

Sécuriser un projet web

🕒 Created	@May 20, 2025 11:11 AM
☰ AI keywords	Cybersecurity Web Security

Sécuriser un projet web

www.siamnews.net

www.thailand-business-news.com

Pour sécuriser Thailand Business News, nous avons eu recours principalement à deux méthodes : la sécurité par reverse proxy dans le cloud avec CloudFlare et un firewall intégré à l'interface Webmin (CSF + LFD qui est une UI pour le iptables de Linux).

DDoS Distributed Denial-of-Service (DDoS)/ attaque DDoS

Les attaques sont plus difficiles à déjouer car elles saturent le serveur avec des connexions en provenance de multiples IP réparties dans diverses zones géographiques (botnet).


CloudFlare possède une fonction DDoS intégrée qui constitue une première couche de sécurité.

Security rules	DDoS protection
Network-layer and SSL/TLS DDoS attack protection	
Rulesets managed by Cloudflare that automatically mitigates SSL/TLS-based and Network-layer DDoS attacks. Only Magic Transit and Spectrum customers on an Enterprise plan can create overrides for these rulesets.	
DDoS Mitigation ● Always active	
Ruleset	Description
SSL/TLS DDoS attack protection	Automatic mitigation of SSL/TLS based DDoS attacks and encryption-based attacks such as DDoS attacks, SSL exhaustion floods, and SSL negotiation attacks. Active
Network-layer DDoS attack protection	Automatic mitigation of network-layer DDoS attacks such as ACK floods, SYN-ACK amplification attacks, UDP attacks, ICMP attacks and DDoS attacks launched by botnets such as Mirai. Active

Nous avons ajouté en plus deux règles supplémentaires :

1. Rate limite dans le firewall (WAF)

Rate limiting rules 1 active [Create rule](#) [Go to web application exploits settings](#)

Order	Name	Match against	Action	CSR ⓘ	Events last 24h	
1	all /*	URI Path wildcard /* and URI Path does not start with /wp...	Block	-	 2.78k	Active ⋮

Expression Preview [Edit expression](#)

(http.request.uri.path wildcard "/*" and not starts_with(http.request.uri.path, "/wp-content") and not http.request.uri.path contains "ajax" and not http.request.uri.path contains "json" and not starts_with(http.request.uri.path, "/wp-includes")) or (cf.waf.credential_check.password_leaked)

With the same characteristics...

IP ▼

When rate exceeds...

Requests (required) Period (required)

Then take action...

Choose action

Blocks matching requests and stops evaluating other rules

For duration...

Duration (required)

Mais comme on le voit dans l'expression les dossiers wp-content et wp-includes sont exclus de cette protection car ils contiennent des fichiers (notamment js et css) qui sont lus très fréquemment même dans le cadre d'un trafic régulier. Les bloquer aurait risqué de nuire au trafic du site et à sa visibilité.

2. Mise en place d'un Edge cache pour les fichiers css et js et autres fichiers statiques fréquemment demandés en requête http.

Avec cette règle les fichiers statiques sont servis en cache sur les serveurs proxy de CloudFlare **sans que la requête soit recue par le serveur d'origine**, ce qui réduit la charge CPU et limite les risques de "plantage" en cas de DDoS.

Expression Preview

[Edit expression](#)

```
(http.request.uri.path.extension in {"7z" "avi" "avif" "apk" "bin" "bmp" "bz2" "class" "css" "csv" "doc" "docx" "dmg" "ejs" "eot" "eps" "exe" "flac" "gif" "gz" "ico" "iso" "jar" "jpg" "jpeg" "js" "mid" "midi" "mkv" "mp3" "mp4" "ogg" "otf" "pdf" "pict" "pls" "png" "ppt" "pptx" "ps" "rar" "svg" "svgz" "swf" "tar" "tif" "tiff" "ttf" "webm" "webp" "woff" "woff2" "xls" "xlsx" "zip" "zst"})
```

Then...

Cache eligibility (required)

Mark whether the request's response from origin is eligible for caching. Caching itself will still depend on the cache-control header and your other caching configurations. [Learn more](#)

- ☐ Bypass cache
- ☒ **Eligible for cache**

Edge TTL (optional)

Specify if and how long Cloudflare should cache the response, depending on if a cache-control header is present on the origin response.

