Secure Firewall Threat Defense 7.4でのAppID早期パケット検出の設定

内容

はじめに

このドキュメントでは、Cisco Secure Firewall 7.4でAppID早期パケット検出(EARLY PACKET DETECTION)を設定する方法について説明します。

背景 – 問題（お客様の要件）

- ディープパケットインスペクションによるアプリケーション検出では、トラフィックの識別に複数のパケットが必要になる場合があります。
- アプリケーションサーバのIPやポートが既知の場合、追加のパケットの検査を回避できる場合があります。

最新情報

- 新しいSnortベースのLua AppID APIが作成され、IPアドレス、ポート、およびプロトコルをそれぞれにマッピングできるようになりました。
  - アプリケーションプロトコル(service appid)、
  - クライアントアプリケーション(client appid)および
  - Webアプリケーション（ペイロードappid）。
- このアプリケーション検出用APIを使用して、FMC上にカスタムアプリケーションディテクタを作成できます。
- このディテクタがアクティブになると、この新しいAPIを使用して、セッションの最初のパケットでアプリケーションを特定できるようになります。

機能の概要

- APIは次のように識別されます。
  - **addHostFirstPktApp** （protocol_appId、client_appId、payload_appId、IPアドレス、ポート、プロトコル、再検査）

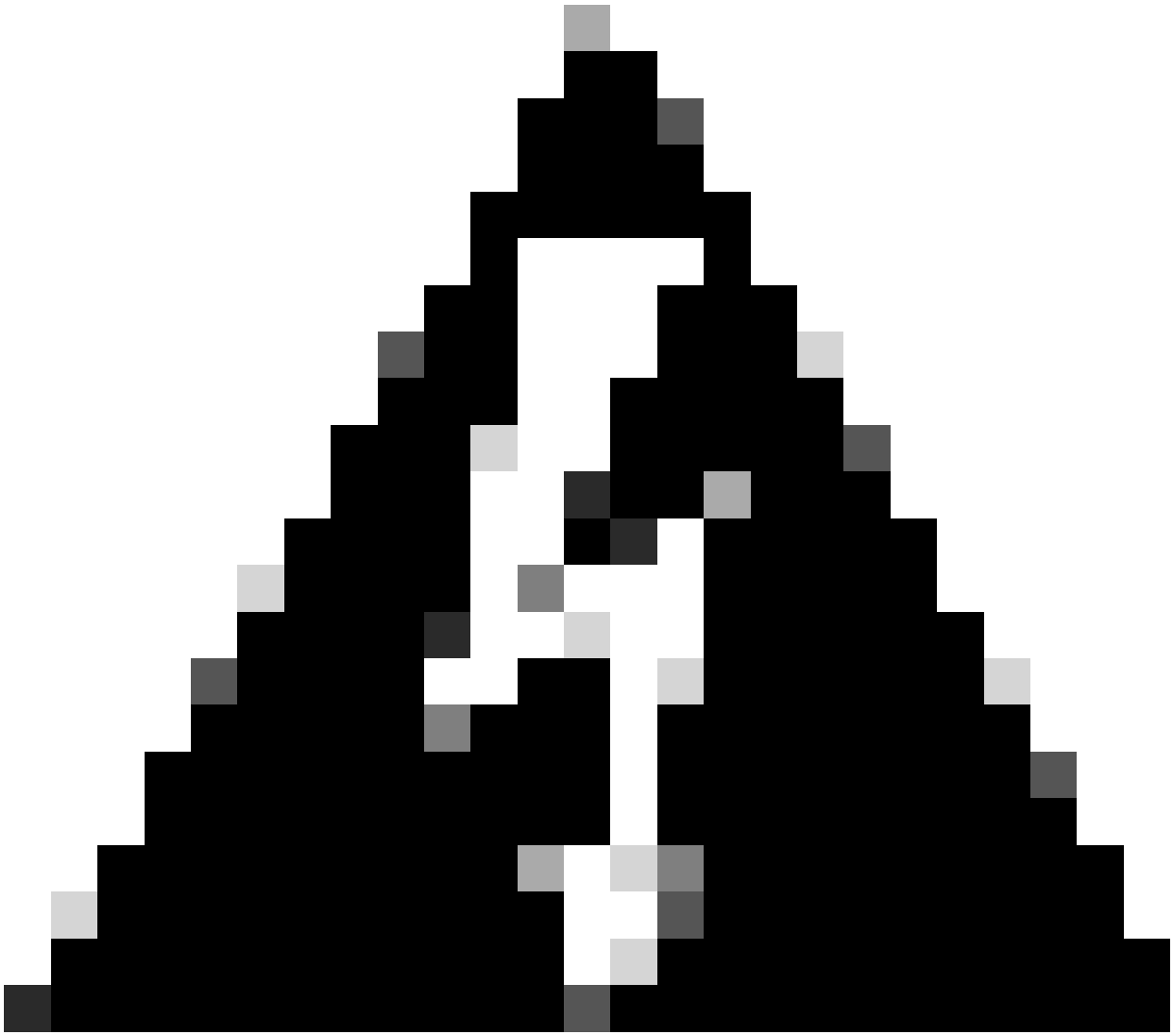
- キャッシュエントリは、カスタムアプリケーションディテクタで作成されたマッピングごとに作成されます。
- すべての着信セッションの最初のパケットが検査され、一致するものがキャッシュ内にあるかどうかが確認されます。
- 一致が見つかると、対応するappidをセッションに割り当て、アプリケーション検出プロセスが停止します。
- ユーザは、APIによって一致が検出された後でも、トラフィックを再検査するオプションを使用できます。
- reinspect引数は、最初のパケットで検出されたアプリケーションを再検査する必要があるかどうかを示すブール値です。
- 再検査がtrueの場合、APIが一致を見つけてもアプリケーション検出は続行されます。
- この場合、最初のパケットで割り当てられたAPPIDが変更される可能性があります。

