sysdig

Falcoルール チューナー

本文の内容は、Falcoルールチューナーのドキュメント(https://docs.sysdig.com/en/the-falco-rules-tuner.html）2020年11月25日時点を元に日本語に翻訳・再構成した内容となっております。

Falcoルールチューナー

Sysdigのポリシーは、Falcoのルールやマクロなど、ルールの上に構築されています。(レビュー：Sysdig Secureルールの理解とSysdig Secure内でのFalcoの使用) Sysdigは常に、よく知られたコンテナやOSSアプリケーションについてキャプチャーしたアクティビティに基づいて、すぐに使えるポリシーの改善に取り組んでいます。それにもかかわらず、独自のユーザー環境で実行されているプロプライエタリなソフトウェアは、カスタマイズされたアプローチが必要になることがあります。

Falcoルールチューナーは、既存のルールセットを更新して誤検知を減らすプロセスを簡素化するために作成されました。

このツールは、設定可能な時間ウィンドウ（EVENT_LOOKKBACK_MINUTES）の間に生成されたポリシーイベントを取得し、発生したしきい値（EVENT_COUNT_THRESHOLD）に基づいて、ルールの更新を提案します。提案を評価し、選択的に変更を適用するかどうかはユーザー次第です。

ルールチューナーを使用するには、いくつかの環境変数を提供し、Dockerコンテナとして実行し、Slackチャンネルまたはターミナルウィンドウで出力を確認し、Sysdig Secure Rules Editorで、推奨されるチューニング調整を必要に応じて適用します。

必要条件

- Sysdig Secure SaaSまたはOn-Premバージョン3.5.0以上
- 利用可能なSlackチャンネル（オプション、出力情報の受信用）
- 環境変数の値を以下の表に示します。

変数を設定してコンテナを実行する

以下の環境変数に必要な値を集めます。

表11. Falcoルールチューナーに必要な環境変数

変数	説明
`SECURE_CUSTOMER`	任意です。事業体の名前。デフォルト: test

`SECURE_ENDPOINT`	チューニング・エンジンがクエリするエンドポイント。 SaaS の場合は、「SaaS リージョンと IP レンジ」を参照してください。 オンプレミスの場合、エンドポイントはユーザー定義されています。
`SECURE_TOKEN`	Secure バックエンドへのアクセスに使用される Sysdig Secure API トークン。Sysdig API トークンを見つけるを参照してください。
`SLACK_WEBHOOK`	オプションです。イベントサマリーとルールチューニングの推奨事項を受け取るためのSlackのウェブフックURLです。 例: https://hooks.slack.com/services/...
`EVENT_LOOKBACK_MINUTES`	Falco Rule Tunerがイベントを収集するために振り返るべき分数。 デフォルト: 60
`EVENT_COUNT_THRESHOLD`	チューニングが推奨されるイベントのしきい値の数。デフォルト: 5。 閾値を1に設定すると、すべてのポリシーイベントが偽陽性とみなされることになります。

Dockerコンテナとして実行します。

```
docker run -e SECURE_ENDPOINT=${SECURE_ENDPOINT} -e SECURE_TOKEN=${SECURE_TOKEN}
quay.io/sysdig/falco_rules_tuner
```

ターミナルウィンドウの出力には、調整する推奨ルールと、推奨/生成されたマクロとその条件が表示されます。

```
... <etc.>


# Change for rule: Write below root

- macro: elasticsearch-scripts_python_access_fileshost_exe_access_files

  condition: (container, image, repository endswith locationservices/elasticsearch-scripts
and proc.name=python and (fd.name startswith=/root/app/))
```

Slackチャンネルでの出力チェック（オプション）

ターミナルウィンドウで提供される出力には、推奨されるルール変更のみが含まれます。環境変数にSlackチャンネルのURLを指定すると、Tunerはイベントのサマリーと推奨ルールの変更の両方を出力します。

Policy Events Incoming Webhook APP 10:24 AM

Events Summary (6 policies triggered) | Customer: Test
Policy ID: 6833 (Change thread namespace) | Falco Rule: Change thread namespace
598 events generated in 5 minutes (Burst Ratio: 29.90)
across [0 containers | 4 hosts | 0 images | 1 processes | 0 file descriptors]
Processes: [{5 598}]