[Learn more](#)

- ☐ Use cache-control header if present, bypass cache if not
- ☐ Use cache-control header if present, cache request with Cloudflare's default TTL for the response status if not
- ☒ **Ignore cache-control header and use this TTL**

Input time-to-live (TTL) (required)

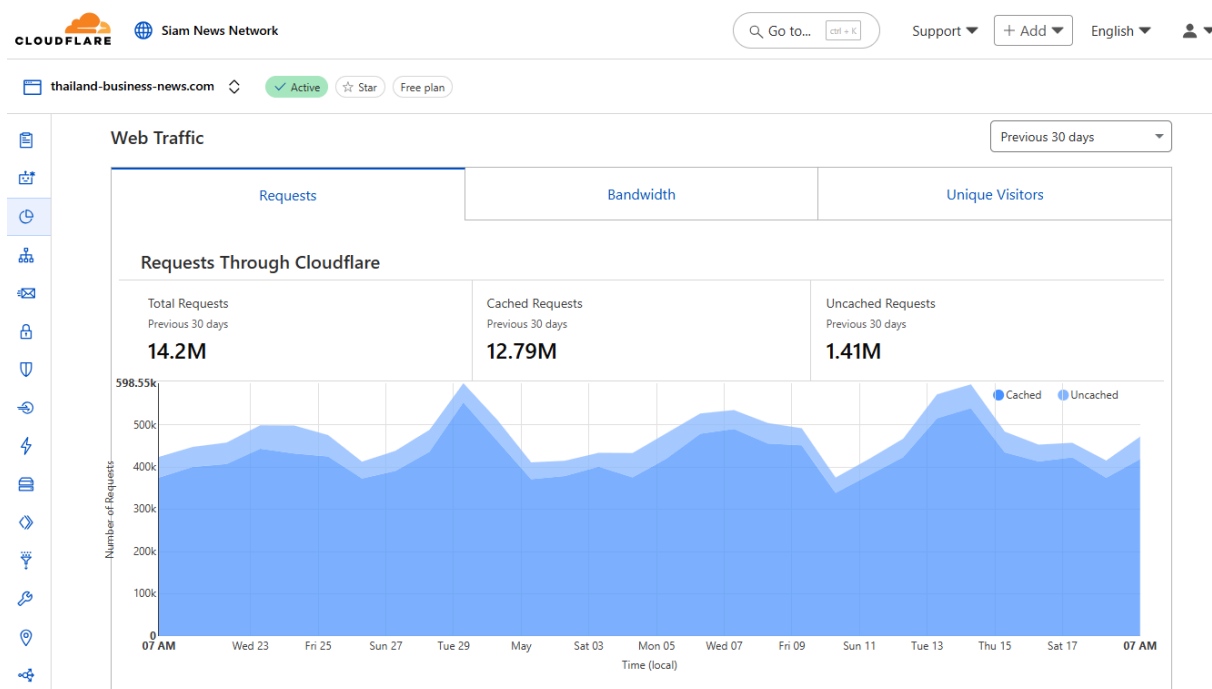
1 month

Status code TTL

Specify how long Cloudflare should cache the response based on the status code from the origin.

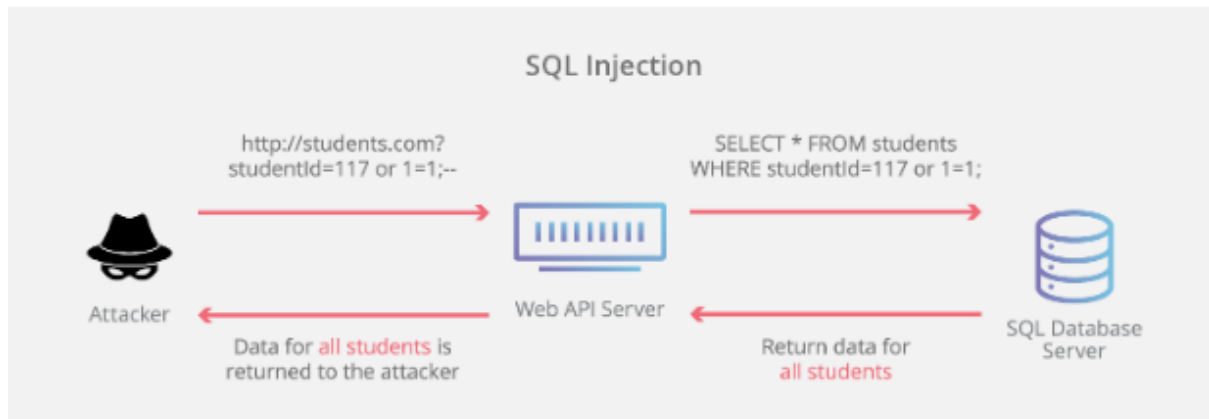
[+ Add status code setting](#)

De fait environ 90% des requetes http(s) sont servies directement en reverse proxy :



SQL Injection/ Injection SQL

"L'injection en langage de requête structurée (SQL*) est une technique d'injection de code utilisée pour modifier ou récupérer des données dans des bases de données SQL."