前提条件、サポート対象プラットフォーム、ライセンス

最低限のソフトウェアおよびハードウェアプラットフォーム

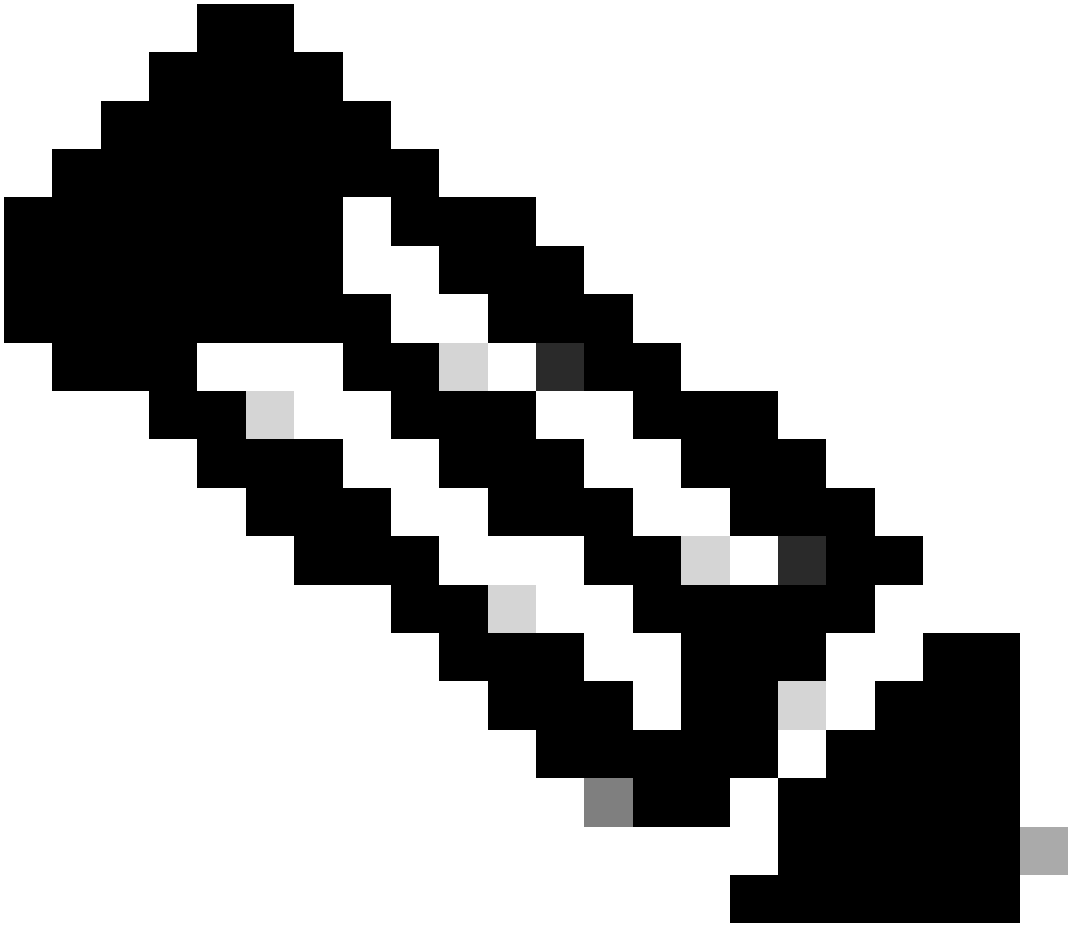
アプリケーションと最小バージョン	サポートされるマネージドプラットフォームとバージョン	マネージャ	注意事項

Cisco Secure Firewall 7.4 Snort3の使用	FTD 7.4をサポートするすべてのプラットフォーム	FMCオンプレミス+ FTD	これはデバイス側の機能です。FTDは7.4上にある必要があります。



