

Forum:[Hack Forums](#)

Thread: [Enhancing Web Application Security with .htaccess](#)

Clumsy - October 29, 2024, 8:29 pm

Put this script in your htaccess To block these type of attacks.

This .htaccess code strengthens web application security by implementing protective measures against common attack vectors and unauthorized access. It enables URL rewriting with mod_rewrite, blocks requests from tools like Wget and Curl, enforces request rate limiting to mitigate Denial of Service attacks, and filters harmful query patterns to prevent SQL Injection. Additionally, it guards against XSS and related attacks by analyzing request methods and content types, denies access to sensitive files, disables directory browsing, and sets critical HTTP security headers. Finally, it logs blocked access attempts for monitoring and threat assessment.

Code # BEGIN Enhanced Security Configuration

Enable mod_rewrite for URL rewriting capabilities

<IfModule mod_rewrite.c>

RewriteEngine On

Block requests from common tools like Wget, Curl, and Nmap

RewriteCond %{HTTP_USER_AGENT} (wget|curl|nmap) [NC]

RewriteRule ^ - [F,L]

Also block requests from other potential bots

RewriteCond %{HTTP_USER_AGENT} (java|python|php|libwww|httpclient) [NC]

RewriteRule ^ - [F,L]

Implement request rate limiting to protect against DoS attacks

RewriteCond %{REQUEST_URI} !^/path/to/allowed/resource

RewriteCond %{REMOTE_ADDR} !^123\.45\.67\.89 # Replace with allowed IP address

RewriteRule ^ - [E=LIMIT:1]

Block SQL Injection attempts

RewriteCond %{QUERY_STRING} (union|select|insert|delete|update|drop|--|#|*|*V) [NC]

RewriteRule ^ - [F,L]

Block Cross-Site Scripting (XSS) attempts

RewriteCond %{REQUEST_URI} \.php\$ [NC]

RewriteCond %{QUERY_STRING} (<|%3C)[^>]*(script|%3Cscript|%3C%20script) [NC]

RewriteRule ^ - [F,L]

Block Cross-Site Script Inclusion (XSSI) attempts

RewriteCond %{QUERY_STRING} (eval|base64_decode|gzinflate|assert) [NC]

RewriteRule ^ - [F,L]

Block XML External Entity (XXE) attacks

RewriteCond %{REQUEST_METHOD} ^(POST|GET)\$

RewriteCond %{HTTP:Content-Type} application/xml [NC,OR]

```

RewriteCond %{HTTP:Content-Type} text/xml [NC]
RewriteCond %{QUERY_STRING} (<!ENTITY|<!DOCTYPE|%20SYSTEM|%20PUBLIC|file://)
[NC]
RewriteRule ^ - [F,L]
</IfModule>

```

```

# Deny access to sensitive files to protect your site
<FilesMatch "\.(htaccess|htpasswd|ini|log|env|sql|xml)$">
    Require all denied
</FilesMatch>

```

```

# Prevent directory browsing to enhance security
Options -Indexes

```

```

# Set important security headers for better protection
<IfModule mod_headers.c>
    Header set X-Content-Type-Options "nosniff"
    Header set X-XSS-Protection "1; mode=block"
    Header set X-Frame-Options "DENY"
    Header set Referrer-Policy "no-referrer"
</IfModule>

```

```

# Log blocked access attempts for monitoring
CustomLog "/usr/bin/tee -a /var/log/blocked_requests.log" combined env=LIMIT

```

```

# END Enhanced Security Configuration

```

This .htaccess code is designed to enhance the security of a web application by implementing various protective measures against common attack vectors and unwanted access. It begins by enabling the mod_rewrite module to facilitate URL rewriting. Several rules are defined to block requests from well-known tools and programming languages commonly used for web scraping or automated testing, such as Wget, Curl, and Nmap, as well as other potential bots. To protect against Denial of Service (DoS) attacks, it enforces request rate limiting, allowing specific IP addresses to bypass this restriction. The code also contains rules to identify and block SQL Injection (SQLi) attempts by filtering out harmful query patterns, as well as Cross-Site Scripting (XSS), Cross-Site Script Inclusion (XSSI), and XML External Entity (XXE) attacks by analyzing both the request method and content type. Sensitive files are explicitly denied access to enhance security, and directory browsing is disabled to prevent unauthorized viewing of directory contents. It also sets critical HTTP security headers to further safeguard the application. Lastly, any blocked access attempts are logged for monitoring purposes, enabling the website owner to review potential threats and improve their security posture.

