーを表示しない

csv-timelineコマンド

Options:

-t, --thread-number <NUMBER>	スレッド数 (デフォルト: パフォーマンスに最適な数値)
-Q, --quiet-errors	Quiet errorsモード: エラーログを保存しない
-c, --rules-config <DIRECTORY>	ルールフォルダのコンフィグディレクトリ (デフォルト: ./rules/config)
-v, --verbose	詳細な情報を出力する

Input:

-d, --directory <DIRECTORY>	.evtxファイルを持つディレクトリのパス
-f, --file <FILE>	1つの.evtxファイルに対して解析を行う
-l, --live-analysis	ローカル端末のC:\Windows\System32\winevt\Logs

フォルダを解析する

Advanced:

--target-file-ext <EVTX_FILE_EXT>	evtx以外の拡張子を解析対象に追加する。
(例1: evtx_data 例2 : evtx1,evtx2)	
-r, --rules <DIRECTORY/FILE>	ルールファイルまたはルールファイルを持つディレクトリ (デフォルト: ./rules)

Output:

-P, --profile <PROFILE>	利用する出力プロファイル名を指定する
-o, --output <FILE>	タイムラインを保存する (csv-timelineの例: results.csv, json-timelineの例: results.json)
-H, --html-report <FILE>	HTML形式で詳細な結果を出力する (例: results.html)

Filtering:

--enable-deprecated-rules	Deprecatedルールを有効にする
--exclude-status <STATUS>	読み込み対象外とするルール内でのステータス (ex: experimental) (ex: stable, test)
-m, --min-level <LEVEL>	結果出力をするルールの最低レベル (デフォルト: informational)
-n, --enable-noisy-rules	Noisyルールを有効にする
--timeline-end <DATE>	解析対象とするイベントログの終了時刻 (例: "2022-02-22 23:59:59 +09:00")
--timeline-start <DATE>	解析対象とするイベントログの開始時刻 (例: "2020-02-22 00:00:00 +09:00")
-e, --eid-filter	イベントIDによるフィルタリングを行う (コンフィグファイル: ./rules/config/target_event_IDs.txt)

Time Format:

--European-time	ヨーロッパ形式で日付と時刻を出力する (例: 22-02-2022 22:00:00.123 +02:00)
--ISO-8601	ISO-8601形式で日付と時刻を出力する (ex: 2022-02-22T10:10:10.1234567Z) (いつもUTC)
--RFC-2822	RFC 2822形式で日付と時刻を出力する (例: Fri, 22 Feb 2022 22:00:00 -0600)
--RFC-3339	RFC 3339形式で日付と時刻を出力する (例: 2022-02-22 22:00:00.123456-06:00)
--US-military-time	24時間制(ミリタリータイム)のアメリカ形式で日付と時刻を出力する (例: 02-22-2022 22:00:00.123 -06:00)
--US-time	アメリカ形式で日付と時刻を出力する (例: 02-22-2022 10:00:00.123 PM -06:00)
-U, --UTC	UTC形式で日付と時刻を出力する (デフォルト: 現地時間)

Display Settings:

-T, --visualize-timeline	イベント頻度タイムラインを出力する
--no-summary	結果概要を出力しない

json-timelineコマンド

Usage: json-timeline <INPUT> [OPTIONS]

Options:

-t, --thread-number <NUMBER>	スレッド数 (デフォルト: パフォーマンスに最適な数値)
-Q, --quiet-errors	Quiet errorsモード: エラーログを保存しない
-c, --rules-config <DIRECTORY>	ルールフォルダのコンフィグディレクトリ (デフォルト: ./rules/config)
-v, --verbose	詳細な情報を出力する

Input:

-d, --directory <DIRECTORY>	.evtxファイルを持つディレクトリのパス
-f, --file <FILE>	1つの.evtxファイルに対して解析を行う
-l, --live-analysis	ローカル端末のC:\Windows\System32\winevt\Logs

フォルダを解析する

Advanced:

--target-file-ext <EVTX_FILE_EXT> evtx以外の拡張子を解析対象に追加する。
(例1: evtx_data 例2 : evtx1,evtx2)

-r, --rules <DIRECTORY/FILE> ルールファイルまたはルールファイルを持つ
ディレクトリ (デフォルト: ./rules)

Output:

-p, --profile <PROFILE> 利用する出力プロファイル名を指定する
-o, --output <FILE> タイムラインを保存する (csv-timelineの例:
result.csv, json-timelineの例: result.json)

-H, --html-report <FILE> HTML形式で詳細な結果を出力する (例: results.html)
-J, --jsonl JSONNL形式でタイムラインを保存する (ex: -J -o
results.jsonl)

Filtering:

--enable-deprecated-rules	Deprecatedルールを有効にする
--exclude-status <STATUS>	読み込み対象外とするルール内でのステータス (ex: experimental) (ex: stable,test)
-m, --min-level <LEVEL>	結果出力をするルールの最低レベル (デフォルト: informational)
-n, --enable-noisy-rules	Noisyルールを有効にする
--timeline-end <DATE> 02-22 23:59:59 +09:00")	解析対象とするイベントログの終了時刻 (例: "2022-
--timeline-start <DATE> 02-22 00:00:00 +09:00")	解析対象とするイベントログの開始時刻 (例: "2020-
-e, --eid-filter	イベントIDによるフィルタリングを行う (コンフィグフ イル: ./rules/config/target_event_IDs.txt)

