Red Hat JBoss Core Services 2.4.57

Red Hat JBoss 核心服务 ModSecurity 指南

用于 Red Hat JBoss Middleware 产品。

Last Updated: 2024-04-12

Red Hat JBoss Core Services 2.4.57 Red Hat JBoss 核心服务 ModSecurity 指南

用于 Red Hat JBoss Middleware 产品。

法律通告

摘要

配置和使用 Red Hat JBoss Core Services ModSecurity 模块作为 Web 应用程序防火墙。

目录

提供有关 RED HAT JBOSS CORE SERVICES 文档的反馈

要报告错误或改进文档，请登录到 Red Hat JIRA 帐户并提交问题。如果您没有 Red Hat Jira 帐户，则会提示您创建一个帐户。

流程

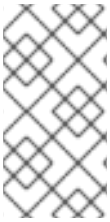
1. 单击以下链接 **以创建 ticket**。
2. 在 **Summary** 中输入问题的简短描述。
3. 在 **Description** 中提供问题或功能增强的详细描述。包括一个指向文档中问题的 URL。
4. 点 **Submit** 创建问题，并将问题路由到适当的文档团队。

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 CTO Chris Wright 的信息。

第 1 章 MODSECURITY 模块

ModSecurity 模块是一个 Web 应用程序防火墙(WAF)，您可以使用它来过滤、监控和阻止 Web 客户端发送到 Web 服务器应用程序的 HTTP 流量。与常规防火墙不同，WAF 使用过滤器来确定哪些应用程序和用户可以与 Apache HTTP 服务器应用程序交互。ModSecurity 的有效性依赖于用户定义的规则，使 ModSecurity 能够对 HTTP 流量进行可配置和实时监控，以立即检测攻击。



注意

Red Hat JBoss Core Services ModSecurity Guide 提供了 Red Hat JBoss Core Services 2.4.57 发行版本提供的 ModSecurity 版本 2.9 模块的信息和示例。ModSecurity 的有效性取决于用户生成的规则。本文档论述了如何创建和实施规则。本文档不提供一组要使用的规则。

第 2 章 在 RHEL 中配置 MODSECURITY

当您在 Red Hat Enterprise Linux (RHEL)上安装 Red Hat JBoss Core Services 时，您可以将 ModSecurity 模块配置为充当 Apache HTTP 服务器的 web 应用程序防火墙(WAF)。

注意

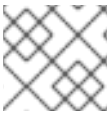
JBCS 2.4.57 目前不为 RHEL 9 提供 Apache HTTP 服务器的存档文件分发。

2.1. RHEL 上的 MODSECURITY 依赖项

ModSecurity 有几个依赖项才能成功运行。其中一些依赖项已作为 Red Hat JBoss Core Services 的一部分包含在内。

下表提供了 ModSecurity 依赖项列表：

依赖项	RHEL 上的 JBCS 的一部分？
Apache Portable Runtimes (APR)	是
aPR-Util	是
mod_unique_id	是
libcurl	是
Perl-Compatible Regular Expressions (PCRE)	是
libxml2	否



注意

在 RHEL 上，Red Hat JBoss Core Services 包括了所有这些依赖项，但 **libxml2** 库除外。

2.2. RHEL 上的 MODSECURITY 安装

ModSecurity 模块包含在 Red Hat JBoss Core Services 安装中。

您可以按照 Red Hat JBoss Core Services Apache HTTP Server 安装指南中的步骤 为您的操作系统下载并安装 Apache HTTP 服务器。

其他资源

- Red Hat JBoss Core Services Apache HTTP Server 安装指南

2.3. 加载 MODSECURITY

您可以使用 **LoadModule** 命令加载 ModSecurity 模块。

流程

- 要载入 ModSecurity 模块，请输入以下命令：

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

2.4. 在 RHEL 上配置规则目录

ModSecurity 功能要求您创建系统使用的规则。Apache HTTP 服务器在 ***HTTPD_HOME*/modsecurity.d 目录中提供预配置的 mod_security.conf. sample** 文件。要使用 ModSecurity 规则，您必须使用适合您环境的设置修改 **mod_security.conf.sample** 文件。您可以将 ModSecurity 规则存储在 **modsecurity.d** 目录或 **modsecurity.d/activated_rules** 子目录中。

流程

1. 进入 ***HTTPD_HOME*/modsecurity.d** 目录。
2. 将 **mod_security.conf.sample** 文件重命名为 **mod_security.conf** ：