Policy ID: 6825 (Write below etc) | Falco Rule: Write below etc
330 events generated in 5 minutes (Burst Ratio: 4.40)
across [15 containers | 0 hosts | 10 images | 8 processes | 7 file descriptors]
Images: [{iopsnetwork/f5api-helloworld 296} {incomplete 8} {invoicecapture/textractor 8} {red-green/dns-failover-sidecar 7} {crusaders/quick-expense-service 6} {containerhosting/github-search 1} {incredibles/entity-management-service 1} {others 3}]
Processes: [{finish 296} {bash 23} {run.sh 4} {sed 3} {docker-entrypoi 1} {python 1} {sh 1} {others 1}]
Process + FD: [{finish /etc/service/ 296} {bash /etc/hosts 16} {bash /etc/resolv.conf 4} {bash /etc/profile.d/ 3} {run.sh /etc/resolv.conf.new 2} {sed /etc/unbound/ 2} {run.sh /etc/resolv.conf 2} {others 5}]
Image + Process + FD: [{iopsnetwork/f5api-helloworld++finish++/etc/service/ 296} {invoicecapture/textractor++bash++/etc/hosts 8} {incomplete++bash++/etc/hosts 8} {crusaders/quick-expense-service++bash++/etc/resolv.conf 3} {crusaders/quick-expense-service++bash++/etc/profile.d/ 3} {red-green/dns-failover-sidecar++run.sh++/etc/resolv.conf 2} {red-green/dns-failover-sidecar++sed++/etc/unbound/ 2} {others 8}]

10:24 AM

Suggesting changes in 4 falco rules

1. Run shell untrusted

```
- macro: sandbox_image
  condition: (container.image.repository endswith cse/sandbox)
```

2. Launch Privileged Container

```
- macro: hyperkube_image
  condition: (container.image.repository endswith coreos/hyperkube)
- macro: ceph_image
  condition: (container.image.repository endswith rook/ceph)
- macro: concourse_image
  condition: (container.image.repository endswith concourse/concourse)
```

3. Write below etc

```
- macro: textractor_bash_access_files
  condition: (container.image.repository endswith invoicecapture/textractor
and proc.name=bash
    and (fd.name startswith=/etc/hosts))
- macro: f5api-helloworld_finish_access_files
  condition: (container.image.repository endswith iopsnetwork/f5api-
helloworld and
    proc.name=finish and (fd.name startswith=/etc/service/))
- macro: incomplete_bash_access_files
  condition: (container.image.repository endswith incomplete and
proc.name=bash and
    (fd.name startswith=/etc/hosts))
```

4. Write below root

```
- macro: elasticsearch-scripts_python_access_files
  condition: (container.image.repository endswith
locationservices/elasticsearch-scripts
    and proc.name=python and (fd.name startswith=/root/app/))
```

