Cisco Secure Web Appliance 向け AsyncOS API の概要

Cisco Secure Web Appliance 向け AsyncOS API（または AsyncOS API）は Representational State Transfer（REST）ベースの一連の操作で、Secure Web Appliance レポート、レポートカウンタ、トラッキングへのセキュアで認証済みのアクセスを提供します。この API を使用して、Secure Web Appliance のレポートおよびトラッキングデータを取得できます。このリリースでは、設定情報をクエリできます。

（注） シスコのコンテンツセキュリティ管理アプライアンスと REST API を使用して、Cisco Secure Web Appliance を設定できます。両方の方法を使用して Cisco Secure Web Appliance を設定すると、以前の方法で行った設定が上書きされます。

この章は、次の項で構成されています。

AsyncOS API 使用の前提条件

AsyncOS API を使用するには、次の知識が必要です。

- HTTP。API トランザクションに使用されるプロトコル。TLS 経由で保護された通信。
- JavaScript Object Notation（JSON）。API がリソースの表記作成に使用。
- JSON Web Token（JWT）。

- cURL など、HTTP や HTTPS を使用して AsyncOS API に対して要求の開始と応答の受信を行うクライアントまたはプログラミングライブラリ。クライアントまたはプログラミングライブラリは、APIからの応答を解釈できるようにJSONをサポートする必要があります。
- AsyncOS API へのアクセスの許可。認可 （5 ページ）を参照してください。
- Web インターフェイスまたは CLI を使用して有効化されている AsyncOS API。AsyncOS API の有効化 （2 ページ） を参照してください。

AsyncOS API の有効化

はじめる前に

CLI で `interfaceconfig` コマンドにアクセスできることを確認します。CLI へのアクセスが許可されるのは管理者、電子メール管理者、クラウド管理者、およびオペレータのみです。

CLI で `interfaceconfig` コマンドを使用すると、AsyncOS API を有効にできます。

ステップ 1 CLI にログインして `interfaceconfig` コマンドを実行します。

ステップ 2 編集するインターフェイスを選択します。

ステップ 3 AsyncOS API（モニタリング）HTTP を有効にするための次の質問に回答します。

- `Do you want to enable AsyncOS API (monitoring) HTTP on this interface? [Y]>` Y を入力します。
- `Which port do you want to use for AsyncOS API (monitoring) HTTP?[6080]>` デフォルトのポート 6080 か定義するポートを入力します。

ステップ 4 AsyncOS API（モニタリング）HTTPS を有効にするための次の質問に回答します。

- `Do you want to enable AsyncOS API (Monitoring) HTTPS on this interface? [Y]>` Y を入力します。
- `Which port do you want to use for AsyncOS API (Monitoring) HTTPS?[6443]>` デフォルトのポート 6443 か定義するポートを入力します。

（注） AsyncOS API は HTTP / 1.1 を使用して通信します。

HTTPS を選択して、セキュア通信用に独自の証明書を使用する場合は、AsyncOS API との安全な通信 （3 ページ） を参照してください。

（注） HTTPS は常に実稼働環境で使用することをお勧めします。API のトラブルシューティングおよびテストには、HTTP のみを使用します。

ステップ 5 変更を送信し、保存します。

AsyncOS API との安全な通信

独自の証明書を使用してセキュア HTTP 経由で AsyncOS API と通信できます。

（注） HTTPS およびセキュア通信用の独自の証明書を使用して Web インターフェイスをすでに起動している場合は、この手順を実行しないでください。AsyncOS API は、HTTPS 経由で通信するため Web インターフェイスと同じ証明書を使用します。

ステップ 1 CLI で `certconfig` コマンドを使用して証明書を設定します。手順については、ユーザー ガイドまたはオンライン ヘルプを参照してください。

ステップ 2 CLI で `interfaceconfig` コマンドを使用して、IP インターフェイスで使用する HTTPS 証明書を独自の証明書に変更します。手順については、ユーザー ガイドまたはオンライン ヘルプを参照してください。