<https://www.cloudflare.com/fr-fr/learning/security/threats/sql-injection/>

SQLi inférentielle (également connue sous le nom de Blind SQL injection):

- **SQLi basée sur le temps** : Les pirates envoient une requête SQL à la base de données, faisant attendre la base de données pendant quelques secondes avant de répondre par vrai ou faux à la requête.
- **SQLi booléenne** : Les pirates envoient une requête SQL à la base de données, laissant l'application répondre en générant un résultat vrai ou faux.

Exemples d'injection SQL sur des organisations qui ont été victimes de SQLi:

Cisco, 2018

Dans le gestionnaire de licences Cisco Prime, une vulnérabilité injection SQL a été détectée. Ca permettait au hackers d'obtenir un accès shell aux systèmes sur lesquels le gestionnaire de licences était déployé

Tesla, 2014

En 2014, Tesla a détecté une vulnérabilité injection SQL, obtenant des privilèges d'administration et volant les données des utilisateurs

Comment prévenir une attaque par injection SQL?

Moins de privilège:

Pour réduire l'exposition à l'injection SQL, en limitant les autorisations au strict nécessaire. Un compte administratif ne doit en aucun cas exécuter des commandes SQL

Sécurisation des informations sensibles (comme le hachage des mots de passe) pour atténuer les conséquences d'une éventuelle fuite de données.

Pour une install WordPress on peut filtrer avec le fichier .htaccess (apache) avec RewriteCond pour bloquer les termes suspects

```
RewriteCond %{QUERY_STRING} [^a-z](declare|char|set|cast|convert|delete|drop|exec|insert|meta|script|select|truncate|update)[^a-z] [NC]
RewriteRule (.*) - [F]
```

Bot Attack / Attaques de Robot agressifs

Il y a différents types d'attaque causés par des robots agressifs. Pour cet exemple, je vais expliquer Bourrage d'identifiants (Credential Stuffing) et les connexions par force brute.



Spike in automated traffic detected

[Configure Super Bot Fight Mode](#)

Cloudflare scores every request on our network to determine the likelihood the request came from a bot or a human. We've detected an increase in your automated traffic that may indicate malicious bot activity.

"Les connexions par force brute sont des attaques qui utilisent des bots pour attaquer et infiltrer des comptes protégés en essayant toutes les combinaisons de mots de passe possibles ou en perçant des clés de chiffrement afin d'obtenir un accès non autorisé à des données sensibles."

Des exemples de Brute Force sur les serveurs pour le site de Siam News et Thailand Business News:

193.32.162.136 # lfd: (sshd) Failed SSH login from 193.32.162.136 (NL/The Netherlands/-): 5 in the last 3600 secs - Tue May 20 13:20:05 2025

45.148.10.240 # lfd: (sshd) Failed SSH login from 45.148.10.240 (AD/Andorra/-): 5 in the last 3600 secs - Tue May 20 16:41:38 2025

119.195.45.147 # lfd: (sshd) Failed SSH login from 119.195.45.147 (KR/South Korea/-): 5 in the last 3600 secs - Tue May 20 16:47:14 2025

36.189.207.209 # lfd: 36.189.207.209 (CN/China/-), 7 distributed sshd attacks on account [root] in the last 3600 secs - Tue May 20 16:50:24 2025

68.183.102.75 # lfd: (sshd) Failed SSH login from 68.183.102.75 (US/United States/-): 5 in the last 3600 secs - Tue May 20 16:54:49 2025

138.124.30.225 # lfd: (sshd) Failed SSH login from 138.124.30.225 (NL/The Netherlands/-): 5 in the last 3600 secs - Tue May 20 19:46:03 2025

