HAYABUSA

[English] | [**日本語**]

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生成**および**スレット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。

目次

- Hayabusa について
  - 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 (-Vオプション)
  - 結果サマリ画面
  - 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
    - Linuxでのコンパイルの注意点
- 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
- 使用方法
    - 主なコマンド
    - コマンドラインオプション
    - 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`プロファイルの出力
        - 8. `timesketch`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
        - 最多検知日の出力
        - 最多検知端末名の出力

- 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、2600以上のSigmaルールと130以上の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可能です。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。

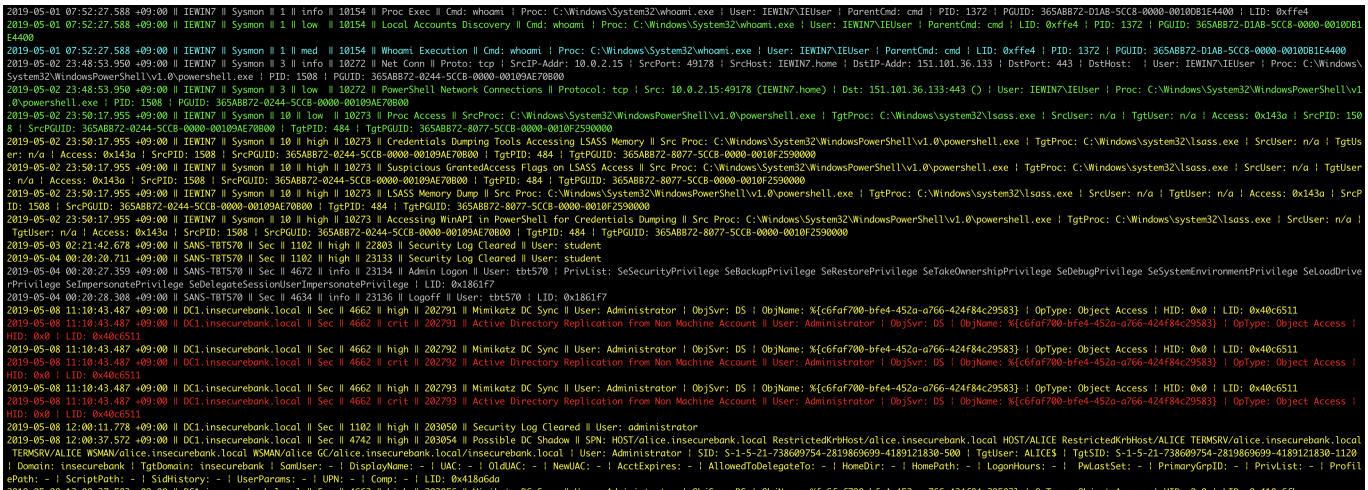
フォレンジックタイムラインの高速生成

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。

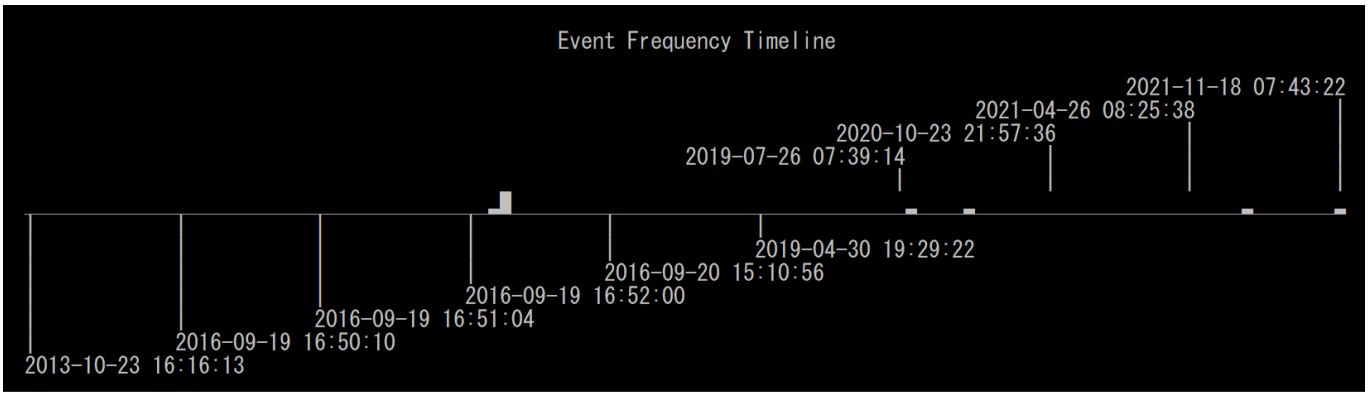
スクリーンショット

起動画面

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 (–V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:                                    Top high alerts:
 ------------------------------------------------------------------------------------------------------------------------
 Sticky Key Like Backdoor Usage (10)                     Metasploit SMB Authentication (3,562)
 Meterpreter or Cobalt Strike Getsystem Service Installation (6)  Malicious Svc Possibly Installed (271)
 Active Directory Replication from Non Machine Account (6)  Susp Svc Installed (257)
 Windows Defender Alert (4)                              PowerShell Scripts Installed as Services (253)
 WannaCry Ransomware (4)                                 Suspicious Service Installation Script (250)
 ------------------------------------------------------------------------------------------------------------------------
 Top medium alerts:                                      Top low alerts:
 ------------------------------------------------------------------------------------------------------------------------
 Potentially Malicious PwSh (235)                        Logon Failure_Wrong Password (3,564)
 Proc Injection (104)                                    Susp CmdLine (Possible LOLBIN) (1,418)
 Reg Key Value Set_Sysmon Alert (103)                    Non Interactive PowerShell (325)
 Suspicious Remote Thread Target (93)                    Rare Service Installations (321)
 Cscript Visual Basic Script Execution (60)              Windows Processes Suspicious Parent Directory (282)
 ------------------------------------------------------------------------------------------------------------------------
 Top informational alerts:
 ------------------------------------------------------------------------------------------------------------------------
 Proc Exec (11,173)                                      Explicit Logon (342)
 NetShare File Access (2,564)                            Svc Installed (331)
 PwSh Scriptblock (789)                                  New Non-USB PnP Device (268)
 PwSh Pipeline Exec (680)                                Logon (Type 3 Network) (228)
 NetShare Access (433)                                   File Created (210)