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

3. 打开 **mod_security.conf** 文件，并为您要与 ModSecurity 规则一起使用的所有配置指令指定参数。

2.5. 密钥 MODSECURITY 配置选项

您可以使用键 ModSecurity 配置选项提高正则表达式的性能，调查 ModSecurity 2.6 阶段，进入阶段两个 hook，并允许使用 **.htaccess** 文件中的某些指令。

enable-pcre-jit

在 Perl-Compatible Regular Expressions (PCRE)库 8.20 或更高版本中启用 Just-In-Time (JIT)编译器支持，以提高正则表达式的性能。

enable-request-early

启用测试 ModSecurity 2.6 将从阶段移到阶段 2 hook

enable-htaccess-config

当设置了 **AllowOverride Options** 时，启用使用 **.htaccess** 文件中的指令

第 3 章 在 WINDOWS 服务器上配置 MODSECURITY

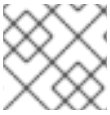
在 Windows Server 上安装 Red Hat JBoss Core Services 时，您可以将 ModSecurity 模块配置为充当 Apache HTTP 服务器的 Web 应用程序防火墙(WAF)。

3.1. WINDOWS SERVER 上的 MODSECURITY 依赖项

ModSecurity 有几个依赖项才能成功运行。其中一些依赖项已作为 Red Hat JBoss Core Services 的一部分包含在内。

下表提供了 ModSecurity 依赖项列表：

依赖项	Windows Server 中的 JBCS 的一部分？
Apache Portable Runtimes	是
aPR-Util	是
mod_unique_id	是
libcurl	是
Perl-Compatible Regular Expressions (PCRE)	是
libxml2	是



注意

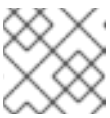
在 Windows Server 上，红帽 JBoss 核心服务包括所有这些依赖项。

3.2. 在 WINDOWS SERVER 上安装 MODSECURITY

ModSecurity 模块包含在 Red Hat JBoss Core Services 安装中。Apache HTTP 服务器提供在 Windows Server 上运行 ModSecurity 所需的许多项目。但是，您必须确保您的系统符合某些条件，以允许 ModSecurity 正常工作。

先决条件

- 您从源构建软件的文件夹包含用于构建 Apache HTTP 服务器的 Apache 源和 ModSecurity 源。例如：
  - Apache 源位于 **C:\ *sourceFolder*\httpd-2.4.57**
  - Apache 已安装 **C:\Apache2457**
  - ModSecurity 源位于 **C:\ *sourceFolder*\mod_security**



注意

在本例中，***sourceFolder*** 是您将与项目结合使用的通用文件夹。

- 您的构建环境会被正确设置。
  例如：
  - 确保 ***PATH*** 环境变量包含由 **vsvars32.bat** 设置的 Visual Studio 变量。
  - 确保 ***PATH*** 环境变量包含 **CMAKE** 的 **bin** 文件夹。
  - 为 Apache 源代码目录设置环境变量，它位于 **C:\ *sourceDirectory*\httpd-2.4.57**。

流程

- 按照 Red Hat JBoss Core Services Apache HTTP Server 安装指南中的步骤，将 Apache HTTP 服务器下载并安装到 **C:** 驱动器中的相应位置。

其他资源

- Red Hat JBoss Core Services Apache HTTP Server 安装指南

3.3. 在 WINDOWS 服务器上配置规则文件夹

ModSecurity 功能要求您创建系统使用的规则。Apache HTTP 服务器在 ***HTTPD_HOME*\modsecurity.d 文件夹中提供预配置的 mod_security.conf. sample** 文件。要使用 ModSecurity 规则，您必须使用适合您环境的设置修改 **mod_security.conf.sample** 文件。您可以将 ModSecurity 规则存储在 **modsecurity.d** 文件夹或 **modsecurity.d\activated_rules** 子文件夹中。

流程

1. 进入 ***HTTPD_HOME*\modsecurity.d** 文件夹。
2. 将 **mod_security.conf.sample** 文件重命名为 **mod_security.conf**。
3. 打开 **mod_security.conf** 文件，并为您要与 ModSecurity 规则一起使用的所有配置指令指定参数。