推奨されるチューニングをルールに適用

レビュー：ルールエディタの使い方

チューナーは、過剰なアラート「ノイズ」を誘発している可能性のあるルールを検出し、そのノイズを軽減するようなコンテンツ関連のマクロやマクロ条件を提案します。

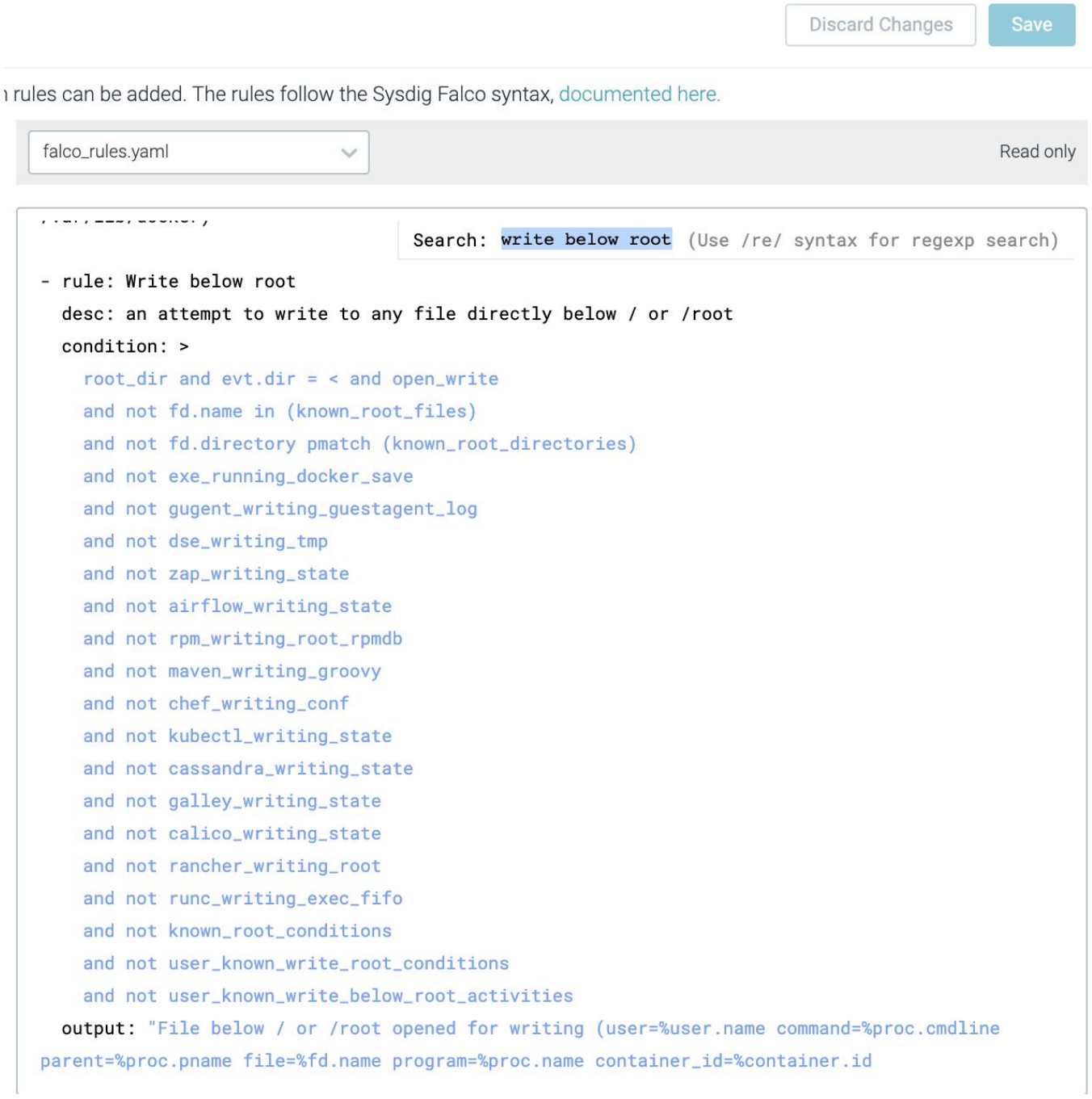
提案を実装するには、1) ルールの内容をルールエディタの左パネルに直接コピーして編集するか、2) そのルールのために作成された既存のプレースホルダマクロ（通常の形式：user_known_<ルール名>）を見つけて、そこに提案されたマクロと条件を追加します。

ルールの定義を直接編集すると、Sysdigのバージョンをアップグレードする際に上書き問題が発生する可能性があることに注意してください。カスタムルールを作成するか、user_known_knownプレースホルダを使用するのがより安全な手順です。

例えば、上の画像のTunerプロンプト4を実装することにしたとします。1つの方法としては、次のような方法があります。

1. falco_rules.yamlで「**Write below root**」を検索する。

ルール自体の両方を見つけることができます。



とプレースホルダマクロ、user_known_write_below_root_activities と user_known_write_below_root_conditions があります。どちらか一方を使用することができます。

```
# This is a placeholder for user to extend the whitelist for write below root rule
- macro: user_known_write_below_root_activities
  condition: (never_true)

- macro: runc_writing_exec_fifo
  condition: (proc.cmdline="runc:[1:CHILD] init" and fd.name=/exec.fifo)
```

2. エディタの左側のCustom Rulesパネルにプレースホルダを1つコピーします: user_known_write_below_root_activities .
3. チューナーが生成したマクロ(ここでは elasticsearch-scripts_python_access_files)と条件をカスタムルールパネルにコピーして、デフォルトの条件 never_true を上書きします。結果は以下のようになります。

```
# generated by tuner and copied to here (custom panel in the rules editor)
- macro: elasticsearch-xxx
  condition: (...)
- macro: user_known_write_below_root_acitivies
  condition: (elasticsearch-xxx) # updated from "never_true" with the generated macro
name
```

4. Save をクリックします。

この調整は、ポリシー内でWrite below rootルールが呼び出された場合に適用されます。

注意事項

これらの変更は、編集されたマクロ (user_known_write_below_root) が使用されている場所であればどこでも適用されます。マクロの中には、複数のルールや他のマクロに埋め込まれているものもあります。慎重に編集してください。