警告: Snort 2はこのAPIをサポートしていません。

Snort 3、マルチインスタンス、およびHA/クラスタリングのサポート



注:Snort 3を検出エンジンにする必要があります。

FTD	
マルチインスタンスはサポートされていますか？	Yes
HAデバイスでサポート	Yes

クラスタデバイスでサポートされていますか。	Yes

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。
・ 7.4以降を実行するCisco Firepower Threat Defense

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

機能の詳細

機能の説明

今回のリリースとの対比

Secure Firewall 7.3以前	Secure Firewall 7.4の新機能
・ 既知のIP/ポート/プロトコルの組み合わせに対するアプリケーション検出は、他のすべてのアプリケーション検出メカニズムを使い果たした後に、フォールバックオプションとしてのみ使用できました。 ・ 基本的に、セッションの最初のパケットの検出はサポートされていませんでした。	・ 新しいlua検出APIは、他のアプリケーション検出メカニズムよりも先に評価されます。 ・ したがって、7.4では、セッションの最初のパケットでの検出をサポートしています。

仕組み

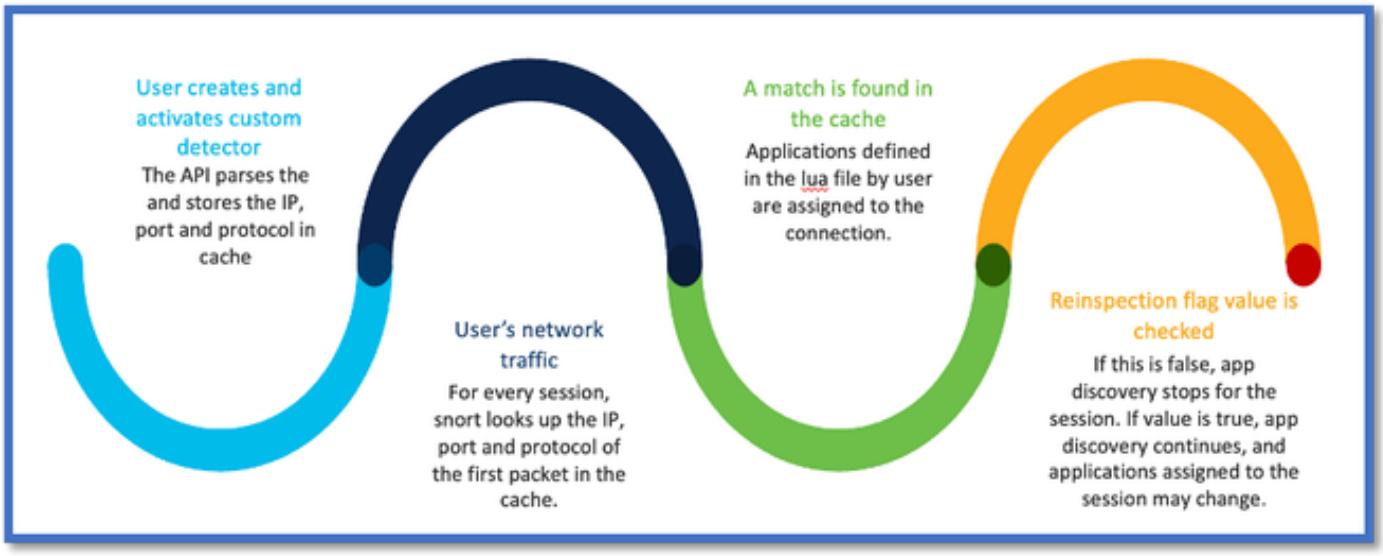
- luaファイルの作成：ファイルがluaテンプレート内にあることを確認します（構文エラーはありません）。また、ファイルのAPIに指定されている引数が正しいことも確認します。

- 新しいカスタム検出機能を作成する：FMCで新しいカスタム検出機能を作成し、その中にluaファイルをアップロードします。ディテクタを起動します。

- 実行トラフィック：カスタムアプリケーション検出で定義されているIP/ポート/プロトコルの組み合わせに一致するトラフィックをデバイスに送信します。

- 接続イベントのチェック：FMCで、IPおよびポートでフィルタリングされた接続イベントをチェックします。ユーザ定義のアプリケーションが特定されます。

AppID Early Packet Detection APIワークフロー



カスタム検出機能の例のAPIフィールドの説明

gDetector:addHostFirstPktApp

(gAppIdProto、gAppIdClient、gAppId、0, "192.0.2.1", 443, DC.ipproto.tcp);