Elapsed Time: 00:00:28.827
```

Excelでの解析

Time	Computername	Eventi	Level	Alert	Details
2021-05-03 17:58:38.774 +09:00	webiis01.offsec.lan	4624	informational	Logon Type 3 - Network	User: admmig : 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: admmig : 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: admmig : 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\s
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: admmig
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: admmig
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\OpenS
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

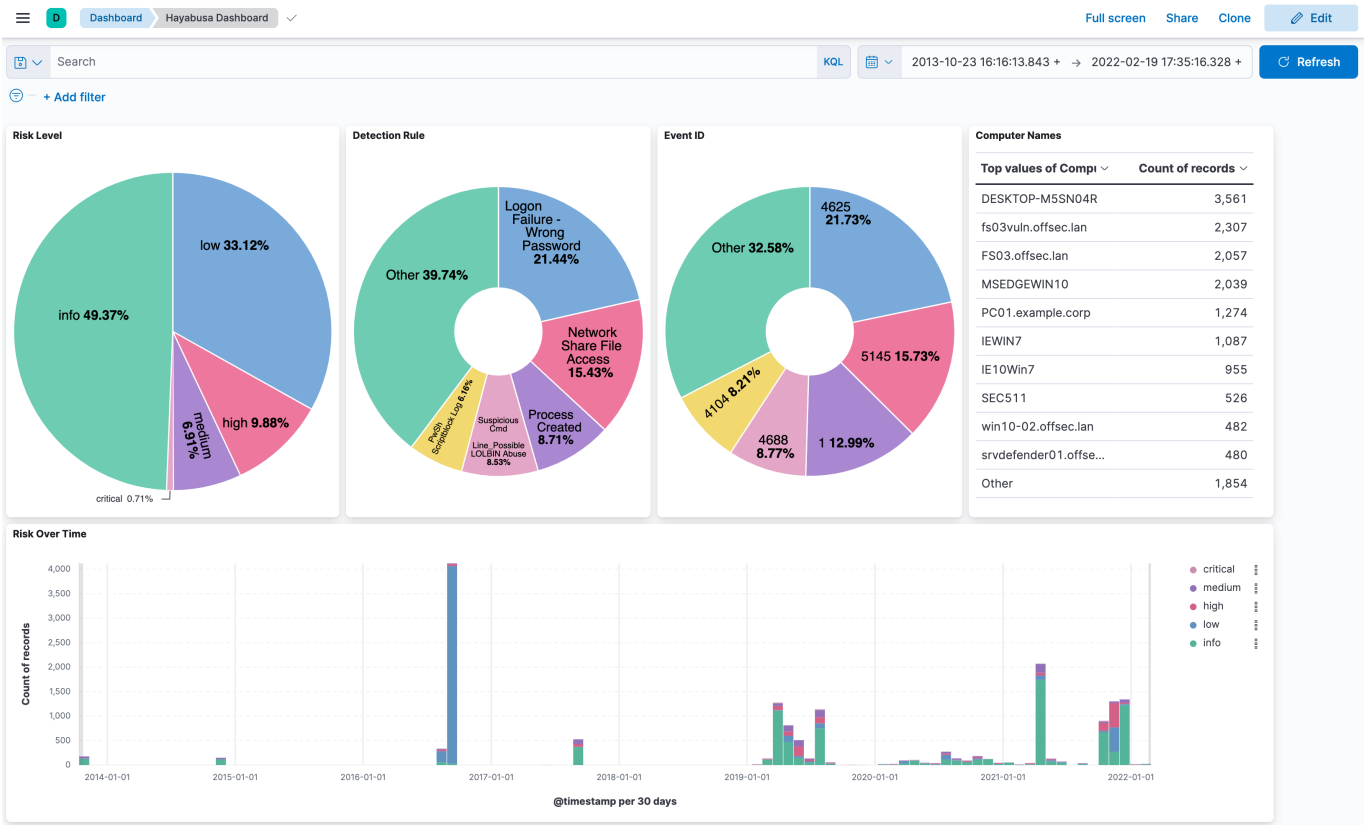
Time	Computername	Eventid	Level	Alert	Details
2021-05-22 05:43:18.227 +09:00	fs01.offsec.lan	4648	informational	Explicit Logon	Source User: FS01$: Target User: admmig
2021-05-22 05:43:22.562 +09:00	fs01.offsec.lan	4625	low	Logon Failure - Wrong Password	User: admmig@offsec.lan : Type: 8 : Wor
2021-05-22 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 22:02:34.375 +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-1-5-21-1
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

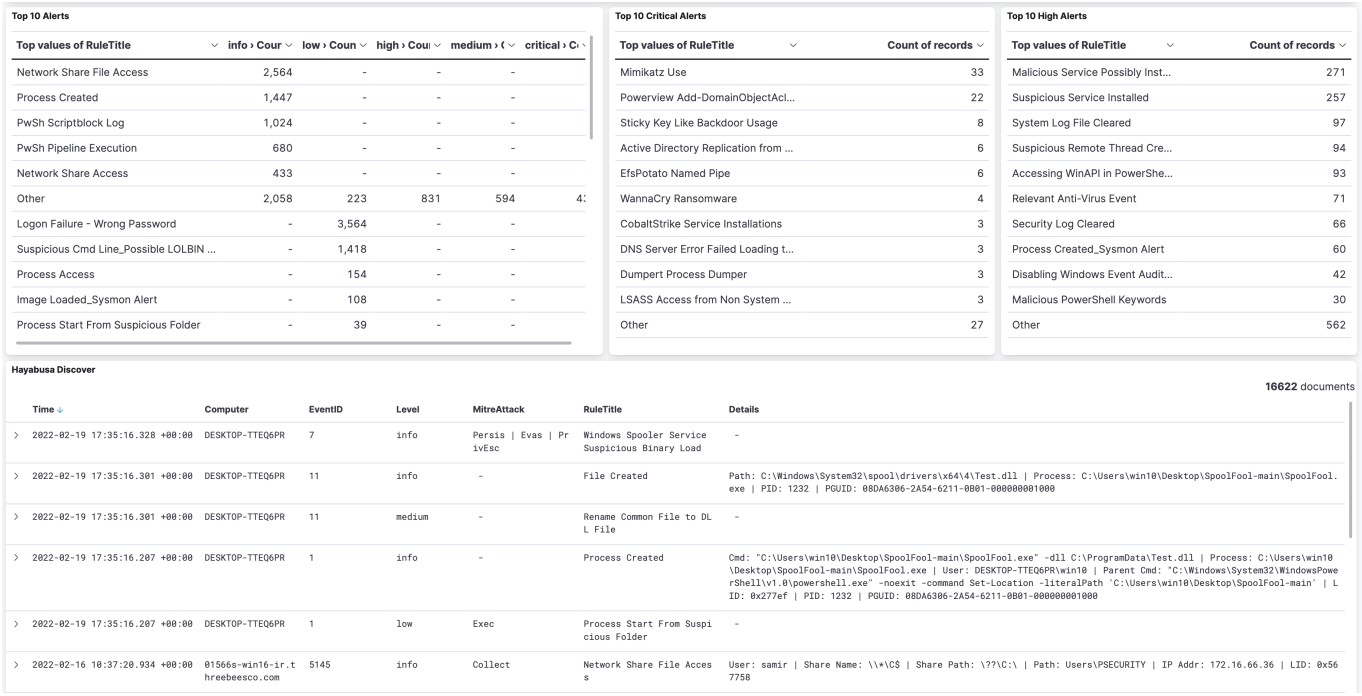
Criticalアラートのフィルタリングとコンピュータごとのグルーピング

Computername

Line	Tag	Time	Eventid	Level	Alert
=				= critical	
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 values of RuleTitle	info › Cour	low › Coun	high › Cou	medium › (	critical › C
Network Share File Access	2,564	-	-	-	
Process Created	1,447	-	-	-	
PwSh Scriptblock Log	1,024	-	-	-	
PwSh Pipeline Execution	680	-	-	-	
Network Share Access	433	-	-	-	
Other	2,058	223	831	594	4
Logon Failure - Wrong Password	-	3,564	-	-	
Suspicious Cmd Line_Possible LOLBIN ...	-	1,418	-	-	
Process Access	-	154	-	-	
Image Loaded_Sysmon Alert	-	108	-	-	
Process Start From Suspicious Folder	-	39	-	-	

Top 10 Critical Alerts

Top values of RuleTitle	Count of records
Mimikatz Use	33
Powerview Add-DomainObjectAcl...	22
Sticky Key Like Backdoor Usage	8
Active Directory Replication from ...	6
EfsPotato Named Pipe	6
WannaCry Ransomware	4
CobaltStrike Service Installations	3
DNS Server Error Failed Loading t...	3
Dumpert Process Dumper	3
