ModSecurity Configuration for Tiki

1. Introduction

ModSecurity is a powerful, open-source web application firewall (WAF) module that enhances security by protecting **web applications, including Tiki sites, from a wide range of threats** such as **SQL injection, cross-site scripting (XSS), and malicious bots attempting to scrape content or exploit vulnerabilities**. It operates based on predefined rules to filter and block potentially harmful requests. This guide provides a comprehensive walkthrough for setting up and configuring ModSecurity, ensuring **optimal security while preserving Tiki's usability and functionality**.

2. Installation

Step 1: Install ModSecurity

For Apache (Debian/Ubuntu)

sudo apt update sudo apt install libapache2-mod-security2

Step 2: Enable ModSecurity

Enable ModSecurity by copying the recommended configuration file:

```
sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
```

Then, **edit the file**:

```
sudo nano /etc/modsecurity/modsecurity.conf
```

Find:

apache

```
SecRuleEngine DetectionOnly
```

Change it to:

apache

```
SecRuleEngine On
```

Save and close the file.

Step 3: Verify Installation

Check if ModSecurity is enabled:

```
sudo apachectl -M | grep security2
```

Expected output:

```
security2_module (shared)
```

If the module is not loaded, restart Apache:

```
sudo systemctl restart apache2
```

3. Basic Configuration

Step 1: Enable the OWASP CRS Rules

Enable the **OWASP Core Rule Set (CRS)**:

```
sudo nano /etc/apache2/mods-enabled/security2.conf
```

Ensure this line is included:

```
apache
```

```
IncludeOptional /usr/share/modsecurity-crs/*.conf
```

Restart Apache:

```
sudo systemctl restart apache2
```

Step 2: Adjust Anomaly Scoring

Modify anomaly scoring to **reduce false positives**:

```
sudo nano /etc/modsecurity/crs/crs-setup.conf
```

Change:

apache

```
SecAction "id:900110,phase:1,nolog,pass,t:none,setvar:tx.inbound_anomaly_score_threshold=10000" SecAction "id:900120,phase:2,nolog,pass,t:none,setvar:tx.inbound_anomaly_score_threshold=10000" SecAction "id:900130,phase:1,nolog,pass,t:none,setvar:tx.outbound_anomaly_score_threshold=10000"
```

Restart Apache:

```
sudo systemctl restart apache2
```

4. Tiki-Specific Configuration

Step 1: Handling False Positives

Exclude **static files**:

```
sudo nano /etc/modsecurity/crs/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf
```

Add:

apache

```
SecRule REQUEST_URI "\.(jpeg|jpg|gif|png|bmp|ico|css|js)$" "id:1000017,phase:1,pass,nolog,ctl:ruleEngine=Off"
```

Allow **file uploads in Tiki**:

apache

```
SecRule REQUEST_URI "@beginsWith /tiki-upload_file.php"
"id:1000021,phase:2,pass,nolog,ctl:ruleRemoveById=200004"
```

Restart Apache:

```
sudo systemctl restart apache2
```

Step 2: Handling Language-Specific False Positives

Some actions by users on Tiki sites may trigger alerts or blocking due to ModSecurity's filtering rules. For example, words with multiple accented characters in a single word, like **"Měšťáček"** (Czech), can be flagged as suspicious. To prevent such cases from causing a **500 error** or blocking the page:

Review ModSecurity logs for blocked requests:

```
sudo tail -f /var/log/apache2/modsec_audit.log
```

Identify the specific rule blocking the request.

Create an exception rule in `REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf`.

apache

```
SecRule REQUEST_URI "@beginsWith /tiki-editpage.php"
"id:1000022,phase:2,pass,nolog,ctl:ruleRemoveById=942100"
```

Restart Apache:

sudo systemctl restart apache2

This ensures ModSecurity does not incorrectly block legitimate content written in different languages.

Conclusion

This guide helps secure Tiki with ModSecurity, prevent false positives, and block malicious bots. Regularly monitor log and adjust exclusion rules for usability.

related pages

Security Admin
Advanced Settings

external links

- http://www.modsecurity.org
- http://es.wikipedia.org/wiki/Mod_Security
- http://sourceforge.net/projects/mod-security/

aliases for this page

mod security | mod_security