ステップ 3 変更を送信し、保存します。

AsyncOS API の認証と認可

このセクションでは、認証方式、API にアクセスできるユーザーロール、ユーザーにアクセス可能な API をクエリする方法について説明します。

- 認証 （3 ページ）
- 認可 （5 ページ）

認証

次の 2 つのいずれかの方法を使用すると、API へのクエリを認証できます。

- Base64 エンコード形式で、API へのすべての要求と一緒に、 Secure Web Appliance のユーザー名とパスワードを送信します。
- ヘッダーにトークンキーを含む API 要求で JSON Web トークン（JWT）を使用します。

アプライアンスのユーザー非アクティブ タイムアウトの設定は、JWT の有効期間に適用されます。要求の認証ヘッダーに有効なクレデンシャルが含まれない場合、API は 401 エラーメッセージを送信します。base64 ライブラリを使用すると、クレデンシャルを base64 エンコード形式に変換できます。

JSON Web トークンを使用した API クエリの認証

JWT を生成し、API クエリで使用することができます。

（注） アプライアンスのユーザー非アクティブ タイムアウトの設定は、JWT の有効期間に適用されます。 Secure Web Appliance は、その有効期間の JWT を含むすべての API クエリをチェックします。JWT の有効期間が 5 分以内の場合、タイムアウトになると、新しい更新 JWT が応答ヘッダーと共に送信されます。API クエリでこの新しい更新 JWT を使用するか、新しい JWT を生成する必要があります。

概要	`POST /wsa/api/v2.0/login`
	二要素認証には、次の構文を使用します。
	`POST /wsa/api/v2.0/login/two_factor`	

本文パラメータ	Base64 エンコード クレデンシャルを使用します。 `{ "data": { "userName":"YWRtaW4=", "passphrase":"aXJvbnBvcnQ=" } }`	
要求ヘッダー		Host、Accept、Authorization
応答ヘッダー		Content-Type、Content-Length、Connection

次の例では、Base64 エンコード クレデンシャルでログインし、JWT を生成するクエリを示します。

サンプル リクエスト

```
POST /wsa/api/v2.0/login
HTTP/1.1
Content-Type: application/json
cache-control: no-cache
User-Agent: curl/7.54.0
Accept: */*
Host: wsa.cisco.com:6080
accept-encoding: gzip, deflate
content-length: 95
Connection: keep-alive
{
    "data":
    {
        "userName":"YWRtaW4=",
        "passphrase":"aXJvbnBvcnQ="
    }
}
```

サンプル応答

```
HTTP/1.1 200 OK
Server: API/2.0
Date: Mon, 26 Nov 2018 07:22:47 GMT
Content-type: application/json
```

```
Content-Length: 618
Connection: close
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: content-type, jwttoken, mid, h, email
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, POST, DELETE, OPTIONS
Access-Control-Expose-Headers: Content-Disposition, jwtToken

{
    "data": {
        "userName": "admin",
        "is2FactorRedirectRequired": "false",
        "role": "Administrator",
        "email": [],
       "jwtToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyTmFtZSI6ImFkbWluIiwiaXM

        yRmFjdG9yQ2hlY2tSZXF1aXJlZCI6ZmFsc2UsImNvb2tpZSI6IlRucEZOVTFFWTNwTlZFMDlDanRMYVR

        oeENqdFpiV1J6VFVSQk5VMURNWGRpTWxGMVdUSnNlbGt5T0hWWk1qbDBUMnBaZDA5RVFUMEtcbk8xVkh

        PWHBrUnpGb1lteEtNV0p1VW5CaVYxVjJUbmswTUV4cVFUMEtPMVJVUlhkTlJsazNUVlJKZFUxRE5IZE1

        WRWw1VFdwek1FMXFcblNUVlNhazVDVDBWRk1rOUVaM2xTUlVreVRYcGtSazFwTVVSTlZFMHpUbFZXUjA1

    }
}
```

認可

AsyncOS API はロール ベースのシステムで、API クエリの範囲はユーザーのロールごとに定義されています。次のロールを使用する Cisco Secure Web Appliance のユーザーは、AsyncOS API にアクセスできます。

- 管理者
- オペレータ
- 専門技術者
- 読み取り専用オペレータ
- ゲスト
- Web Administrator
- Web Policy Administrator
- URL フィルタリング管理者
- メール管理者
- ヘルプ デスク ユーザー