LSASS Access from Non System ...	3
Other	27

Top 10 High Alerts

Top values of RuleTitle	Count of records
Malicious Service Possibly Inst...	271
Suspicious Service Installed	257
System Log File Cleared	97
Suspicious Remote Thread Cre...	94
Accessing WinAPI in PowerShe...	93
Relevant Anti-Virus Event	71
Security Log Cleared	66
Process Created_Sysmon Alert	60
Disabling Windows Event Audit...	42
Malicious PowerShell Keywords	30
Other	562

Hayabusa Discover

16622 documents

Time ↓	Computer	EventID	Level	MitreAttack	RuleTitle	Details
2022-02-19 17:35:16.328 +00:00	DESKTOP-TTEQ6PR	7	info	Persis | Evas | PrivEsc	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\SpoolFool-main\SpoolFool.exe | PID: 1232 | PGUID: 08DA6306-2A54-6211-0B01-000000001000
2022-02-19 17:35:16.301 +00:00	DESKTOP-TTEQ6PR	11	medium	-	Rename Common File to DLL File	-
2022-02-19 17:35:16.207 +00:00	DESKTOP-TTEQ6PR	1	info	-	Process Created	Cmd: "C:\Users\win10\Desktop\SpoolFool-main\SpoolFool.exe" -dll C:\ProgramData\Test.dll | Process: C:\Users\win10\Desktop\SpoolFool-main\SpoolFool.exe | User: DESKTOP-TTEQ6PR\win10 | Parent Cmd: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -noexit -command Set-Location -literalPath 'C:\Users\win10\Desktop\SpoolFool-main' | LID: 0x277ef | PID: 1232 | PGUID: 08DA6306-2A54-6211-0B01-000000001000
2022-02-19 17:35:16.207 +00:00	DESKTOP-TTEQ6PR	1	low	Exec	Process Start From Suspicious Folder	-
2022-02-16 10:37:20.934 +00:00	01566s-win16-ir.threebeesco.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	☐	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	☐	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	☐	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	☐	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	☐	Dumpert Process Dumper	Path: C:\Windows\Temp\dumpert.dmp | Process: C:\Users\administrator\Desktop\x64\Outflank-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のタイムライン結果のサンプルはこちらで確認できます。

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の出力。

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、バグがある可能性もあるので、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-1.6.0-win-x64.exe -u
```

