Red Hat JBoss Core Services 2.4.57

Red Hat JBoss Core Services ModSecurity ガイド

Red Hat JBoss ミドルウェア製品との使用

Last Updated: 2024-02-08

Red Hat JBoss Core Services 2.4.57 Red Hat JBoss Core Services ModSecurity ガイド

Red Hat JBoss ミドルウェア製品との使用

法律上の通知

概要

Red Hat JBoss Core Services ModSecurity モジュールを Web アプリケーションファイアウォールとして設定して使用します。

目次

RED HAT JBOSS CORE SERVICES ドキュメントへのフィードバック

エラーを報告したり、ドキュメントを改善したりするには、Red Hat Jira アカウントにログインし、課題を送信してください。Red Hat Jira アカウントをお持ちでない場合は、アカウントを作成するように求められます。

手順

1. **このリンクをクリック** してチケットを作成します。
2. **Summary** に課題の簡単な説明を入力します。
3. **Description** に課題や機能拡張の詳細な説明を入力します。問題があるドキュメントのセクションへの URL を含めてください。
4. **Submit** をクリックすると、課題が作成され、適切なドキュメントチームに転送されます。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

第1章 MODSECURITY モジュール

ModSecurity モジュールは、Web クライアントが Web サーバーアプリケーションに送信する HTTP トラフィックのフィルター、監視、ブロックに使用できる Web アプリケーションファイアウォール (WAF) です。通常のファイアウォールとは異なり、WAF はフィルターを使用して Apache HTTP Server アプリケーションと対話できるアプリケーションおよびユーザーを決定します。ModSecurity の有効性は、ModSecurity が設定可能な HTTP トラフィックのリアルタイム監視を実行して攻撃を即座に検出できるようにするユーザー定義のルールに依存しています。

注記

Red Hat JBoss Core Services ModSecurity Guide には、Red Hat JBoss Core Services 2.4.57 リリースで利用可能な ModSecurity バージョン 2.9 モジュールの情報と例が記載されています。ModSecurity の有効性は、ユーザーが生成したルールによって異なります。このドキュメントでは、ルールを作成および実装する方法について説明します。この文書では、使用するための一連のルールは提供されません。

第2章 RHEL での MODSECURITY の設定

Red Hat JBoss Core Services を Red Hat Enterprise Linux (RHEL) にインストールする場合は、Apache HTTP Server の Web アプリケーションファイアウォール (WAF) として機能するように ModSecurity モジュールを設定できます。

注記

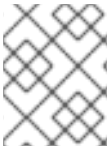
JBCS 2.4.57 は現在、RHEL 9 用の Apache HTTP サーバーのアーカイブファイルのディストリビューションを提供していません。

2.1. RHEL に対する MODSECURITY の依存関係

ModSecurity が正常に機能するには、いくつかの依存関係があります。これらの依存関係の一部は、Red Hat JBoss Core Services の一部としてすでに組み込まれています。

次の表に、ModSecurity の依存関係のリストを示します。

依存関係	RHEL 上の JBCS の一部ですか?
Apache ポータブルランタイム (APR)	はい
APR-Util	はい
mod_unique_id	はい
libcurl	はい
Perl-Compatible Regular Expressions (PCRE)	はい
libxml2	×



注記

RHEL では、Red Hat JBoss Core Services には、**libxml2** ライブラリーを除くこれらの依存関係がすべて含まれています。

2.2. RHEL への MODSECURITY のインストール

ModSecurity モジュールは、Red Hat JBoss Core Services インストールの一部として含まれています。

Red Hat JBoss Core Services Apache HTTP Server インストールガイド の手順に従って、オペレーティングシステム用の Apache HTTP Server をダウンロードしてインストールできます。

関連情報

- Red Hat JBoss Core Services Apache HTTP Server Installation Guide

2.3. MODSECURITY をロード中

LoadModule コマンドを使用して ModSecurity モジュールをロードできます。

手順

- ModSecurity モジュールをロードするには、次のコマンドを入力します。

```
LoadModule security2_module modules/mod_security2.so
```

