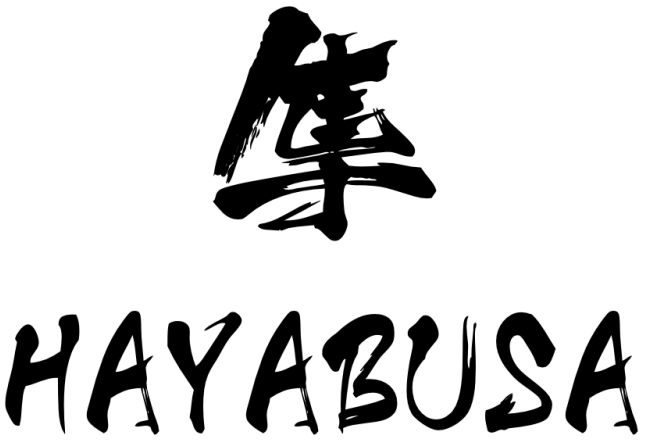


[English] | [**日本語**]

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等で簡単に分析できるようになります。

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

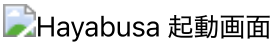
Hayabusaには現在、2300以上のSigmaルールと130以上のHayabusa検知ルールがあり、定期的にルールが追加されています。 VelociraptorのHayabusa artifactを用いることで企業向けの広範囲なスレットハンティングだけでなくDFIR(デジタルフォレンジックとインシデントレスポンス)にも無料で利用することが可能です。この2つのオープンソースを組み合わせることで、SIEMが設定されていない環境でも実質的に遡及してSIEMを再現することができます。具体的な方法はEric Cupuanoのこちらの動画で学ぶことができます。 最終的な目標はインシデントレスポンスや定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

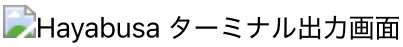
Windowsのイベントログは、 1）解析が困難なデータ形式であること 2）データの大半がノイズであり調査に有用でないこと から、従来は非常に長い時間と手間がかかる解析作業となっていました。 Hayabusa は、有用なデータのみを抽出し、専門的なトレーニングを受けた分析者だけでなく、Windowsのシステム管理者であれば誰でも利用できる読みやすい形式で提示することを主な目的としています。 Hayabusaは従来のWindowsイベントログ分析解析と比較して、分析者が20%の時間で80%の作業を行えるようにすることを目指しています。