Time Format:

--European-time	ヨーロッパ形式で日付と時刻を出力する (例: 22-02-2022 22:00:00.123 +02:00)
--ISO-8601	ISO-8601形式で日付と時刻を出力する (ex: 2022-02- 22T10:10:10.1234567Z) (いつもUTC)
--RFC-2822	RFC 2822形式で日付と時刻を出力する (例: Fri, 22 Feb 2022 22:00:00 -0600)
--RFC-3339	RFC 3339形式で日付と時刻を出力する (例: 2022-02-22 22:00:00.123456-06:00)
--US-military-time	24時間制(ミリタリータイム)のアメリカ形式で日付と時刻を出力 する (例: 02-22-2022 22:00:00.123 -06:00)
--US-time	アメリカ形式で日付と時刻を出力する (例: 02-22-2022 10:00:00.123 PM -06:00)
-U, --UTC	UTC形式で日付と時刻を出力する (デフォルト: 現地時間)

Display Settings:

-T, --visualize-timeline イベント頻度タイムラインを出力する
--no-summary 結果概要を出力しない

logon-summaryコマンド

Usage: logon-summary <INPUT> [OPTIONS]

Options:

- t, --thread-number <NUMBER> スレッド数 (デフォルト: パフォーマンスに最適な数値)
- Q, --quiet-errors Quiet errorsモード: エラーログを保存しない
- c, --rules-config <DIRECTORY> ルールフォルダのコンフィグディレクトリ (デフォルト: ./rules/config)
- v, --verbose 詳細な情報を出力する

Input:

- d, --directory <DIRECTORY> .evtxファイルを持つディレクトリのパス
- f, --file <FILE> 1つの.evtxファイルに対して解析を行う
- l, --live-analysis ローカル端末のC:\Windows\System32\winevt\Logs
フォルダを解析する

Advanced:

- target-file-ext <EVTX_FILE_EXT> evtx以外の拡張子を解析対象に追加する
(例1: evtx_data 例2: evtx1, evtx2)

Output:

- o, --output <FILE> ログオンサマリをCSV形式で保存する (例: logon-summary.csv)

metricsコマンド

Usage: metrics <INPUT> [OPTIONS]

Options:

- t, --thread-number <NUMBER> スレッド数 (デフォルト: パフォーマンスに最適な数値)
- Q, --quiet-errors Quiet errorsモード: エラーログを保存しない
- c, --rules-config <DIRECTORY> ルールフォルダのコンフィグディレクトリ (デフォルト: ./rules/config)
- v, --verbose 詳細な情報を出力する

Input:

- d, --directory <DIRECTORY> .evtxファイルを持つディレクトリのパス
- f, --file <FILE> 1つの.evtxファイルに対して解析を行う
- l, --live-analysis ローカル端末のC:\Windows\System32\winevt\Logs
フォルダを解析する

Advanced:

- target-file-ext <EVTX_FILE_EXT> evtx以外の拡張子を解析対象に追加する。
(例1: evtx_data 例2: evtx1, evtx2)

Output:

- o, --output <FILE> 成功と失敗したログオン情報の要約を出力する (例: metrics.csv)

pivot-keywords-listコマンド

Usage: pivot-keywords-list <INPUT> [OPTIONS]

Options:

- t, --thread-number <NUMBER> スレッド数（デフォルト：パフォーマンスに最適な数値）
- Q, --quiet-errors Quiet errorsモード：エラーログを保存しない
- c, --rules-config <DIRECTORY> ルールフォルダのコンフィグディレクトリ（デフォルト: ./rules/config）
- v, --verbose 詳細な情報を出力する

Input:

- d, --directory <DIRECTORY> .evtxファイルを持つディレクトリのパス
- f, --file <FILE> 1つの.evtxファイルに対して解析を行う
- l, --live-analysis ローカル端末のC:\Windows\System32\winevt\Logs
フォルダを解析する

Advanced:

- target-file-ext <EVTX_FILE_EXT> evtx以外の拡張子を解析対象に追加する。
(例1: evtx_data 例2: evtx1,evtx2)

Output:

- o, --output <FILE> 成功と失敗したログオン情報の要約を出力する（例: pivot-keywords.txt）

Filtering:

- enable-deprecated-rules Deprecatedルールを有効にする
- exclude-status <STATUS> 読み込み対象外とするルール内でのステータス（ex: experimental）(ex: stable,test)
- m, --min-level <LEVEL> 結果出力をするルールの最低レベル（デフォルト: informational）
- n, --enable-noisy-rules Noisyルールを有効にする
- timeline-end <DATE> 解析対象とするイベントログの終了時刻（例: "2022-02-22 23:59:59 +09:00"）
- timeline-start <DATE> 解析対象とするイベントログの開始時刻（例: "2020-02-22 00:00:00 +09:00"）
- e, --eid-filter イベントIDによるフィルタリングを行う（コンフィグファイル: ./rules/config/target_event_IDs.txt）

update-rulesコマンド

Usage: update-rules [OPTIONS]