2.4. RHEL での RULES ディレクトリーの設定

ModSecurity 機能では、システムが使用するルールを作成する必要があります。Apache HTTP Server は、事前設定された **mod_security.conf.sample** ファイルを **HTTPD_HOME/modsecurity.d** ディレクトリーに提供します。ModSecurity ルールを使用するには、環境に適した設定で **mod_security.conf.sample** ファイルを変更する必要があります。ModSecurity ルールは、**modsecurity.d** ディレクトリーまたは **modsecurity.d/activated_rules** サブディレクトリーに保存できます。

手順

1. **HTTPD_HOME/modsecurity.d** ディレクトリーに移動します。
2. **mod_security.conf.sample** ファイルの名前を **mod_security.conf** に変更します。

```
mv mod_security.conf.sample ./mod_security.conf
```

3. **mod_security.conf** ファイルを開き、ModSecurity ルールで使用するすべての設定ディレクティブのパラメーターを指定します。

2.5. 主要な MODSECURITY 設定オプション

主要な ModSecurity 設定オプションを使用して、正規表現のパフォーマンスを向上させ、ModSecurity 2.6 フェーズ 1 のフェーズ 2 フックへの移行を調査し、**.htaccess** ファイルで特定のディレクティブの使用を許可することができます。

enable-pcre-jit

Perl 互換正規表現 (PCRE) ライブラリー 8.20 以降でジャストインタイム (JIT) コンパイラーのサポートを有効にし、正規表現のパフォーマンスを向上させます。

enable-request-early

ModSecurity 2.6 のフェーズ 1 からフェーズ 2 フックへの移行のテストを有効にします。

enable-htaccess-config

AllowOverride Options が設定されている場合に、**.htaccess** ファイル内のディレクティブの使用を有効にします。

第3章 WINDOWS SERVER での MODSECURITY の設定

Windows Server に Red Hat JBoss Core Services をインストールする場合、Apache HTTP Server の Web アプリケーションファイアウォール (WAF) として機能するように ModSecurity モジュールを設定できます。

3.1. WINDOWS SERVER 上の MODSECURITY の依存関係

ModSecurity が正常に機能するには、いくつかの依存関係があります。これらの依存関係の一部は、Red Hat JBoss Core Services の一部としてすでに組み込まれています。

次の表に、ModSecurity の依存関係のリストを示します。

依存関係	Windows Server 上の JBCS の一部ですか?
Apache ポート可能なランタイム	はい
APR-Util	はい
mod_unique_id	はい
libcurl	はい
Perl-Compatible Regular Expressions (PCRE)	はい
libxml2	はい

注記

Windows Server では、Red Hat JBoss Core Services にはこれらの依存関係がすべて含まれています。

3.2. WINDOWS SERVER への MODSECURITY のインストール

ModSecurity モジュールは、Red Hat JBoss Core Services インストールの一部として含まれています。Apache HTTP Server は、Windows Server 上で ModSecurity を実行するために必要なアイテムの多くを提供します。ただし、ModSecurity が正しく機能するには、システムが特定の基準に準拠していることを確認する必要があります。

前提条件

- ソースからソフトウェアをビルドするフォルダーには、Apache HTTP サーバーのビルドに使用する Apache ソースと ModSecurity ソースの両方が含まれています。
  以下に例を示します。
  - Apache ソースは **C:\ sourceFolder\httpd-2.4.57** にあります
  - Apache は **C:\Apache2457** にインストールされています
  - ModSecurity ソースは **C:\ sourceFolder \mod_security** にあります

注記

この場合、**sourceFolder** は、プロジェクトと組み合わせて使用する汎用フォルダーです。

- ビルド環境は正しくセットアップされています。
以下に例を示します。
  - **PATH** 環境変数に、**vsvars32.bat** によって設定された Visual Studio 変数が含まれていること。
  - **PATH** 環境変数に **CMAKE** の **bin** フォルダーが含まれていること。
  - **C:\ sourceDirectory\httpd-2.4.57 にある Apache ソース** コードディレクトリーの環境変数を設定する。