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:

```
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
```

macOSでのコンパイルの注意点

opensssl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

opensssl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
```

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。

Linuxでのコンパイルの注意点

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-1.6.0-windows-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-1.6.0-linux-x64-gnu
```

次に、Hayabusaのルートディレクトリから実行します：

```
./hayabusa-1.6.0-linux-x64-gnu
```

macOS

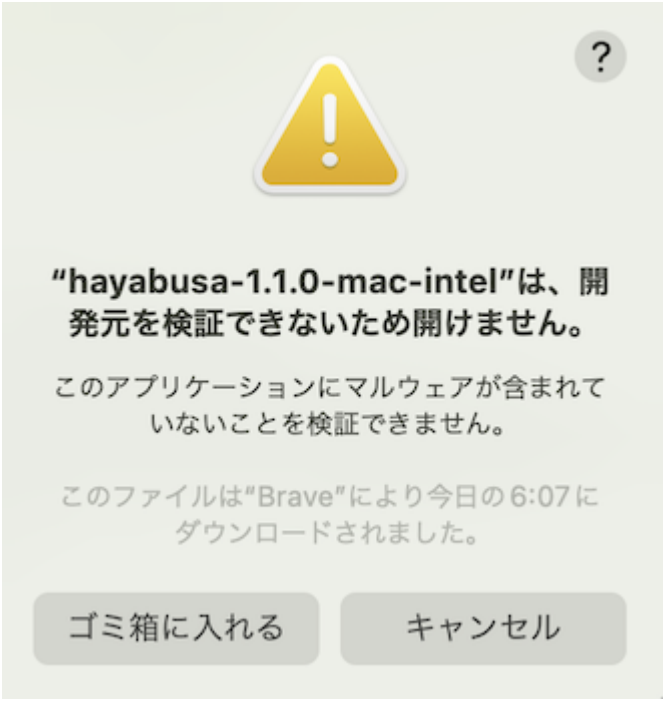
まず、ターミナルやiTerm2からバイナリに実行権限を与える必要があります。

```
chmod +x ./hayabusa-1.6.0-mac-intel
```

