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

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/*.load
```

Enable DoS Protection (Optional)

ModSecurity includes optional anti-automation and DoS (Denial of Service) protection. You can enable it by uncommenting and customizing a rule found in:

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

Add or uncomment this rule to activate burst-based blocking:

```
SecAction \ "id:900700, \ phase:1, \ nolog, \ pass, \ t:none, \ setvar:tx.dos_burst_time_slice=60, \
setvar:tx.dos_counter_threshold=100, \ setvar:tx.dos_block_timeout=600"
```

What it does:

- If a client makes more than 100 requests (excluding static files) in 60 seconds, it's considered a burst.
- After two bursts, the client is blocked for 600 seconds (10 minutes).
- Static file requests are excluded from this count (see tx.static_extensions in the same config file).
- This helps mitigate high-frequency scraping or brute-force automation attempts without blocking normal user traffic.

Restart Apache:

```
sudo systemctl restart apache2
```

4. Tiki-Specific Configuration

Tiki uses complex URLs, dynamic AJAX calls, and multiple languages. Without tailoring rules, ModSecurity might block legitimate Tiki features like editing wiki pages, uploading files, or using certain character sets.

Without proper adjustments, users may experience unexplained 403 or 500 errors during normal site usage. Below are specific steps to tailor ModSecurity to better support Tiki's functionality while maintaining security.

Handling False Positives

When ModSecurity blocks a valid request, it logs the event in the audit log. To avoid these disruptions:

1. Identify the rule causing the block in the audit log (/var/log/apache2/modsec_audit.log)
2. Create an exception for that rule in:

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

Common Example: File Upload Blocked

To fix file upload issues on Tiki, add the below rule in **/etc/modsecurity/crs/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf**

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

Then restart Apache

Language-Specific False Positives

Tiki supports many languages and character sets. A user writing in Czech, for example, might use a word like "Měšťáček", which contains multiple diacritic marks. ModSecurity may incorrectly flag this as malicious input.

Review the ModSecurity audit log:

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

Identify the triggered rule ID, then create an exclusion:

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

Restart Apache to apply changes.

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

5. Blocking Bots with ModSecurity

Bots can overload your server, scrape content, or scan for vulnerabilities. Blocking known bad bots protects performance and security.

Identifying Bots in Logs

```
grep -oiP '\w+(bot|spider|crawler)' /PATH_TO_YOUR_VHOST/logs/access_log | sort | uniq -c | sort -nr
```

This command extracts and counts how many times bots (with names like bot, spider, or crawler) appear in your access logs to help identify the most active ones.

Adding Rules to Block Bots

Instead of one rule per bot, use a list.

Create **/etc/modsecurity/bad_bots.txt**:

Add:

```
spider crawl claudebot ClaudeBot AliyunSecBot AhrefsBot SemrushBot MJ12bot DotBot Bytespider Amazonbot PetalBot Brightbot Scrapy
```

Add the rule:

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

SecRule REQUEST_HEADERS:User-Agent "@pmFromFile /etc/modsecurity/bad_bots.txt" "id:1000025,phase:1,log,deny,status:403,msg:'Blocked known bad bots from file'"

Blocking Bots by IP Address

SecRule REMOTE_ADDR "@ipMatch IP 1,IP 2" "id:1000026,phase:1,log,deny,status:403,msg:'Blocked bot IP addresses'"

6. Testing & Troubleshooting

Testing with CURL

curl -A "AhrefsBot" https://yourdomain.com

Reviewing Logs

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

7. Final Checks & Maintenance

- Monitor logs weekly
- Update **bad_bots.txt** with newly detected bots
- Review CRS updates (OWASP CRS releases often)
- Backup your configuration before changes

Conclusion

This guide helps secure Tiki with ModSecurity, prevent false positives, and block malicious bots. Regularly monitor logs 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