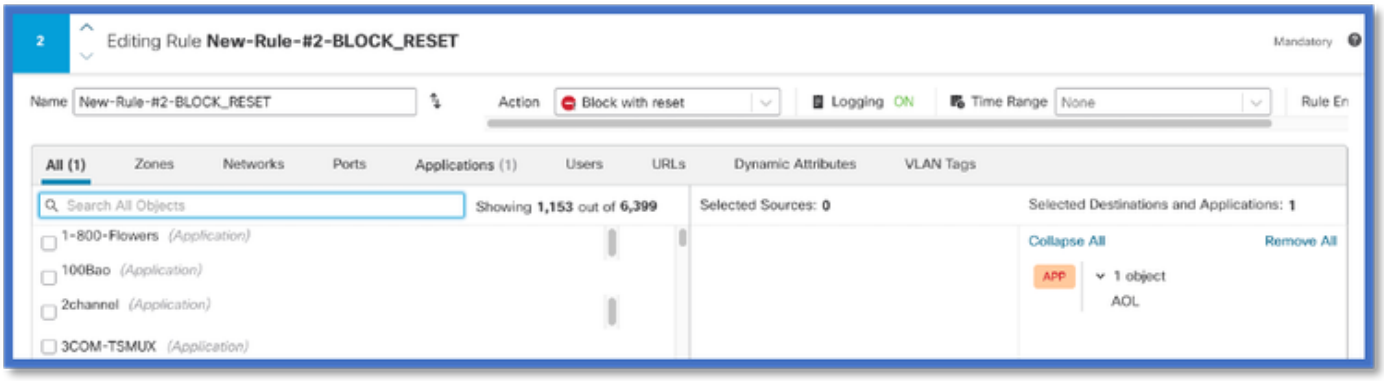
- 強調表示されている引数は、reinspectフラグ、IPアドレス、ポート、およびプロトコルに対するユーザ定義の値です。
- 0はワイルドカードを示します。

Arguments	説明	期待値
Reinspectフラグ	ユーザがIP/ポート/プロトコルに基づいてファイアウォールアクションを実行する代わりにトラフィックを検査する場合は、reinspectフラグ値を1に設定できます。	0 = reinspect disabledまたは 1 = reinspectが有効

IP アドレス	サーバのターゲットIP（サブネット内の単一または範囲のIP）。セッション内の最初のパケットの宛先IP。	192.168.4.198または 192.168.4.198/24または 2a03:2880:f103:83:face:b00c:0:25deまたは 2a03:2880:f103:83：面：b00c:0:25de/32
ポート	セッション内の最初のパケットの宛先ポート。	0 ~ 65535
プロトコル	ネットワークプロトコル	TCP/UDP/ICMP

使用例：トラフィックを迅速にブロックする方法

- ポリシービュー：アプリケーション「AOL」のブロックルール。



- curl https://www.example.com v/s curl https://192.0.2.1/（TESTのIPアドレスの1つ）を使用したcurlを使用したトラフィックのテスト

<#root>

```
> curl https://www.example.com/
```

```
curl: (35) OpenSSL SSL_connect: SSL_ERROR_SYSCALL in connection to www.example.com:443
```

```
> curl https://192.0.2.1/
```

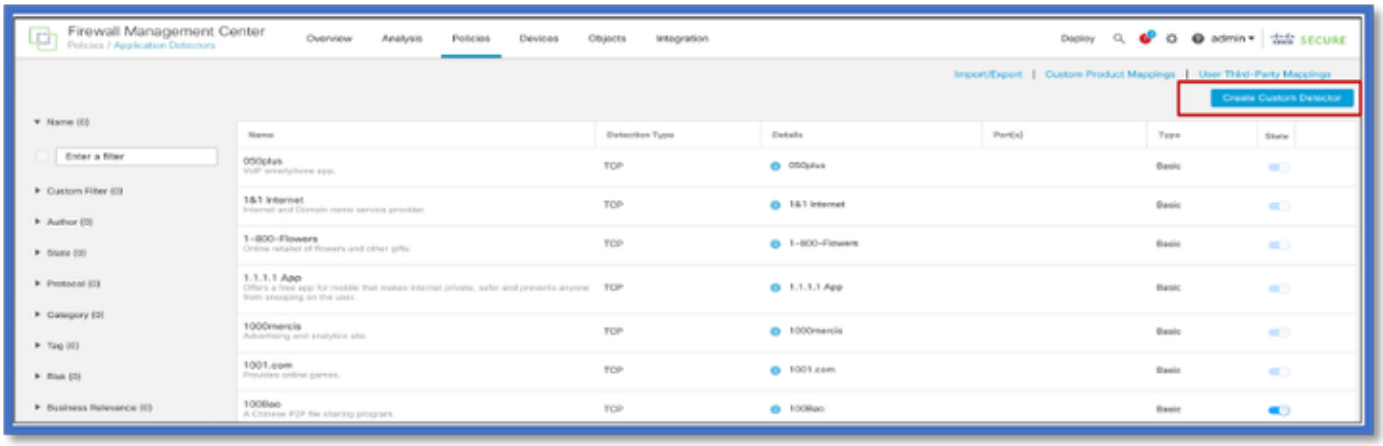
```
curl: (7) Failed to connect to 192.0.2.1 port 443: Connection refused
```