スクリーンショット

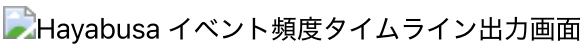
起動画面



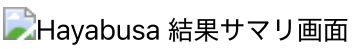
ターミナル出力画面



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



Excelでの解析

Timeline Explorerでの解析

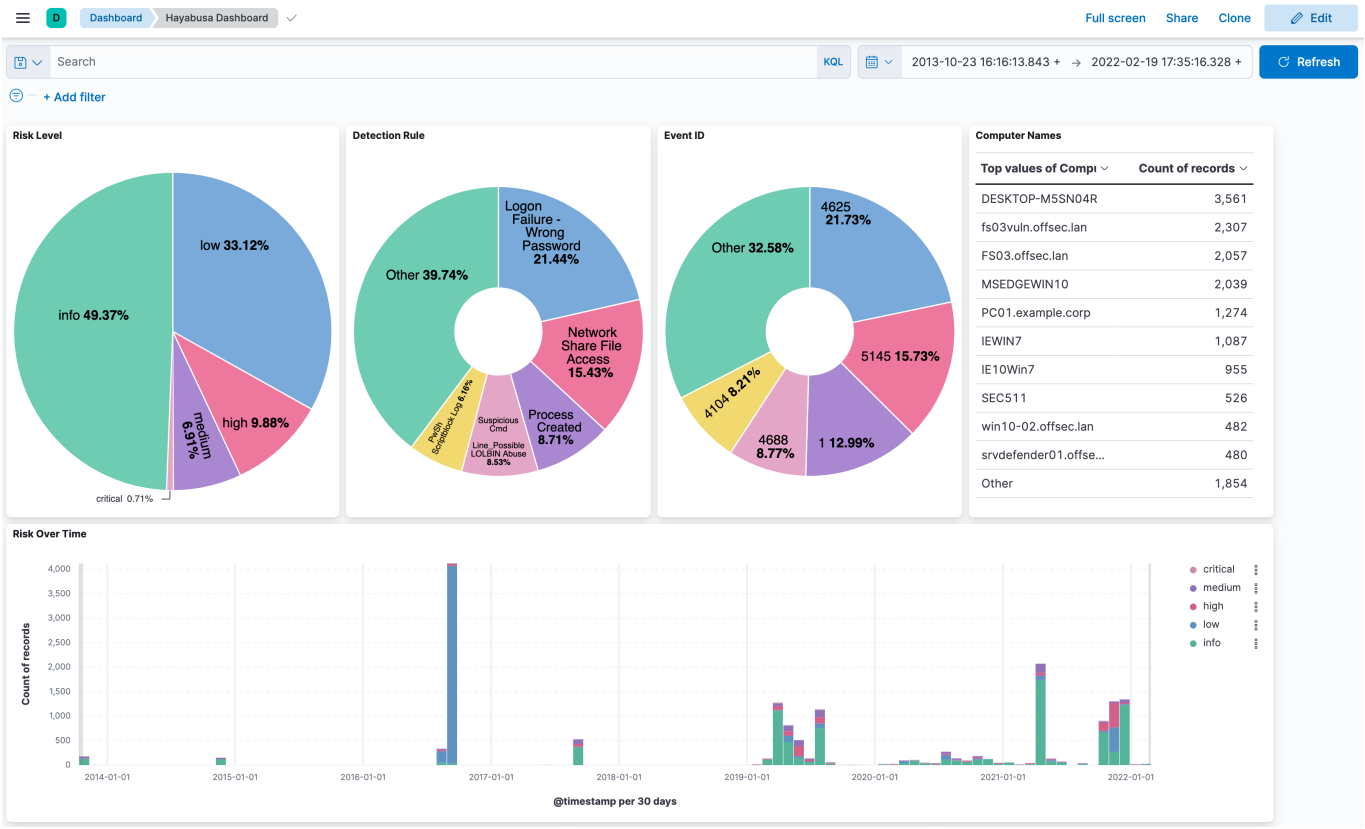
Time	Computername	Eventid	Level	Alert	Details
2021-05-22 05:43:18.227 +09:00	fs01.offsec.lan	4648	informational	Explicit Logon	Source User: FS01$: Target User: admmig
2021-05-22 05:43:22.562 +09:00	fs01.offsec.lan	4625	low	Logon Failure - Wrong Password	User: admmig@offsec.lan : Type: 8 : Wor
2021-05-22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