Options:

- no-color カラーで出力しない
- q, --quiet Quietモード：起動バナーを表示しない

Advanced:

```
-r, --rules <DIRECTORY/FILE> ルールファイルまたはルールファイルを持つディレクトリ (デフォルト: ./rules)
```

level-tuningコマンド

```
Usage: level-tuning [OPTIONS]
```

Options:

```
-f, --file <FILE> ルールlevelのチューニング (デフォルト:  
./rules/config/level_tuning.txt)  
--no-color カラーで出力しない  
-q, --quiet Quietモード: 起動バナーを表示しない
```

set-default-profileコマンド

```
Usage: set-default-profile [OPTIONS]
```

Options:

```
-p, --profile <PROFILE> 利用する出力プロファイル名を指定する  
--no-color カラーで出力しない  
-q, --quiet Quietモード: 起動バナーを表示しない
```

list-profilesコマンド

```
Usage: list-profiles [OPTIONS]
```

Options:

```
--no-color カラーで出力しない  
-q, --quiet Quietモード: 起動バナーを表示しない
```

使用例

- 1つのWindowsイベントログファイルに対してHayabusaを実行する:

```
hayabusa-2.0.0-win-x64.exe csv-timeline -f eventlog.evtx
```

- verboseプロファイルで複数のWindowsイベントログファイルのあるsample-evtxディレクトリに対して、Hayabusaを実行する:

```
hayabusa-2.0.0-win-x64.exe csv-timeline -d .\hayabusa-sample-evtx -P  
verbose
```

- 全てのフィールド情報も含めて1つのCSVファイルにエクスポートして、Excel、Timeline Explorer、Elastic Stack等でさらに分析することができる(注意: **super-verbose**プロファイルを使すると、出力するファイルのサイズがとても大きくなる!):

```
hayabusa-2.0.0-win-x64.exe csv-timeline -d .\hayabusa-sample-evtx -o results.csv -P super-verbose
```

- タイムラインをJSON形式で保存する:

```
hayabusa-2.0.0-win-x64.exe json-timeline -d .\hayabusa-sample-evtx -o results.json
```

- Hayabusaルールのみを実行する(デフォルトでは-r .\rulesにあるすべてのルールが利用される):

```
hayabusa-2.0.0-win-x64.exe csv-timeline -d .\hayabusa-sample-evtx -r .\rules\hayabusa -o results.csv
```

- Windowsでデフォルトで有効になっているログに対してのみ、Hayabusaルールを実行する:

```
hayabusa-2.0.0-win-x64.exe csv-timeline -d .\hayabusa-sample-evtx -r .\rules\hayabusa\builtin -o results.csv
```

- Sysmonログに対してのみHayabusaルールを実行する:

```
hayabusa-2.0.0-win-x64.exe csv-timeline -d .\hayabusa-sample-evtx -r .\rules\hayabusa\sysmon -o results.csv
```

- Sigmaルールのみを実行する:

```
hayabusa-2.0.0-win-x64.exe csv-timeline -d .\hayabusa-sample-evtx -r .\rules\sigma -o results.csv
```

- 廃棄(deprecated)されたルール(statusがdeprecatedになっているルール)とノイジールール(.\\rules\\config\\noisy_rules.txtにルールIDが書かれているルール)を有効にする:

```
hayabusa-2.0.0-win-x64.exe csv-timeline -d .\hayabusa-sample-evtx --enable-noisy-rules --enable-deprecated-rules -o results.csv
```

- ログオン情報を分析するルールのみを実行し、UTCタイムゾーンで出力する:

```
hayabusa-2.0.0-win-x64.exe csv-timeline -d .\hayabusa-sample-evtx -r .\rules\hayabusa\builtin\Security\LogonLogoff\Logon -U -o results.csv
```

- 起動中のWindows端末上で実行し（Administrator権限が必要）、アラート（悪意のある可能性のある動作）のみを検知する:

```
hayabusa-2.0.0-win-x64.exe csv-timeline -l -m low
```

- criticalレベルのアラートからピボットキーワードの一覧を作成する（結果は結果毎にkeywords-IpAddress.txtやkeywords-Users.txt等に出力される）:

```
hayabusa-2.0.0-win-x64.exe pivot-keywords-list -l -m critical -o keywords
```

- イベントIDの統計情報を出力する:

```
hayabusa-2.0.0-win-x64.exe metrics -f Security.evtx
```

- ログオンサマリを出力する:

```
hayabusa-2.0.0-win-x64.exe logon-summary -f Security.evtx
```

- 詳細なメッセージを出力する（処理に時間がかかるファイル、パースエラー等を特定するのに便利）:

```
hayabusa-2.0.0-win-x64.exe csv-timeline -d .\hayabusa-sample-evtx -v
```