手順

- Red Hat JBoss Core Services Apache HTTP Server インストールガイド の手順に従って、Apache HTTP Server をダウンロードし、**C:** ドライブの適切な場所にインストールします。

関連情報

- Red Hat JBoss Core Services Apache HTTP Server Installation Guide

3.3. WINDOWS SERVER での RULES フォルダーの設定

ModSecurity 機能では、システムが使用するルールを作成する必要があります。Apache HTTP Server は、事前設定された **mod_security.conf.sample** ファイルを **HTTPD_HOME/modsecurity.d** フォルダーに提供します。ModSecurity ルールを使用するには、環境に適した設定で **mod_security.conf.sample** ファイルを変更する必要があります。ModSecurity ルールは、**modsecurity.d** フォルダーまたは **modsecurity.d\activated_rules** サブフォルダーに保存できます。

手順

1. **HTTPD_HOME\modsecurity.d** フォルダーに移動します。
2. **mod_security.conf.sample** ファイルの名前を **mod_security.conf** に変更します。
3. **mod_security.conf** ファイルを開き、ModSecurity ルールで使用するすべての設定ディレクティブのパラメーターを指定します。

3.4. 主要な MODSECURITY 設定オプション

主要な ModSecurity 設定オプションを使用して、正規表現のパフォーマンスを向上させ、ModSecurity 2.6 フェーズ 1 のフェーズ 2 フックへの移行を調査し、**.htaccess** ファイルで特定のディレクティブの使用を許可することができます。

enable-pcre-jit

Perl 互換正規表現 (PCRE) ライブラリー 8.20 以降でジャストインタイム (JIT) コンパイラーのサポートを有効にし、正規表現のパフォーマンスを向上させます。

enable-request-early

ModSecurity 2.6 のフェーズ 1 からフェーズ 2 フックへの移行のテストを有効にします。

enable-htaccess-config

AllowOverride Options が設定されている場合に、**.htaccess** ファイル内のディレクティブの使用を有効にします。

第4章 MODSECURITY ルールの作成

ModSecurity は主にカスタムのユーザー定義ルールに基づいて機能します。これらのルールは、ModSecurity が実行するセキュリティーチェックの種類を決定します。

4.1. APACHE リクエストサイクルの MODSECURITY ルール

Apache リクエストサイクルの 5 つの ModSecurity 処理フェーズのいずれかにルールを適用できます。

要求ヘッダー

ルール構文で **REQUEST_HEADERS** 変数を指定して、ModSecurity ルールをこのフェーズに適用します。

要求の body

ルール構文で **REQUEST_BODY** 変数を指定して、ModSecurity ルールをこのフェーズに適用します。

応答ヘッダー

ルール構文で **RESPONSE_HEADERS** 変数を指定して、ModSecurity ルールをこのフェーズに適用します。

レスポンスのボディー

ルール構文で **RESPONSE_BODY** 変数を指定して、ModSecurity ルールをこのフェーズに適用します。

ロギング

ルール構文で **LOGGING** 変数を指定して、ModSecurity ルールをこのフェーズに適用します。

関連情報

- ModSecurity リファレンスマニュアル: 処理フェーズ

4.2. MODSECURITY ルールの構造

ModSecurity ルールは通常、次の 4 つの主要な部分で設定されます。

- 設定ディレクティブ
- 1 つ以上の変数
- 1 人以上の演算子
- 1 つ以上のアクション

4.3. MODSECURITY 設定ディレクティブ

ModSecurity ルールは設定ディレクティブで始まります。ModSecurity の設定ディレクティブは、Apache HTTP Server ディレクティブと似ています。ほとんどの ModSecurity ディレクティブは、さまざまな Apache スコープディレクティブ内で使用できます。ただし、一部の ModSecurity ディレクティブはメイン設定ファイルで 1 回のみ使用できます。