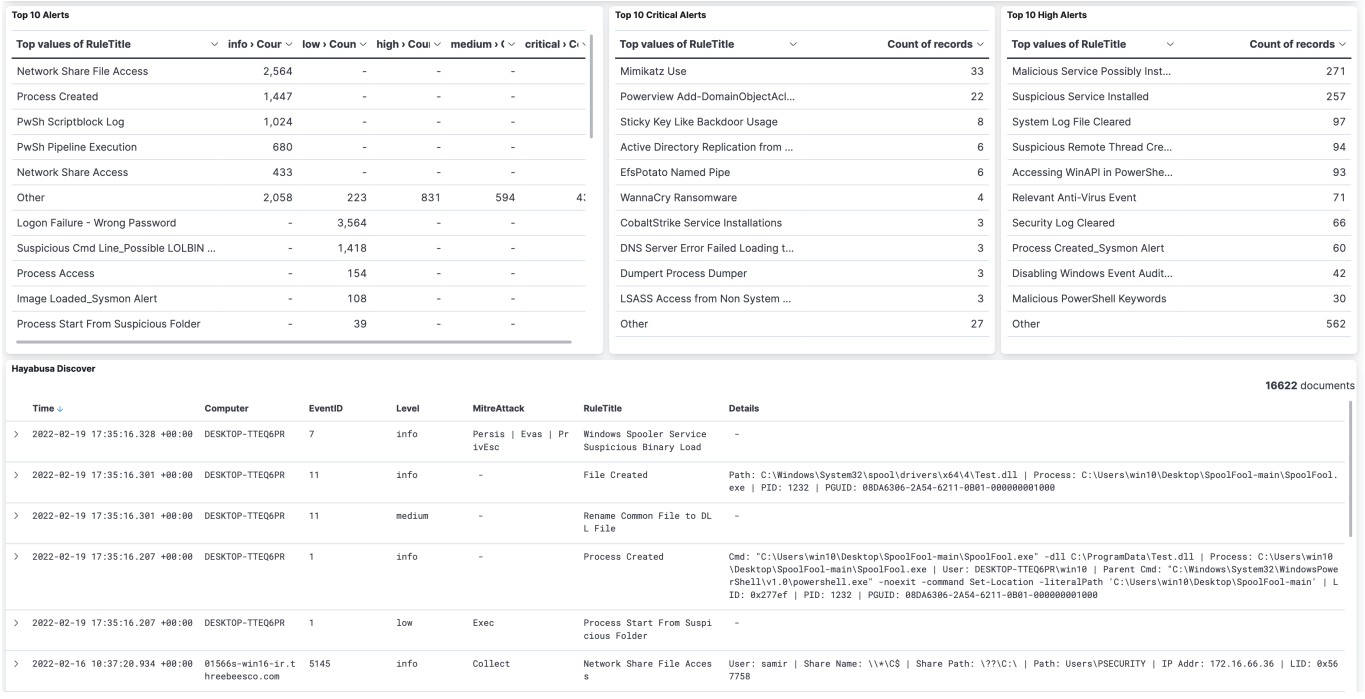
Criticalアラートのフィルタリングとコンピュータごとのグルーピング

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





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。

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。

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-1.4.0-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 clean
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

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
```

Hayabusaの実行

注意: アンチウィルス/EDRの誤検知

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

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-1.4.0-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-1.4.0-linux-x64-gnu
```

次に、Hayabusaのルートディレクトリから実行します：

```
./hayabusa-1.4.0-linux-x64-gnu
```

macOS

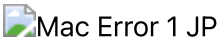
まず、ターミナルやiTerm2からバイナリに実行権限を与える必要があります。

```
chmod +x ./hayabusa-1.4.0-mac-intel
```

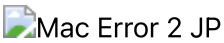
次に、Hayabusaのルートディレクトリから実行してみてください：

```
./hayabusa-1.4.0-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：



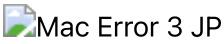
macOSの環境設定から「セキュリティとプライバシー」を開き、「一般」タブから「このまま許可」ボタンをクリックしてください。



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-1.4.0-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

コマンドラインオプション

```
USAGE:
    hayabusa.exe -f file.evtx [OPTIONS] / hayabusa.exe -d evtx-directory
[OPTIONS]

OPTIONS:
        --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）
        --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）
        --all-tags                                出力したCSVファイルにルール内
のタグ情報を全て出力する
    -c, --rules-config <RULE_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）
        --contributors                            コントリビュータの一覧表示
    -d, --directory <DIRECTORY>                   .evtxファイルを持つディレクト
リのパス
    -D, --enable-deprecated-rules                 Deprecatedルールを有効にする
        --end-timeline <END_TIMELINE>             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"）
        --exclude-status <EXCLUDE_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）
    -f, --filepath <FILE_PATH>                    1つの.evtxファイルに対して解
析を行う
    -F, --full-data                               全てのフィールド情報を出力する
    -h, --help                                    ヘルプ情報を表示する
    -l, --live-analysis                           ローカル端末の
C:\Windows\System32\winevt\Logsフォルダを解析する
    -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 [<LEVEL_TUNING_FILE>]      ルールlevelのチューニング
（デフォルト: ./rules/config/level_tuning.txt）
    -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
        --no_color                                カラー出力を無効にする
    -o, --output <CSV_TIMELINE>                   タイムラインをCSV形式で保存す
る（例: results.csv）
    -p, --pivot-keywords-list                     ピボットキーワードの一覧作成
    -q, --quiet                                   Quietモード: 起動バナーを表
示しない
    -Q, --quiet-errors                            Quiet errorsモード: エラー
ログを保存しない
    -r, --rules <RULE_DIRECTORY/RULE_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）
    -R, --hide-record-id                          イベントレコードIDを表示しな
い
    -s, --statistics                              イベントIDの統計情報を表示す
る
        --start-timeline <START_TIMELINE>         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"）
    -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）
    -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
    -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）
    -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
        --version                                 バージョン情報を表示する
```

使用例

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