ファイアウォール管理センターのウォークスルー

APIを使用したカスタム検出機能の作成手順

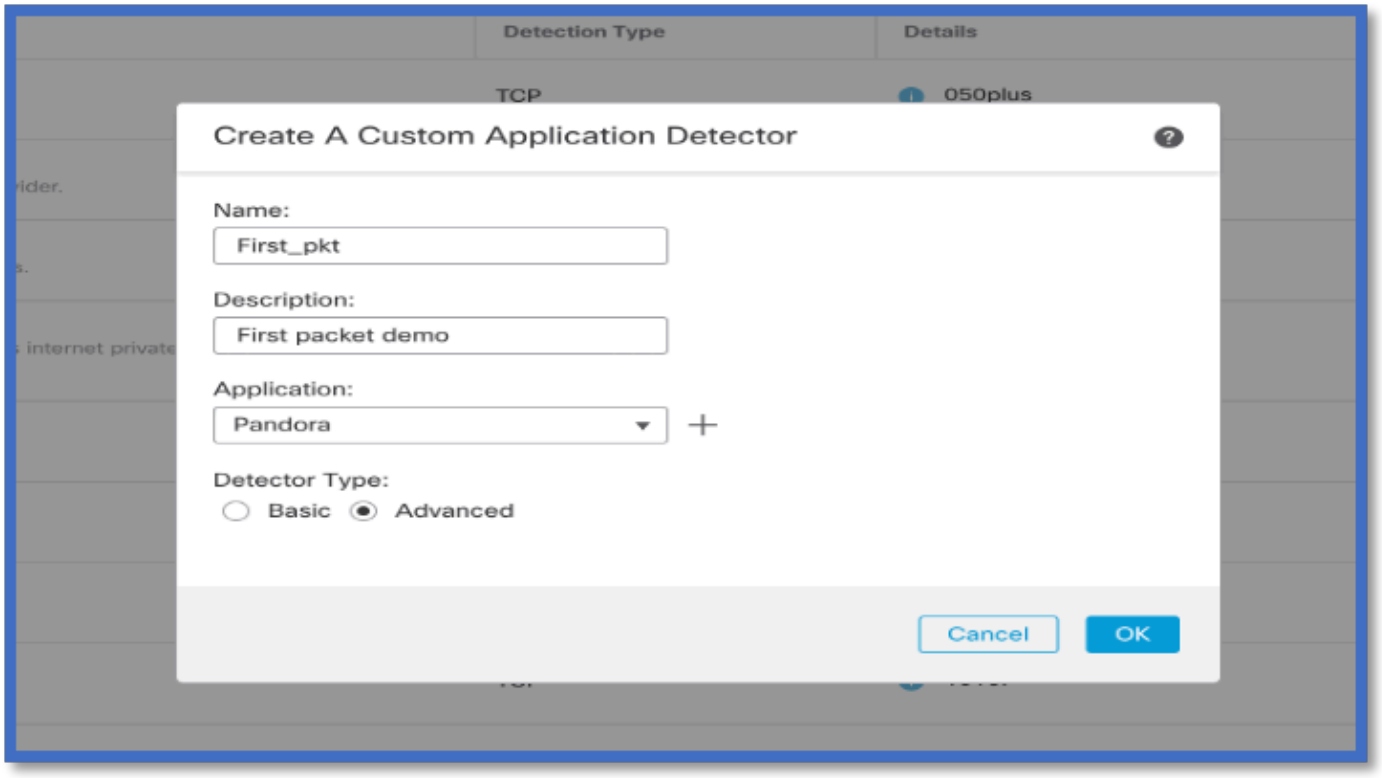
次の場所からFMC上に新しいカスタム検出機能を作成します。

- Policies > Application Detectors > Create Custom Detector .