- Verbose出力の例:

```
Checking target evtx FilePath: "./hayabusa-sample-evtx/YamatoSecurity/T1027.004_0bfuscated Files or Information\u{a0}Compile After Delivery/sysmon.evtx"  
1 / 509 [>-----  
-----  
-] 0.20 % 1s  
Checking target evtx FilePath: "./hayabusa-sample-evtx/YamatoSecurity/T1558.004_Steal or Forge Kerberos Tickets AS-REP Roasting/Security.evtx"  
2 / 509 [>-----
```

```

-----]
-] 0.39 % 1s
Checking target evtx FilePath: "./hayabusa-sample-
evtx/YamatoSecurity/T1558.003_Steal or Forge Kerberos
Tickets\u{a0}Kerberoasting/Security.evtx"
3 / 509 [=-----]

-----]
-] 0.59 % 1s
Checking target evtx FilePath: "./hayabusa-sample-
evtx/YamatoSecurity/T1197_BITS Jobs/Windows-BitsClient.evtx"
4 / 509 [=-----]

-----]
-] 0.79 % 1s
Checking target evtx FilePath: "./hayabusa-sample-
evtx/YamatoSecurity/T1218.004_Signed Binary Proxy
Execution\u{a0}InstallUtil/sysmon.evtx"
5 / 509 [=-----]

-----]
-] 0.98 % 1s

```

- 結果を[Timesketch](#)にインポートできるCSV形式に保存する:

```
hayabusa-2.0.0-win-x64.exe csv-timeline -d ../hayabusa-sample-evtx --RFC-3339 -o timesketch-import.csv -P timesketch -U
```

- エラーログの出力をさせないようにする: デフォルトでは、Hayabusaはエラーメッセージをエラーログに保存します。エラーメッセージを保存したくない場合は、[-Q](#)を追加してください。

ピボットキーワードの作成

[pivot-keywords-list](#)コマンドを使うことで不審なユーザやホスト名、プロセスなどを一覧で出力することができます。イベントログから素早く特定することができます。ピボットキーワードのカスタマイズは[./config/pivot_keywords.txt](#)を変更することで行うことができます。以下はデフォルトの設定になります:

```

Users.SubjectUserName
Users.TargetUserName
Users.User
Logon IDs.SubjectLogonId
Logon IDs.TargetLogonId
Workstation Names.WorkstationName
Ip Addresses.IpAddress
Processes.Image

```

形式は[KeywordName.FieldName](#)となっています。例えばデフォルトの設定では、[Users](#)というリストは検知したイベントから[SubjectUserName](#)、[TargetUserName](#)、[User](#)のフィールドの値が一覧として出力されます。hayabusaのデフォルトでは検知したすべてのイベントから結果を出力するため、[pivot-keyword-list](#)コマンド

を使うときには `-m` もしくは `--min-level` オプションを併せて使って検知するイベントのレベルを指定することをおすすめします。まず `-m critical` を指定して、最も高い `critical` レベルのアラートのみを対象として、レベルを必要に応じて下げていくとよいでしょう。結果に正常なイベントにもある共通のキーワードが入っている可能性が高いため、手動で結果を確認してから、不審なイベントにありそうなキーワードリストを 1 つのファイルに保存し、`grep -f keywords.txt timeline.csv` 等のコマンドで不審なアクティビティに絞ったタイムラインを作成することができます。

ログオン情報の要約

`logon-summary` コマンドを使うことでログオン情報の要約(ユーザ名、ログイン成功数、ログイン失敗数)の画面出力ができます。単体の `evtx` ファイルを解析したい場合は `-f` オプションを利用してください。複数の `evtx` ファイルを対象としたい場合は `-d` オプションを合わせて使うことで `evtx` ファイルごとのログイン情報の要約を出力できます。

サンプル `evtx` ファイルで Hayabusa をテストする

Hayabusa をテストしたり、新しいルールを作成したりするためのサンプル `evtx` ファイルをいくつか提供しています:
<https://github.com/Yamato-Security/Hayabusa-sample-evtx>

以下のコマンドで、サンプルの `evtx` ファイルを新しいサブディレクトリ `hayabusa-sample-evtx` にダウンロードすることができます:

```
git clone https://github.com/Yamato-Security/hayabusa-sample-evtx.git
```

Hayabusa の出力

プロファイル

Hayabusa の `config/profiles.yaml` 設定ファイルでは、5 つのプロファイルが定義されています:

1. `minimal`
2. `standard` (デフォルト)
3. `verbose`
4. `all-field-info`
5. `all-field-info-verbose`
6. `super-verbose`
7. `timesketch-minimal`
8. `timesketch-verbose`

このファイルを編集することで、簡単に独自のプロファイルをカスタマイズしたり、追加したりすることができます。`set-default-profile -P <profile>` オプションでデフォルトのプロファイルを変更することもできます。利用可能なプロファイルとそのフィールド情報を表示するには、`csv-timeline --list-profiles` オプションを使用してください。

1. `minimal` プロファイルの出力

%Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %RuleTitle%, %Details%

2. standardプロファイルの出力

%Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %RecordID%, %RuleTitle%,
%Details%

3. verboseプロファイルの出力

%Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %MitreTactics, %MitreTags%,
%OtherTags%, %RecordID%, %RuleTitle%, %Details%, %RuleFile%, %EvtxFile%

4. all-field-infoプロファイルの出力

最小限のdetails情報を出力する代わりに、イベントにあるすべてのEventDataフィールド情報(%AllFieldInfo%)が出力されます。

%Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %RecordID%, %RuleTitle%,
%AllFieldInfo%, %RuleFile%, %EvtxFile%

5. all-field-info-verboseプロファイルの出力

all-field-infoとタグ情報が出力されます。

%Timestamp%, %Computer%, %Channel%, %EventID%, %Level%, %MitreTactics, %MitreTags%,
%OtherTags%, %RecordID%, %RuleTitle%, %AllFieldInfo%, %RuleFile%, %EvtxFile%

6. super-verboseプロファイルの出力

verboseプロファイルで出力される情報とイベントにあるすべてのEventDataフィールド情報(%AllFieldInfo%)の両方が出力されます。 (注意: 出力ファイルサイズは約2倍になります!)