（注）
- 外部認証ユーザーは API にアクセスできます。
- また、管理者から委任されたカスタムロールも API にアクセスできます。
- 管理者権限を持つユーザーのみが、REST API を使用して設定を変更できます。オペレータや読み取り専用オペレータなどの他のすべてのユーザーは、これらの設定の表示のみが許可されます。

AsyncOS API の要求と応答

（注） API の完全なリストについては、『AsyncOS API - Addendum to the Getting Started Guide for Secure Web Appliance』を参照してください。

AsyncOS API 要求

API に対する要求には次の特性があります。

- 要求は HTTP または HTTPS 経由で送信されます。
- 各要求には、次の形式で有効な URI が含まれている必要があります。

```
http://{appliance}:{port}/wsa/api/v2.0/{resource}/{resource_attributes}
https://{appliance}:{port}/wsa/api/v2.0/{resource}/{resource_attributes}
```

引数の説明

- `{appliance}:{port}`

  FQDN またはアプライアンスの IP アドレスと、アプライアンスが待機する TCP ポート番号です。
- `{resource}`

  レポート、トラッキング、隔離、設定、他のカウンタなど、アクセスしようとするリソースです。
- `{resource_attributes}`

  期間など、リソースでサポートされている属性です。

- 各要求には、ユーザー クレデンシャルまたは有効な認証ヘッダーを含める必要があります。
- ヘッダーにトークンキーを含む API 要求で以前生成された JSON Web トークン（JWT）を使用します。詳細については、「JSON Web トークンを使用した API クエリの認証」を参照してください。

- 各要求には、承認を設定する必要があります。

  ```
  application/json
  ```

- HTTPS（独自の証明書を使用）経由で送信された要求には、CA 証明書を含める必要があります。たとえば、cURL の場合、API 要求で CA 証明書を次のように指定することができます。

  ```
  curl --cacert <ca_cert.crt> -u"username:password"
  https://<fqdn>:<port>/wsa/api/v2.0/{resource}/{resource_attributes}
  ```

（注） API 要求では、大文字と小文字が区別され、このマニュアルで示すように入力する必要があります。

AsyncOS API 応答

このセクションでは、応答の主要なコンポーネントとさまざまなHTTPエラーコードについて説明します。

- 応答の主要なコンポーネント （7 ページ）
- HTTP 応答コード （8 ページ）

応答の主要なコンポーネント

コンポーネント		値	説明
ステータス コードと理由		HTTP 応答コード（8 ページ）を参照してください。	HTTP 応答コードと理由。
メッセージ ヘッダー	Content-Type	`application/json`	メッセージ本文の形式を示す。
	Content-Length	適用対象外	オクテットによる応答本文の長さ。
	Connection	`close`	接続用のオプション。

コンポーネント	値	説明

メッセージ本文	適用対象外	メッセージ本文は Content-Type ヘッダーで定義された形式です。次に、メッセージ本文のコンポーネントを示します。
		1. URI。API への要求で指定した URI。
		例
		`:"/api/v2.0/config/"`
		2. カウンタ グループやカウンタ名
		例
		`reporting/mail_security_summary`
		3. クエリ パラメータ
		例
		`startDate=2017-01-30T00:00:00.000Z&endDate=2018-01-30T14:00:00.000Z`
		4. エラー（エラーイベントのみ）。このコンポーネントは、メッセージ、コード、および説明の 3 つのコンポーネントを示します。
		例
		`"error": {"message": "Unexpected attribute - starts_with.","code": "404", "explanation": "404 = Nothing matches the given URI."}`
		メッセージ本文に空のカッコ（{}）が含まれている場合、API がクエリに一致するレコードを見つけられなかったことを表します。
		（注） **totalCount** は、データセットで返されるデータオブジェクトの数です（UI にテーブル形式で表示される結果の場合）。他のクエリでは、デフォルトで -1 が返されます。

HTTP 応答コード

次に、AsyncOS API によって返される HTTP 応答コードのリストを示します。

- 200

- 202
- 300
- 301
- 307
- 400
- 401
- 403
- 404
- 406
- 413
- 414
- 500
- 501
- 503
- 505

これらの HTTP 応答コードの説明については、次の RFC を参照してください。

- RFC1945
- RFC7231

AsyncOS API 機能

AsyncOS API を使用すると、次のカテゴリの情報を取得できます。

- Web 用 API
- 汎用 API

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。