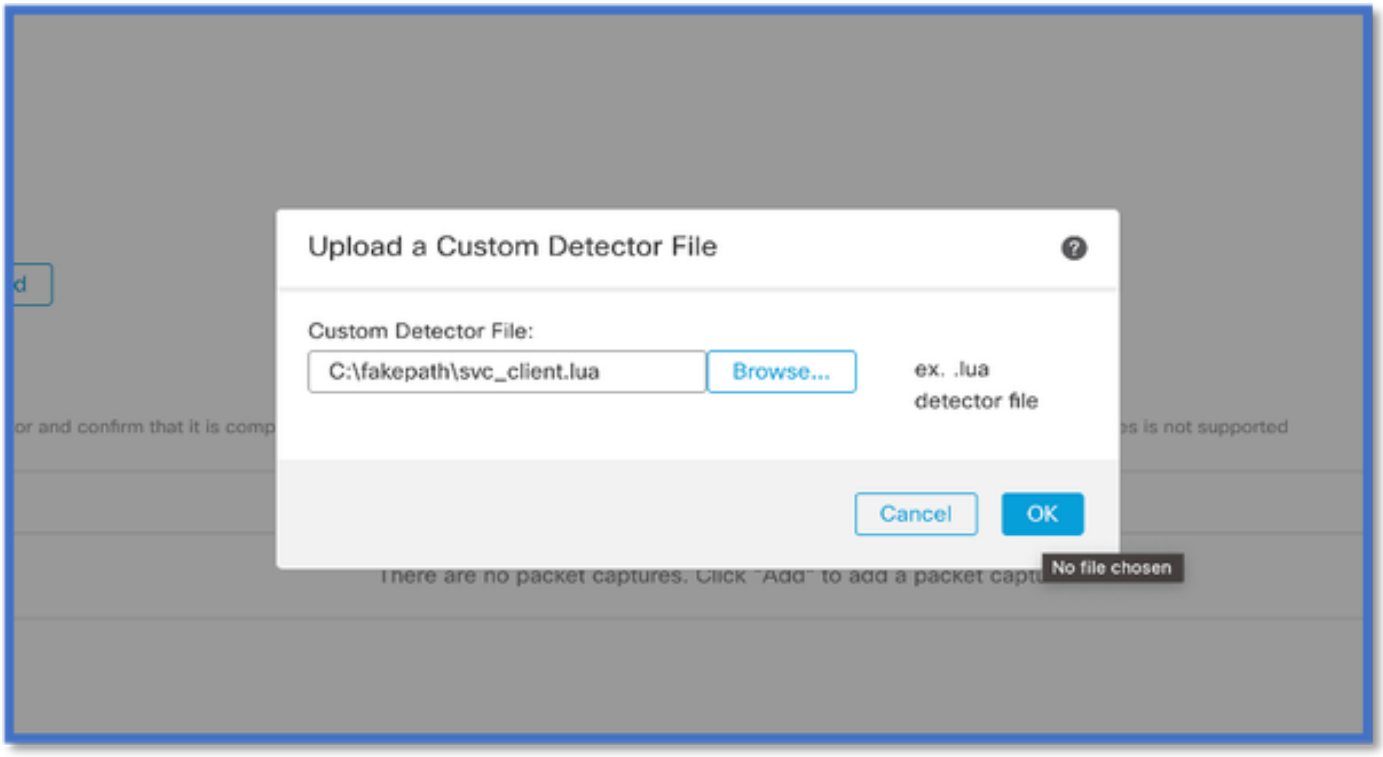


- 名前と説明を定義します。
  - ドロップダウンメニューからアプリケーションを選択します。

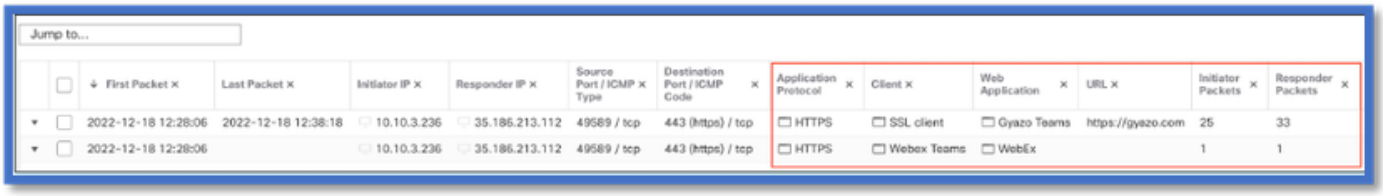
- [Advanced Detector Type]を選択します。



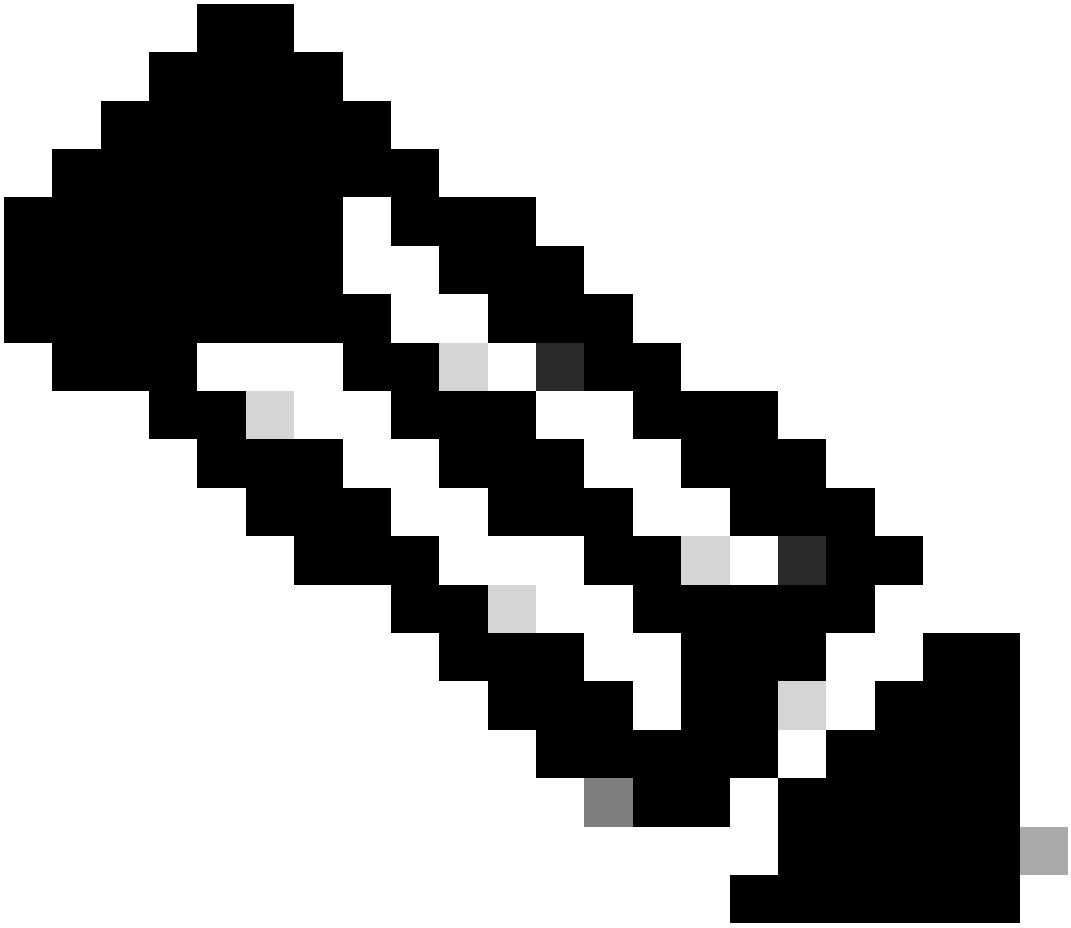
- Detection Criteriaの下にLuaファイルをアップロードします。ディテクタを保存してアクティブにします。



Reinspect有効v/s無効



- 2つのイベントは、再検査が有効な場合に、接続の開始と接続の終了を示します。



注：注意事項：

1.「HTTPS、WebEx、およびWebEx Teams」は、接続の開始時にAPIによって識別されます。再検査が当てはまるため、アプリの検出は継続し、アプリIDは「HTTPS、SSLクライアント、Gyazoチーム」に更新されます。

2. 発信側と応答側のパケットの数に注目します。通常のアプリケーション検出メソッドは、APIよりも多くのパケットを必要とします。

トラブルシューティング/診断

診断概要

- システムサポートアプリケーション識別デバッグに新しいログが追加され、第1パケット検出APIによってアプリケーションが検出されたかどうかが示されます。
- このログには、ユーザがトラフィックの再検査を選択したかどうかも示されます。
- ユーザがアップロードしたlua検出ファイルの内容は、FTDの/var/sf/appid/custom/lua/<UUID>で確認できます。
- luaファイル内のエラーは、ディテクタをアクティブ化する際に/var/log/messagesファイル内のFTDにダンプされます。