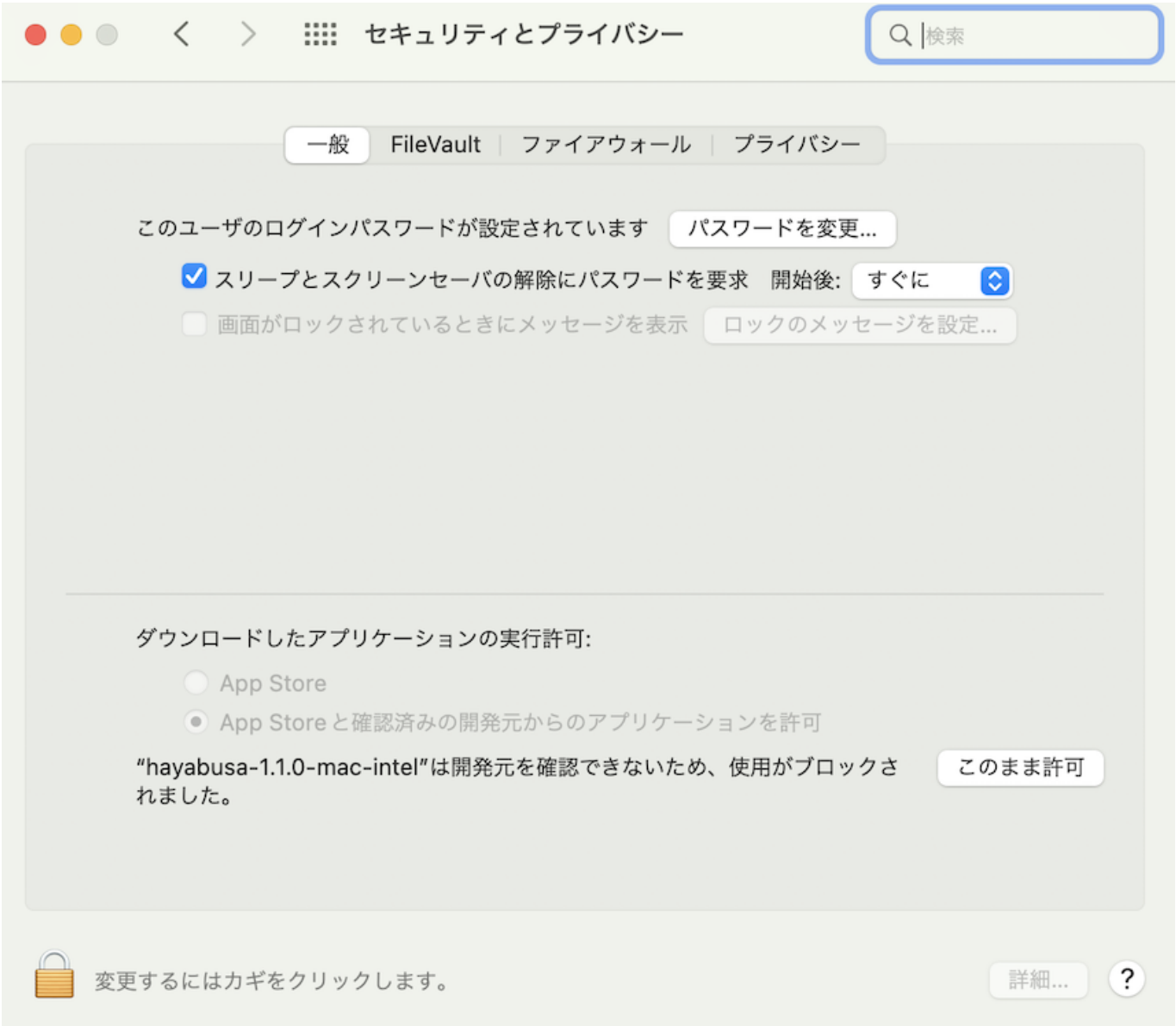
次に、Hayabusaのルートディレクトリから実行してみてください：

```
./hayabusa-1.6.0-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：



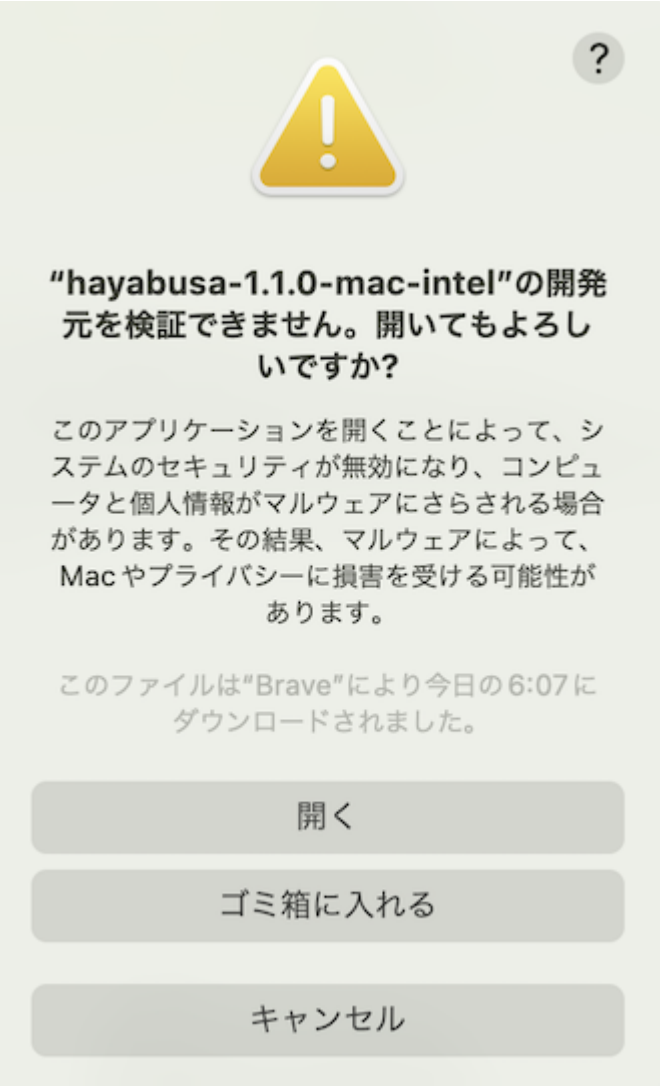
macOSの環境設定から「セキュリティとプライバシー」を開き、「一般」タブから「このまま許可」ボタンをクリックしてください。



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-1.6.0-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。

使用方法

主なコマンド

- デフォルト: ファストフォレンジックタイムラインの作成。
- `--level-tuning`: アラート`level`のカスタムチューニング
- `-L, --logon-summary`: ログオンイベントのサマリを出力する。
- `-P, --pivot-keywords-list`: ピボットする不審なキーワードのリスト作成。
- `-s, --statistics`: 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。
- `--set-default-profile`: デフォルトプロファイルを変更する。
- `-u, --update`: GitHubのhayabusa-rulesリポジトリにある最新のルールに同期させる。

コマンドラインオプション