%Timestamp%, %Computer%, %Channel%, %Provider%, %EventID%, %Level%, %MitreTactics,
%MitreTags%, %OtherTags%, %RecordID%, %RuleTitle%, %RuleAuthor%, %RuleCreationDate%,
%RuleModifiedDate%, %Status%, %Details%, %RuleFile%, %EvtxFile%, %AllFieldInfo%

7. timesketch-minimalプロファイルの出力

Timesketchにインポートできるverboseプロファイル。

%Timestamp%, hayabusa, %RuleTitle%, %Computer%, %Channel%, %EventID%, %Level%,
%MitreTactics, %MitreTags%, %OtherTags%, %RecordID%, %Details%, %RuleFile%, %EvtxFile%

8. timesketch-verboseプロファイルの出力

Timesketchにインポートできるverboseプロファイル。 (注意: 出力ファイルサイズは約2倍になります!)

%Timestamp%, hayabusa, %RuleTitle%, %Computer%, %Channel%, %EventID%, %Level%,
%MitreTactics, %MitreTags%, %OtherTags%, %RecordID%, %Details%, %RuleFile%, %EvtxFile%,
%AllFieldInfo%

プロファイルの比較

以下のベンチマークは、2018年製のマックブックプロ上で7.5GBのEVTXデータに対して実施されました。

プロファイル	処理時間	結果のファイルサイズ
minimal	16分18秒	690 MB
standard	16分23秒	710 MB
verbose	17分	990 MB
timesketch-minimal	17分	1015 MB
all-field-info-verbose	16分50秒	1.6 GB
super-verbose	17分12秒	2.1 GB

Profile Field Aliases

エイリアス名	Hayabusaの出力情報
%Timestamp%	デフォルトではYYYY-MM-DD HH:mm:ss.sss +hh:mm形式になっている。イベントログの<Event><System><TimeCreated SystemTime>フィールドから来ている。デフォルトのタイムゾーンはローカルのタイムゾーンになるが、--UTCオプションでUTCに変更することができる。
%Computer%	イベントログの<Event><System><Computer>フィールド。
%Channel%	ログ名。イベントログの<Event><System><EventID>フィールド。
%EventID%	イベントログの<Event><System><EventID>フィールド。
%Level%	YML検知ルールのlevelフィールド。(例: informational、low、medium、high、critical)
%MitreTactics%	MITRE ATT&CKの戦術(例: Initial Access、Lateral Movement等々)
%MitreTags%	MITRE ATT&CKの戦術以外の情報。attack.g(グループ)、attack.t(技術)、attack.s(ソフトウェア)の情報を出力する。
%OtherTags%	YML検知ルールのtagsフィールドからMitreTactics、MitreTags以外のキーワードを出力する。
%RecordID%	<Event><System><EventRecordID>フィールドのイベントレコードID。
%RuleTitle%	YML検知ルールのtitleフィールド。

エイリアス名	Hayabusaの出力情報
%Details%	YML検知ルールの details フィールドから来ていますが、このフィールドは Hayabusaルールにしかありません。このフィールドはアラートとイベントに関する追加情報を提供し、ログのフィールドから有用なデータを抽出することができます。イベントキーのマッピングが間違っている場合、もしくはフィールドが存在しない場合で抽出ができなかった箇所は n/a (not available) と記載されます。YML検知ルールに details フィールドが存在しない時の details のメッセージを ./rules/config/default_details.txt で設定できます。 default_details.txt では Provider Name 、 EventID 、 details の組み合わせで設定することができます。 default_details.txt や YML検知ルールに対応するルールが記載されていない場合はすべてのフィールド情報を出力します。
%AllFieldInfo%	すべてのフィールド情報。
%RuleFile%	アラートまたはイベントを生成した検知ルールのファイル名。
%EvtxFile%	アラートまたはイベントを起こしたevtxファイルへのパス。
%RuleAuthor%	YML検知ルールの author フィールド。
%RuleCreationDate%	YML検知ルールの date フィールド。
%RuleModifiedDate%	YML検知ルールの modified フィールド。
%Status%	YML検知ルールの status フィールド。
%RuleID%	YML検知ルールの id フィールド。
%Provider%	<Event><System><Provider> フィールド内の Name 属性。
%RenderedMessage%	WEC機能で転送されたイベントログの <Event><RenderingInfo><Message> フィールド。