CLI:system support application-identification-debug

<#root>

192.0.2.1 443 -> 192.168.1.16 51251 6 AS=4 ID=0 New AppId session

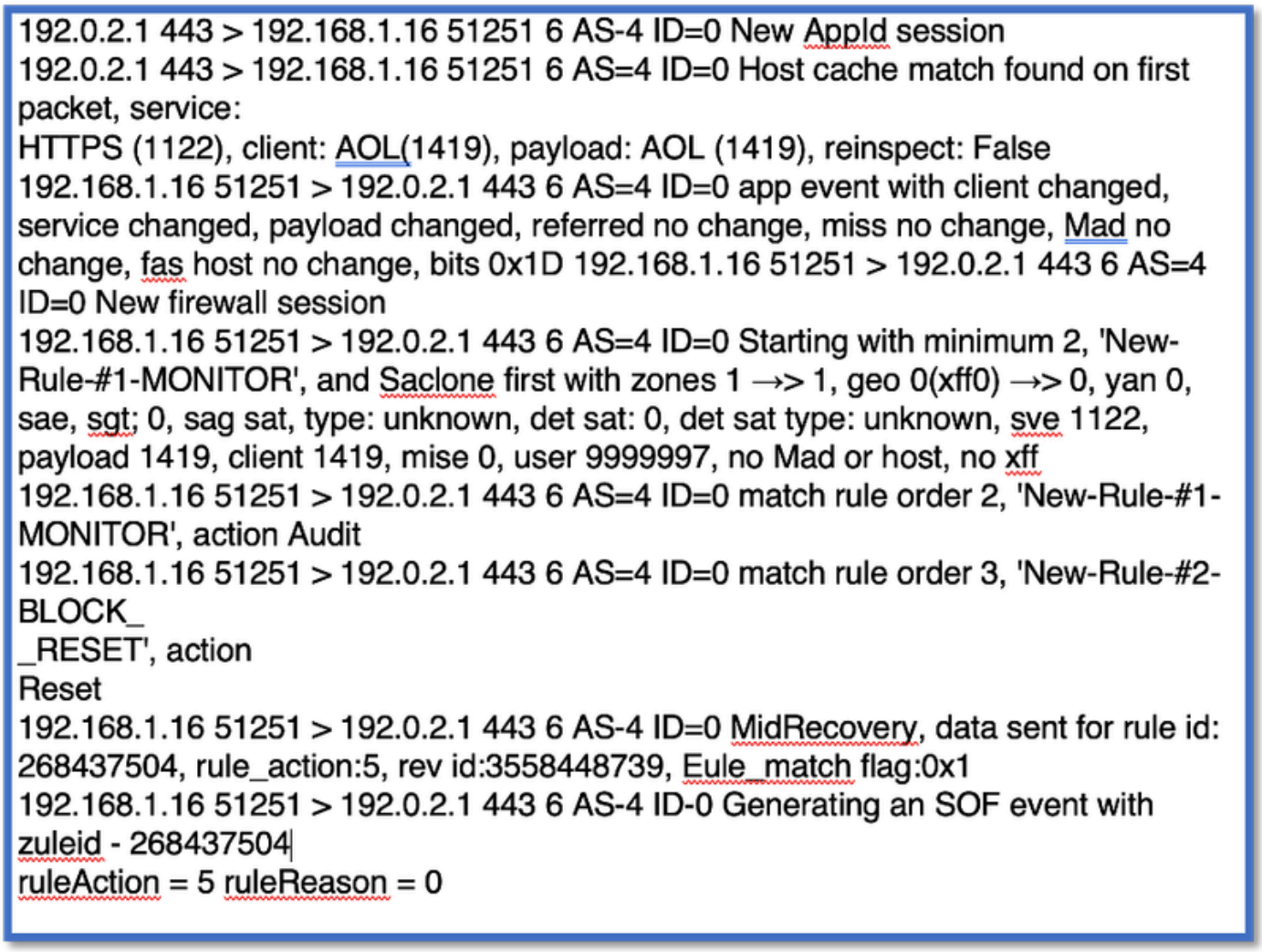
192.0.2.1 443 -> 192.168.1.16 51251 6 AS=4 ID=0 Host cache match found on first packet, service: HTTPS(1

192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 app event with client changed, service changed, payload
192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 New firewall session
192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 Starting with minimum 2, 'New-Rule-#1-MONITOR', and Src
192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 match rule order 2, 'New-Rule-#1-MONITOR', action Audit

192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 match rule order 3, 'New-Rule-#2-BLOCK_RESET', action Re

192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 MidRecovery data sent for rule id: 268437504, rule_acti
192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 Generating an SOF event with rule_id = 268437504 ruleAc

192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 reset action



AppID Lua Detectorコンテンツの場所

この新しいAPIのLua Detectorがデバイス/FTDに存在するかどうかを確認するには、次の2つのアプリケーション検出フォルダでaddHostFirstPktApp APIが使用されているかどうかを調べます。

1. VDB AppIDディテクタ – /var/sf/appid/odp/lua

2. カスタムディテクタ – /var/sf/appid/custom/lua

例：grep addHostFirstPktApp * in each folder.

問題例：

- 問題：カスタムLuaディテクタがFMCでアクティブになっていない。

確認する場所： /var/sf/appid/custom/lua/

期待される結果：FMCでアクティブ化されるカスタムアプリ検出機能ごとに1つのファイルがここで存在する必要があります。内容が、アップロードされたluaファイルと一致することを確認します。

- 問題：アップロードされたlua detectorファイルにエラーがあります。

チェックするファイル： /var/log/messages on FTD

エラーログ：

<#root>

Dec 18 14:17:49 intel-x86-64 SF-IMS[15741]:

Error - appid: can not set env of Lua detector /ngfw/var/sf/appid/custom/lua/6698fbd6-7ede-11ed-972c-d12

トラブルシューティングの手順

問題：ユーザ定義のIPアドレスとポートに送信されるトラフィックに対して、アプリケーションが正しく識別されない。

トラブルシューティングのステップ:

- FTDでluaディテクタが正しく定義され、アクティブになっていることを確認します。
  - FTDでluaファイルの内容を確認し、アクティブ化の際にエラーが表示されないことを確認します。
- トラフィックセッション内の最初のパケットの宛先IP、ポート、およびプロトコルを確認します。
  - これは、luaディテクタで定義されている値と一致する場合があります。

- system-support-application-identification-debugをチェックします。
    - 行を探します。Host cache match found on first packet. この行がない場合は、APIで一致が見つからなかったことを示します。

制限事項の詳細、一般的な問題、回避策

7.4では、APIを使用するためのUIはありません。UIのサポートは今後のリリースで追加される予定です。

改訂履歴

改訂	発行日	注釈
1.0	2024年7月18日	初版リリース

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。