```
USAGE:
    hayabusa.exe <INPUT> [OTHER-ACTIONS] [OPTIONS]

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:
    -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)
    -Q, --quiet-errors                          Quiet errorsモード: エラーロ
グを保存しない
    -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)
    -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)
        --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:
    -j, --json                              タイムラインの出力をJSON形式で保存す
る（例: -j -o results.json)
    -J, --jsonl                             タイムラインの出力をJSONL形式で保存す
る（例: -J -o results.jsonl)
    -o, --output <FILE>                     タイムラインをCSV形式で保存する（例:
results.csv)
    -P, --profile <PROFILE>                 利用する出力プロファイル名を指定する
(minimal, standard, verbose, verbose-all-field-info, verbose-details-and-
all-field-info)

DISPLAY-SETTINGS:
        --no-color              カラー出力を無効にする
        --no-summary            結果概要を出力しない
    -q, --quiet                 Quietモード: 起動バナーを表示しない
    -v, --verbose               詳細な情報を出力する
    -V, --visualize-timeline    イベント頻度タイムラインを出力する

FILTERING:
    -D, --deep-scan                         すべてのイベントIDを対象にしたスキャンを
行う（遅くなる)
        --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"）

OTHER-ACTIONS:
        --contributors                                コントリビュータの一覧表示
    -L, --logon-summary                               成功と失敗したログオン情報の要約
を出力する
        --level-tuning [<FILE>]                       ルールlevelのチューニング（デフ
ォルト: ./rules/config/level_tuning.txt）
    -p, --pivot-keywords-list                         ピボットキーワードの一覧作成
    -s, --statistics                                  イベントIDの統計情報を表示する
        --set-default-profile <PROFILE>               デフォルトの出力コンフィグを設定
する
    -u, --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

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）
        --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）
```

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-1.6.0-win-x64.exe -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-1.6.0-win-x64.exe -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: `verbose-details-and-all-field-info`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-1.6.0-win-x64.exe -d .\hayabusa-sample-evtx -o results.csv -P
verbose-details-and-all-field-info
```

- タイムラインをJSON形式で保存する:

```
hayabusa-1.6.0-win-x64.exe -d .\hayabusa-sample-evtx -o results.json -j
```

- 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-1.6.0-win-x64.exe -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-1.6.0-win-x64.exe -d .\hayabusa-sample-evtx -r
.\rules\hayabusa\default -o results.csv
```

- Sysmonログに対してのみHayabusaルールを実行する:

```
hayabusa-1.6.0-win-x64.exe -d .\hayabusa-sample-evtx -r
.\rules\hayabusa\sysmon -o results.csv
```

- Sigmaルールのみを実行する:

```
hayabusa-1.6.0-win-x64.exe -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-1.6.0-win-x64.exe -d .\hayabusa-sample-evtx --enable-deprecated-
rules --enable-noisy-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-1.6.0-win-x64.exe -d .\hayabusa-sample-evtx -r
.\rules\hayabusa\default\events\Security\Logons -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-1.6.0-win-x64.exe -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-Ip Address.txt`や`keywords-Users.txt`等に出力される):

```
hayabusa-1.6.0-win-x64.exe -l -m critical -p -o keywords
```

- イベントIDの統計情報を出力する:

```
hayabusa-1.6.0-win-x64.exe -f Security.evtx -s
```

- ログオンサマリを出力する:

```
hayabusa-1.6.0-win-x64.exe -L -f Security.evtx -s
```

- 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-1.6.0-win-x64.exe -d .\hayabusa-sample-evtx -v
```

- Verbose出力の例:

```
Checking target evtx FilePath: "./hayabusa-sample-
evtx/YamatoSecurity/T1027.004_Obfuscated 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-1.6.0-win-x64.exe -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

`-p`もしくは`--pivot-keywords-list`オプション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、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`オプションを使うときには `-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

`-L` または `--logon-summary` オプション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 <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。

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`フィールド情報が出力されます。
(注意: 出力ファイルサイズは2倍になります！)

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

7. `timesketch`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`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%`, `%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

プロファイル	処理時間	結果のファイルサイズ
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。
%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。

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、他のイベントキーアライズを定義し、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` : `med`
- `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/output_tag.txt`の設定ファイルで自由に編集できます。 検知したデータの戦術を全て出力したい場合は、`--all-tags`オプションをつけてください。

- `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`
- `DvrFmwk` : `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
- `Cmd` -> Command
- `Comp` -> Computer
- `Conn` -> Connection
- `Dir` -> Directory
- `Dst` -> Destination
- `Exec` -> Execution
- `Grp` -> Group
- `LID` -> Logon ID
- `Net` -> Network
- `Obj` -> Object
- `Proto` -> Protocol
- `Sig` -> Signature
- `Susp` -> Suspicious
- `Src` -> Source
- `Svc` -> Service
- `Svr` -> Server
- `Tgt` -> Target
- `Op` -> Operation
- `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

イベント頻度タイムライン

`-V`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では文字化けがでるので、Windows TerminalやiTerm2等のターミナルをご利用ください。

最多検知日の出力

各レベルで最も検知された日付を画面に出力します。

最多検知端末名の出力

各レベルで多く検知されたユニークなイベントが多い端末名上位5つを画面に出力します。

Hayabusaルール

Hayabusa検知ルールはSigmaのようなYML形式で記述されています。`rules`ディレクトリに入っていますが、将来的にはhttps://github.com/Yamato-Security/hayabusa-rulesのレポジトリで管理する予定なので、ルールのissueと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 `events` とみなされ、`low` 以上は `alerts` とみなされます。

Hayabusaルールのディレクトリ構造は、3つのディレクトリに分かれています。

- `default`: Windows OSでデフォルトで記録されるログ
- `non-default`: グループポリシーやセキュリティベースラインの適用でオンにする必要があるログ
- `sysmon`: sysmonによって生成されるログ。
- `testing`: 現在テストしているルールを配置するための一時ディレクトリ

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。

- アラート形式: `<イベントID>_<イベントの説明>_<リスクの説明>.yml`
- アラート例: `1102_SecurityLogCleared_PossibleAntiForensics.yml`
- イベント形式: `<イベントID>_<イベントの説明>.yml`
- イベント例: `4776_NTLM-LogonToLocalAccount.yml`

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。 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ルール仕様にない集計式(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. 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。
2. Sigmaルール仕様の`count`以外の集計式。
3. `|near`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-1.6.0-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無視しています。 `./rules/config/target_event_IDs.txt`で定義されたIDがスキャンされます。 すべてのイベントをスキャンしたい場合は、`-D, --deep-scan`オプションを使用してください。

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。
- 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タイムラインアナライザ。

- 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。
- Zircolite - 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。 以下のサイトを閲覧することをおすすめします。:

- JSCU-NL (Joint Sigint 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 @kzzzzo2
- 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、ご協力をお願いします。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たなら、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。