3.4. 密钥 MODSECURITY 配置选项

您可以使用键 ModSecurity 配置选项提高正则表达式的性能，调查 ModSecurity 2.6 阶段，进入阶段两个 hook，并允许使用 **.htaccess** 文件中的某些指令。

enable-pcre-jit

在 Perl-Compatible Regular Expressions (PCRE)库 8.20 或更高版本中启用 Just-In-Time (JIT)编译器支持，以提高正则表达式的性能。

enable-request-early

启用测试 ModSecurity 2.6 将从阶段移到阶段 2 hook

enable-htaccess-config

当设置了 **AllowOverride Options** 时，启用使用 **.htaccess** 文件中的指令

第 4 章 创建 MODSECURITY 规则

ModSecurity 主要基于自定义用户定义的规则功能。这些规则决定了 ModSecurity 执行的安全检查类型。

4.1. APACHE 请求周期中的 MODSECURITY 规则

您可以将规则应用到 Apache 请求周期的五个 ModSecurity 处理阶段：

请求的标头（header）

通过在规则语法中指定 **REQUEST_HEADERS** 变量，将 ModSecurity 规则应用到此阶段。

请求正文

通过在规则语法中指定 **REQUEST_BODY** 变量，将 ModSecurity 规则应用到此阶段。

响应标头

通过在规则语法中指定 **RESPONSE_HEADERS** 变量，将 ModSecurity 规则应用到此阶段。

响应正文

通过在规则语法中指定 **RESPONSE_BODY** 变量，将 ModSecurity 规则应用到此阶段。

日志记录

通过在规则语法中指定 **LOGGING** 变量，将 ModSecurity 规则应用到此阶段。

其他资源

- ModSecurity 参考手册：处理阶段

4.2. MODSECURITY 规则的结构

ModSecurity 规则通常由四个主要部分组成：

- 配置指令
- 一个或多个变量
- 一个或多个 operator
- 一个或多个操作

4.3. MODSECURITY 配置指令

ModSecurity 规则以配置指令开头。ModSecurity 的配置指令与 Apache HTTP Server 指令类似。您可以在各种 Apache 范围指令中使用大多数 ModSecurity 指令。但是，您可能仅在主配置文件中使用一些 ModSecurity 指令一次。

您必须在 **httpd.conf** 文件外存储这些规则和核心规则文件。您可以使用 Apache **Include** 指令来调用这些规则。这有助于规则的升级和迁移。

其他资源

- ModSecurity 参考手册：配置指令

4.4. 简单的 MODSECURITY 规则示例

您可以定义以下简单 ModSecurity 规则，例如，检查请求的 URI 部分是否等于特定的小写值：

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

前面的 ModSecurity 规则由以下复杂度组成：

SecRule

一个 *配置指令*，它会创建一个规则来利用指定的 Operator 分析指定的变量

注意

大多数 ModSecurity 规则使用此配置指令。

REQUEST_URI

包含完整请求 URL *的变量*，包括查询字符串数据

"@streq /index.php"

一个 *operator*，其中 **@streq** 检查等于 **/index.php**的字符串值

"id:1,phase:1,t:lowercase,deny"

规则 *执行的操作* 或 *转换*

注意

规则在规则实施前面的 operator 指令前先执行 **小写** 操作。

根据前面的示例，在 Apache 请求周期的第 1 阶段，该规则获取 HTTP 请求的 URI 部分，并将值转换为小写。然后，该规则会检查转换的值是否等于 **/index.php**。如果值等于 **/index.php**，ModSecurity 会拒绝请求，且不会处理任何进一步的规则。

4.5. 复杂 MODSECURITY 规则的示例

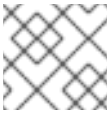
您可以定义以下复杂的 ModSecurity 规则，例如，检查请求是否已更改历史记录：

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

前面的 ModSecurity 规则由以下组件组成：

SecRule

一个 *配置指令*，它会创建一个规则来利用指定的 Operator 分析指定的变量



注意

大多数 ModSecurity 规则使用此配置指令。

`REQUEST_URI|REQUEST_BODY|REQUEST_HEADERS_NAMES|REQUEST_HEADERS `

由管道分隔 *的变量列表*，用于定义规则检查的请求的不同部分

"history.pushstate|history.replacestate"

检查 JavaScript **history.pushstate （）** 和 **history.replacestate （）** 方法的以管道分隔的 *运算符* 对

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

如果找到指定的 operator 值，则规则 *执行的操作* 或 *转换*

根据前面的示例，在 Apache 请求周期的第 4 阶段，规则检查 **history.pushstate （）** 和 **history.replacestate （）** 方法的请求周期的不同部分。如果规则在请求 URL 字符串、请求正文、请求标头名称或请求标头中找到这些方法，规则将执行以下操作：

- **deny**
  停止规则处理并截获事务
- **log**
  将规则成功与 Apache 错误日志文件和 ModSecurity 审计日志匹配
- **msg**
  输出在日志中 **检测到的基于历史记录的攻击** 的消息

4.6. 其他资源（或后续步骤）

- ModSecurity Reference Manual: Actions
- ModSecurity 参考手册：配置指令
- ModSecurity 参考手册：Operator
- ModSecurity 参考手册：转换功能
- ModSecurity 参考手册：变量