これらのエイリアスは、出力プロファイルで使用することができます。また、他の [イベントキーエイリアス](#) を定義し、他のフィールドを出力することもできます。

Levelの省略

簡潔に出力するためにLevelを以下のように省略し出力しています。

- **crit: critical**
- **high: high**
- **med : medium**
- **low : low**
- **info: informational**

MITRE ATT&CK戦術の省略

簡潔に出力するためにMITRE ATT&CKの戦術を以下のように省略しています。

./config/mitre_tactics.txt の設定ファイルで自由に編集できます。

- **Recon** : Reconnaissance (偵察)

- **ResDev** : Resource Development (リソース開発)
- **InitAccess** : Initial Access (初期アクセス)
- **Exec** : Execution (実行)
- **Persis** : Persistence (永続化)
- **PrivEsc** : Privilege Escalation (権限昇格)
- **Evas** : Defense Evasion (防御回避)
- **CredAccess** : Credential Access (認証情報アクセス)
- **Disc** : Discovery (探索)
- **LatMov** : Lateral Movement (横展開)
- **Collect** : Collection (収集)
- **C2** : Command and Control (遠隔操作)
- **Exfil** : Exfiltration (持ち出し)
- **Impact** : Impact (影響)

Channel情報の省略

簡潔に出力するためにChannelの表示を以下のように省略しています。

`./rules/config/channel_abbreviations.txt`の設定ファイルで自由に編集できます。

- **App** : Application
- **AppLocker** : Microsoft-Windows-AppLocker/*
- **BitsCli** : Microsoft-Windows-Bits-Client/Operational
- **CodeInteg** : Microsoft-Windows-CodeIntegrity/Operational
- **Defender** : Microsoft-Windows-Windows Defender/Operational
- **DHCP-Svr** : Microsoft-Windows-DHCP-Server/Operational
- **DNS-Svr** : DNS Server
- **DrvFmwk** : Microsoft-Windows-DriverFrameworks-UserMode/Operational
- **Exchange** : MSExchange Management
- **Firewall** : Microsoft-Windows-Windows Firewall With Advanced Security/Firewall
- **KeyMgtSvc** : Key Management Service
- **LDAP-Cli** : Microsoft-Windows-LDAP-Client/Debug
- **NTLM** : Microsoft-Windows-NTLM/Operational
- **OpenSSH** : OpenSSH/Operational
- **PrintAdm** : Microsoft-Windows-PrintService/Admin
- **PrintOp** : Microsoft-Windows-PrintService/Operational
- **PwSh** : Microsoft-Windows-PowerShell/Operational
- **PwShClassic** : Windows PowerShell
- **RDP-Client** : Microsoft-Windows-TerminalServices-RDPClient/Operational
- **Sec** : Security
- **SecMitig** : Microsoft-Windows-Security-Mitigations/*
- **SmbCliSec** : Microsoft-Windows-SmbClient/Security
- **SvcBusCli** : Microsoft-ServiceBus-Client
- **Sys** : System
- **Sysmon** : Microsoft-Windows-Sysmon/Operational
- **TaskSch** : Microsoft-Windows-TaskScheduler/Operational
- **WinRM** : Microsoft-Windows-WinRM/Operational
- **WMI** : Microsoft-Windows-WMI-Activity/Operational

その他の省略

できるだけ簡潔にするために、以下の略語を使用しています:

- **Acct** -> Account
- **Addr** -> Address
- **Auth** -> Authentication
- **Cli** -> Client
- **Chan** -> Channel
- **Cmd** -> Command
- **Cnt** -> Count
- **Comp** -> Computer
- **Conn** -> Connection/Connected
- **Creds** -> Credentials
- **Crit** -> Critical
- **Disconnect** -> Disconnection/Disconnected
- **Dir** -> Directory
- **Drv** -> Driver
- **Dst** -> Destination
- **EID** -> Event ID
- **Err** -> Error
- **Exec** -> Execution
- **FW** -> Firewall
- **Grp** -> Group
- **Img** -> Image
- **Inj** -> Injection
- **Krb** -> Kerberos
- **LID** -> Logon ID
- **Med** -> Medium
- **Net** -> Network
- **Obj** -> Object
- **Op** -> Operational/Operation
- **Proto** -> Protocol
- **PW** -> Password
- **Reconn** -> Reconnection
- **Req** -> Request
- **Rsp** -> Response
- **Sess** -> Session
- **Sig** -> Signature
- **Susp** -> Suspicious
- **Src** -> Source
- **Svc** -> Service
- **Svr** -> Server
- **Temp** -> Temporary
- **Term** -> Termination/Terminated
- **Tkt** -> Ticket

- **Tgt** -> Target
- **Unkwn** -> Unknown
- **Usr** -> User
- **Perm** -> Permanent
- **Pkg** -> Package
- **Priv** -> Privilege
- **Proc** -> Process
- **PID** -> Process ID
- **PGUID** -> Process GUID (Global Unique ID)
- **Ver** -> Version

プログレスバー

プログレス・バーは、複数のevtxファイルに対してのみ機能します。 解析したevtxファイルの数と割合をリアルタイムで表示します。

標準出力へのカラー設定

Hayabusaの結果は**level**毎に文字色が変わります。 `./config/level_color.txt`の値を変更することで文字色を変えることができます。 形式は**level名, (6桁のRGBのカラーhex)**です。 カラー出力をしないようにしたい場合は`--no-color`オプションをご利用ください。