```
hayabusa-1.4.0-win-x64.exe -f eventlog.evtx
```

- 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を実行します:

```
hayabusa-1.4.0-win-x64.exe -d .\hayabusa-sample-evtx
```

- 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ます(注意: `-F`を有効にすると、出力するファイルのサイズがとても大きくなります！):

```
hayabusa-1.4.0-win-x64.exe -d .\hayabusa-sample-evtx -o results.csv -F
```

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

```
hayabusa-1.4.0-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ルールを実行します:

```
hayabusa-1.4.0-win-x64.exe -d .\hayabusa-sample-evtx -r
.\rules\hayabusa\default -o results.csv
```

- Sysmonログに対してのみHayabusaルールを実行します:

```
hayabusa-1.4.0-win-x64.exe -d .\hayabusa-sample-evtx -r
.\rules\hayabusa\sysmon -o results.csv
```

- Sigmaルールのみを実行します:

```
hayabusa-1.4.0-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)を有効にします:

```
hayabusa-1.4.0-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します:

```
hayabusa-1.4.0-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）のみを検知します:

```
hayabusa-1.4.0-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します(結果は結果毎に`keywords-Ip Address.txt`や`keyworss-Users.txt`等に出力されます):

```
hayabusa-1.4.0-win-x64.exe -l -m critical -p -o keywords
```

- イベントIDの統計情報を取得します:

```
hayabusa-1.4.0-win-x64.exe -f Security.evtx -s
```

- 詳細なメッセージを出力します(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-1.4.0-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
```

- 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
```

> ※ 以下の例でHayabusaを試したい方は、上記コマンドをhayabusaのルートフォルダから実行してください。

Hayabusaの出力

Hayabusaの結果を標準出力に表示しているとき（デフォルト）は、以下の情報を表示します:

- `Timestamp`: デフォルトでは`YYYY-MM-DD HH:mm:ss.sss +hh:mm`形式になっています。イベントログの`<Event><System><TimeCreated SystemTime>`フィールドから来ています。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りますが、`--utc` オプションで UTC に変更することができます。
- `Computer`: イベントログの`<Event><System><Computer>`フィールドから来ています。
- `Channel`: ログ名です。イベントログの`<Event><System><EventID>`フィールドから来ています。
- `Event ID`: イベントログの`<Event><System><EventID>`フィールドから来ています。
- `Level`: YML検知ルールの`level`フィールドから来ています。(例：`informational`, `low`, `medium`, `high`, `critical`) デフォルトでは、すべてのレベルのアラートとイベントが出力されますが、`-m`オプションで最低のレベルを指定することができます。例えば`-m high`オプションを付けると、`high`と`critical`アラートしか出力されません。
- `Title`: YML検知ルールの`title`フィールドから来ています。
- `RecordID`: イベントレコードIDです。`<Event><System><EventRecordID>`フィールドから来ています。`-R`もしくは`--hide-record-id`オプションを付けると表示されません。
- `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。

CSVファイルとして保存する場合、以下の列が追加されます:

- MitreAttack: MITRE ATT&CKの戦術。
- Rule Path: アラートまたはイベントを生成した検知ルールへのパス。
- File Path: 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。

-Fもしくは--full-dataオプションを指定した場合、全てのフィールド情報がRecordInformationカラムにで出力されます。

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。
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

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。

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-1.4.0-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

config/target_eventids.txtにイベントID番号を追加することで、イベントIDでフィルタリングすることができます。 これはパフォーマンスを向上させるので、特定のIDだけを検索したい場合に推奨されます。

すべてのルールのEventIDフィールドと実際のスキャン結果で見られるIDから作成したIDフィルタリストのサンプルをconfig/target_eventids_sample.txtで提供しています。

最高のパフォーマンスを得たい場合はこのリストを使用してください。ただし、検出漏れの可能性が若干あることにご注意ください。

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、私たちが使用した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

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 Cupuano
- 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。