We can also do

Code # BEGIN Enhanced Security Configuration

```

# Enable mod_rewrite for URL rewriting capabilities
<IfModule mod_rewrite.c>
    RewriteEngine On

    # Block requests from common tools like Wget, Curl, and Nmap
    RewriteCond %{HTTP_USER_AGENT} (wget|curl|nmap) [NC]
    RewriteRule ^ - [F,L]

```

Also block requests from other potential bots

RewriteCond %{HTTP_USER_AGENT} (java|python|php|libwww|httpclient) [NC]

RewriteRule ^ - [F,L]

Implement request rate limiting to protect against DoS attacks

RewriteCond %{REQUEST_URI} !^/path/to/allowed/resource

RewriteCond %{REMOTE_ADDR} !^123\.45\.67\.89 # Replace with allowed IP address

RewriteRule ^ - [E=LIMIT:1]

Block SQL Injection attempts

RewriteCond %{QUERY_STRING} (union|select|insert|delete|update|drop|--|#|*|*V) [NC]

RewriteRule ^ - [F,L]

Block Cross-Site Scripting (XSS) attempts

RewriteCond %{REQUEST_URI} \.php\$ [NC]

RewriteCond %{QUERY_STRING} (<|%3C)[^>]*(script|%3Cscript|%3C%20script) [NC]

RewriteRule ^ - [F,L]

Block Cross-Site Script Inclusion (XSSI) attempts

RewriteCond %{QUERY_STRING} (eval\(|base64_decode\(|gzinflate\(|assert\() [NC]

RewriteRule ^ - [F,L]

Block XML External Entity (XXE) attacks

RewriteCond %{REQUEST_METHOD} ^(POST|GET)\$

RewriteCond %{HTTP:Content-Type} application/xml [NC,OR]

RewriteCond %{HTTP:Content-Type} text/xml [NC]

RewriteCond %{QUERY_STRING} (<!ENTITY|<!DOCTYPE|%20SYSTEM|%20PUBLIC|file://)

[NC]

RewriteRule ^ - [F,L]

</IfModule>

Deny access to sensitive files to protect your site

<FilesMatch "\.(htaccess|htpasswd|ini|log|env|sql|xml)\$">

Require all denied

</FilesMatch>

Prevent directory browsing to enhance security

Options -Indexes

Set important security headers for better protection

<IfModule mod_headers.c>

Header set X-Content-Type-Options "nosniff"

Header set X-XSS-Protection "1; mode=block"

Header set X-Frame-Options "DENY"

Header set Referrer-Policy "no-referrer"

</IfModule>

Block Layer 4 DoS attacks

<IfModule mod_evasive.c>

DOSHashTableSize 3097

DOSPageCount 10

DOSSiteCount 100

DOSSiteInterval 1

```
DOSBlockingPeriod 10
</IfModule>
```

```
# Block Layer 7 DDoS attacks
<IfModule mod_ratelimit.c>
    SetEnvIf Request_URI "^/path/to/allowed/resource" allow_access
    SetEnvIf Request_URI "^/api" allow_access # Adjust according to your needs
</Location />
    SetOutputFilter RATE_LIMIT
    SetEnv rate-limit 100 # Allow 100 requests per second
</Location>
</IfModule>
```

```
# Log blocked access attempts for monitoring
CustomLog "|/usr/bin/tee -a /var/log/blocked_requests.log" combined env=LIMIT
```

```
# END Enhanced Security Configuration
```

LeetSpeed - *October 29, 2024, 8:57 pm*

This is a good guide!

Lefty - *October 31, 2024, 3:56 am*

Yo. This is beaut thanks for this. I will use it!

Nachfolger - *October 31, 2024, 5:53 am*

Is this copy and pasted from ChatGTP?