結果のサマリ

元々のイベント数、検知したイベント数、データ削減の統計、検知数情報、最多検知日、最多検知端末名、最多アラート等の情報がスキャン後に出力されます。

イベント頻度タイムライン

`-T`または`--visualize-timeline`オプションを使うことで、検知したイベントの数が5以上の時、頻度のタイムライン(スパークライン)を画面に出力します。 マーカーの数は最大10個です。 デフォルトのCommand PromptとPowerShell Promptでは文字化けがあるので、Windows TerminalやiTerm2等のターミナルをご利用ください。

Hayabusaルール

Hayabusa検知ルールはSigmaのようなYML形式で記述され、**rules**ディレクトリに入っています。

<https://github.com/Yamato-Security/hayabusa-rules>のレポジトリで管理しているので、ルールのissueやpull requestはhayabusaのレポジトリではなく、ルールレポジトリへお願いします。

ルールの作成方法については、[hayabusa-rulesレポジトリのREADME](#)をお読みください。

hayabusa-rulesレポジトリにあるすべてのルールは、**rules**フォルダに配置する必要があります。**level**がinformationのルールは**イベント**とみなされ、**low**以上は**アラート**とみなされます。

Hayabusaルールのディレクトリ構造は、2つのディレクトリに分かれています:

- **builtin**: Windowsの組み込み機能で生成できるログ。
- **sysmon**: [sysmon](#)によって生成されるログ。

ルールはさらにログタイプ（例：Security、Systemなど）によってディレクトリに分けられ、次の形式で名前が付けられます。

現在のルールをご確認いただき、新規作成時のテンプレートとして、また検知ロジックの確認用としてご利用ください。

Hayabusa v.s. 変換されたSigmaルール

Sigmaルールは、最初にHayabusaルール形式に変換する必要があります。変換のやり方は[ここで説明されています](#)。 Hayabusaルールは|contains|all、1 of selection*、all of selection*、Rust正規表現クレートでは機能しない正規表現を使用するルールをデフォルトで対応していないため、コンバータが必要です。殆どのルールはSigmaルールと互換性があるので、Sigmaルールのようにその他のSIEM形式に変換できます。 Hayabusaルールは、Windowsのイベントログ解析専用に設計されており、以下のような利点があります：

1. ログの有用なフィールドのみから抽出された追加情報を表示するための details フィールドを追加しています。
2. Hayabusaルールはすべてサンプルログに対してテストされ、検知することが確認されています。

変換処理のバグ、サポートされていない機能、実装の違い(正規表現など)により、一部のSigmaルールは意図したとおりに動作しない可能性があります。

3. Sigmaルール仕様がない集計式(例：|equalsfield)の利用。

制限事項: 私たちの知る限り、Hayabusa はオープンソースの Windows イベントログ解析ツールの中でSigmaルールを最も多くサポートしていますが、まだサポートされていないルールもあります。

1. Sigmaルール仕様のcount以外の集計式。
2. |near、|base64offset|containsを使用するルール。

検知ルールのチューニング

ファイアウォールやIDSと同様に、シグネチャベースのツールは、環境に合わせて調整が必要になるため、特定のルールを永続的または一時的に除外する必要がある場合があります。

ルールID(例: 4fe151c2-ecf9-4fae-95ae-b88ec9c2fca6)を ./rules/config/exclude_rules.txt に追加すると、不要なルールや利用できないルールを無視することができます。

ルールIDを ./rules/config/noisy_rules.txt に追加して、デフォルトでルールを無視することもできますが、-nまたは--enable-noisy-rulesオプションを指定してルールを使用することもできます。

検知レベルのlevelチューニング

Hayabusaルール、Sigmaルールはそれぞれの作者が検知した際のリスクレベルを決めています。 ユーザが独自のリスクレベルに設定するには./rules/config/level_tuning.txtに変換情報を書き、hayabusa-2.0.0-win-x64.exe level-tuningを実行することでルールファイルが書き換えられます。 ルールファイルが直接書き換えられることに注意して使用してください。

./rules/config/level_tuning.txtの例:

```
id,new_level  
00000000-0000-0000-0000-000000000000,informational # sample level tuning  
line
```

ルールディレクトリ内で `id` が `00000000-0000-0000-0000-000000000000` のルールのリスクレベルが `informational` に書き換えられます。

イベントIDフィルタリング

デフォルトでは、すべてのイベントに対してスキャンを行います。パフォーマンスを上げるために、検知ルールでイベントIDが定義されていないイベントを `-e`, `--eid-filter` オプションで無視することができます。

`./rules/config/target_event_IDs.txt` で定義されたIDがスキャンされます。

その他のWindowsイベントログ解析ツールおよび関連リソース

「すべてを統治する1つのツール」というものではなく、それぞれにメリットがあるため、これらの他の優れたツールやプロジェクトをチェックして、どれが気に入ったかを確認することをお勧めします。