これらのルールとコアルールファイルは、**httpd.conf** ファイルの外部に保存する必要があります。Apache **Include** ディレクティブを使用して、これらのルールを呼び出すことができます。これにより、ルールのアップグレードと移行が容易になります。

関連情報

- ModSecurity リファレンスマニュアル: 設定ディレクティブ

4.4. 単純な MODSECURITY ルールの例

たとえば、リクエストの URI 部分が特定の小文字の値と等しいかどうかを確認するために、次の単純な ModSecurity ルールを定義できます。

SecRule REQUEST_URI "@streq /index.php" "id:1,phase:1,t:lowercase,deny"

前述の ModSecurity ルールは次のコンポーネントで設定されます。

SecRule
指定された演算子を使用して指定された変数を分析するルールを作成する **設定ディレクティブ**

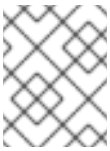
注記

ほとんどの ModSecurity ルールは、この設定ディレクティブを使用します。

REQUEST_URI
クエリー文字列データを含む完全なリクエスト URL を保持する **変数**

"@streq /index.php"
@streq が **/index.php** と等しい文字列値をチェックする **演算子**

"id:1,phase:1,t:lowercase,deny"
ルールが実行する Actions または transformations



注記

ルールは、先行する演算子の命令を実装する前に、最初に **lowercase** アクションを実行します。

前の例に基づいて、Apache 要求サイクルのフェーズ 1 中に、ルールは HTTP 要求の URI 部分を取得し、値を小文字に変換します。次に、ルールは、変換された値が **/index.php** と等しいかどうかをチェックします。値が **/index.php** と等しい場合、ModSecurity は要求を拒否し、それ以上のルールを処理しません。

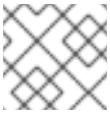
4.5. 複雑な MODSECURITY ルールの例

たとえば、リクエストによって履歴が変更されたかどうかを確認するために、次の複雑な ModSecurity ルールを定義できます。

SecRule REQUEST_URI|REQUEST_BODY|REQUEST_HEADERS_NAMES|REQUEST_HEADERS "history.pushstate|history.replacestate" "phase:4,deny,log,msg:'history-based attacks detected'"

前述の ModSecurity ルールは次のコンポーネントで設定されます。

SecRule
指定された演算子を使用して指定された変数を分析するルールを作成する **設定ディレクティブ**



注記

ほとんどの ModSecurity ルールは、この設定ディレクティブを使用します。

`REQUEST_URI|REQUEST_BODY|REQUEST_HEADERS_NAMES|REQUEST_HEADERS `

ルールがチェックするリクエストのさまざまな部分を定義する **変数** のパイプ区切りのリスト

"history.pushstate|history.replacestate"

JavaScript の **history.pushstate()** メソッドと **history.replacestate()** メソッドをチェックする、パイプで区切られた **演算子** のペア

"phase:4,deny,log,msg:'history-based attacks detected'"

指定された演算子の値が見つかった場合にルールが実行する **アクション** または **変換**

前の例に基づいて、Apache リクエストサイクルのフェーズ 4 中に、ルールはリクエストサイクルのさまざまな部分の **history.pushstate()** メソッドと **history.replacestate()** メソッドをチェックします。ルールがリクエスト URL 文字列、リクエスト本文、リクエストヘッダー名、またはリクエストヘッダーでこれらのメソッドを見つけた場合、ルールは次のアクションを実行します。

- **deny**
  ルール処理を停止し、トランザクションをインターセプトします
- **log**
  ルールの一致が成功したことを Apache エラーログファイルと ModSecurity 監査ログに記録します。
- **msg**
  ログで **history-based attacks detected** として定義されたメッセージを出力します

4.6. 関連情報 (または次の手順)

- ModSecurity リファレンスマニュアル: アクション
- ModSecurity リファレンスマニュアル: 設定ディレクティブ
- ModSecurity Reference Manual: Operators
- ModSecurity リファレンスマニュアル: トランスフォーメーション機能
- ModSecurity リファレンス: 変数