- [APT-Hunter](#) - Pythonで開発された攻撃検知ツール。
- [Awesome Event IDs](#) - フォレンジック調査とインシデント対応に役立つイベントIDのリソース。
- [Chainsaw](#) - Rustで開発されたSigmaベースの攻撃検知ツール。
- [DeepBlueCLI](#) - [Eric Conrad](#) によってPowerShellで開発された攻撃検知ツール。
- [Epagneul](#) - Windowsイベントログの可視化ツール。
- [EventList](#) - [Miriam Wiesner](#)によるセキュリティベースラインの有効なイベントIDをMITRE ATT&CKにマッピングするPowerShellツール。
- [MITRE ATT&CKとWindowイベントログIDのマッピング](#) - 作者：[Michel de CREVOISIER](#)
- [EvtxECmd](#) - [Eric Zimmerman](#)によるEvtxパーサー。
- [EVTXtract](#) - 未使用領域やメモリダンプからEVTXファイルを復元するツール。
- [EvtxToElk](#) - Elastic StackにEvtxデータを送信するPythonツール。
- [EVTX ATTACK Samples](#) - [SBousseaden](#) によるEVTX攻撃サンプルイベントログファイル。
- [EVTX-to-MITRE-Attack](#) - [Michel de CREVOISIER](#)によるATT&CKにマッピングされたEVTX攻撃サンプルログのレポジトリ。
- [EVTX parser](#) - [@OBenamram](#) によって書かれた、Hayabusaが使用しているRustライブラリ。
- [Grafiki](#) - SysmonとPowerShellログの可視化ツール。
- [LogonTracer](#) - [JPCERTCC](#) による、横方向の動きを検知するためにログオンを視覚化するグラフィカルなインターフェース。
- [NSA Windows Event Monitoring Guidance](#) - NSAのWindowsイベントログ監視ガイド。
- [RustyBlue](#) - 大和セキュリティによるDeepBlueCLIのRust版。
- [Sigma](#) - コミュニティベースの汎用SIEMルール。
- [SOF-ELK](#) - [Phil Hagen](#) によるDFIR解析用のElastic Stack VM。
- [so-import-evtx](#) - evtxファイルをSecurityOnionにインポートするツール。
- [SysmonTools](#) - Sysmonの設定とオフライン可視化ツール。
- [Timeline Explorer](#) - [Eric Zimmerman](#) による最高のCSVタイムラインアライザ。

- [Windows Event Log Analysis - Analyst Reference](#) - Forward DefenseのSteve AnsonによるWindowsイベントログ解析の参考資料。
- [WELA \(Windows Event Log Analyzer\)](#) - Yamato SecurityによるWindowsイベントログ解析のマルチツール。
- [Zircelite](#) - Pythonで書かれたSigmaベースの攻撃検知ツール。

Windowsイベントログ設定のススメ

Windows機での悪性な活動を検知する為には、デフォルトのログ設定を改善することが必要です。どのようなログ設定を有効にする必要があるのか、また、自動的に適切な設定を有効にするためのスクリプトを、別のプロジェクトとして作成しました: <https://github.com/Yamato-Security/EnableWindowsLogSettings>

以下のサイトを閲覧することもおすすめします。:

- [JSCU-NL \(Joint SIGHT Cyber Unit Netherlands\) Logging Essentials](#)
- [ACSC \(Australian Cyber Security Centre\) Logging and Fowarding Guide](#)
- [Malware Archaeology Cheat Sheets](#)

Sysmon関係のプロジェクト

フォレンジックに有用な証拠を作り、高い精度で検知をさせるためには、sysmonをインストールする必要があります。以下のサイトを参考に設定することをおすすめします。:

- [Sysmon Modular](#)
- [TrustedSec Sysmon Community Guide](#)
- [SwiftOnSecurityのSysmon設定ファイル](#)
- [Neo23x0によるSwiftOnSecurityのSysmon設定ファイルのフォーク](#)
- [ion-stormによるSwiftOnSecurityのSysmon設定ファイルのフォーク](#)

コミュニティによるドキュメンテーション

英語

- 2022/06/19 [VelociraptorチュートリアルとHayabusaの統合方法](#) by Eric Capuano
- 2022/01/24 [Hayabusa結果をneo4jで可視化する方法](#) by Matthew Seyer (@forensic_matt)

日本語

- 2022/01/22 [Hayabusa結果をElastic Stackで可視化する方法](#) by @kzzzo2
- 2021/12/31 [Windowsイベントログ解析ツール「Hayabusa」を使ってみる](#) by itiB (@itiB_S144)
- 2021/12/27 [Hayabusaの中身](#) by Kazuminn (@k47_um1n)

貢献

どのような形でも構いませんので、ご協力を願いします。プルリクエスト、ルール作成、evtxログのサンプルなどがベストですが、機能リクエスト、バグの通知なども大歓迎です。

少なくとも、私たちのツールを気に入っていただけなら、GitHubで星を付けて、あなたのサポートを表明してください。

バグの報告

見つけたバグを[こちら](#)でご連絡ください。報告されたバグを喜んで修正します！

ライセンス

Hayabusaは[GPLv3](#)で公開され、すべてのルールは[Detection Rule License \(DRL\) 1.1](#)で公開されています。

Twitter

[@SecurityYamato](#)でHayabusa、ルール更新、その他の大和セキュリティツール等々について情報を提供しています。