92.118.39.61 # lfd: (sshd) Failed SSH login from 92.118.39.61 (NL/The Netherlands/-): 5 in the last 3600 secs - Tue May 20 20:43:44 2025

103.20.102.234 # lfd: (sshd) Failed SSH login from 103.20.102.234 (VN/Vietnam/-): 5 in the last 3600 secs - Tue May 20 23:53:49 2025

140.249.222.253 # lfd: (sshd) Failed SSH login from 140.249.222.253 (CN/China/-): 5 in the last 3600 secs - Wed May 21 01:44:00 2025

8.222.203.73 # lfd: 8.222.203.73 (SG/Singapore/-), 7 distributed sshd attacks on account [root] in the last 3600 secs - Wed May 21 02:22:51 2025

193.32.162.157 # lfd: (sshd) Failed SSH login from 193.32.162.157 (NL/The Netherlands/-): 5 in the last 3600 secs - Wed May 21 02:40:21 2025

93.123.16.63 # lfd: (sshd) Failed SSH login from 93.123.16.63 (BG/Bulgaria/momus.ohost.bg): 5 in the last 3600 secs - Wed May 21 05:16:50 2025

209.38.83.177 # lfd: (sshd) Failed SSH login from 209.38.83.177 (AU/Australia/-): 5 in the last 3600 secs - Wed May 21 06:34:07 2025

134.122.9.52 # lfd: 134.122.9.52 (US/United States/-), 5 distributed sshd attacks on account [root] in the last 3600 secs - Wed May 21 06:55:58 2025

59.53.92.190 # lfd: (sshd) Failed SSH login from 59.53.92.190 (CN/China/-): 5 in the last 3600 secs - Wed May 21 08:19:55 2025

196.251.83.136 # lfd: (sshd) Failed SSH login from 196.251.83.136 (SC/Seychelles/undefined.hostname.localhost): 5 in the last 3600 secs - Wed May 21 09:37:27 2025

220.81.148.101 # lfd: (sshd) Failed SSH login from 220.81.148.101 (KR/South Korea/-): 5 in the last 3600 secs - Wed May 21 10:15:48 2025

```

164 62.121.224.104 # lfd: (sshd) Failed SSH login from 62.121.224.104 (CH/Switzerland/-): 5 in the last 3600 secs - Tue Mar 4 15:09:33 2025
165 218.92.0.114 # lfd: (sshd) Failed SSH login from 218.92.0.114 (CN/China/-): 7 distributed sshd attacks on account [root] in the last 3600 secs - Tue Mar 4 15:22:37 2025
166 202.72.235.208 # lfd: (sshd) Failed SSH login from 202.72.235.208 (BD/Bangladesh/208-235-72-202-rankstel.net): 5 in the last 3600 secs - Tue Mar 4 15:41:46 2025
167 175.29.17.244 # lfd: (sshd) Failed SSH login from 175.29.17.244 (IN/India/-): 5 in the last 3600 secs - Tue Mar 4 15:44:50 2025
168 179.33.186.151 # lfd: (sshd) Failed SSH login from 179.33.186.151 (CO/Columbia/-): 5 distributed sshd attacks on account [root] in the last 3600 secs - Tue Mar 4 15:45:53 2025
169 180.76.166.65 # lfd: (sshd) Failed SSH login from 180.76.166.65 (CN/China/-): 5 in the last 3600 secs - Tue Mar 4 15:59:38 2025
170 218.92.0.223 # lfd: (sshd) Failed SSH login from 218.92.0.223 (CN/China/-): 6 distributed sshd attacks on account [root] in the last 3600 secs - Tue Mar 4 16:28:44 2025
171 218.92.0.229 # lfd: (sshd) Failed SSH login from 218.92.0.229 (CN/China/-): 5 distributed sshd attacks on account [root] in the last 3600 secs - Tue Mar 4 16:40:34 2025
172 35.238.164.107 # lfd: (sshd) Failed SSH login from 35.238.164.107 (US/United States/107.164.238.35.bc.googleusercontent.com): 5 in the last 3600 secs - Tue Mar 4 16:44:49 2025
173 218.92.0.222 # lfd: (sshd) Failed SSH login from 218.92.0.222 (CN/China/-): 5 in the last 3600 secs - Tue Mar 4 16:47:45 2025
174 218.92.0.227 # lfd: (sshd) Failed SSH login from 218.92.0.227 (CN/China/-): 6 distributed sshd attacks on account [root] in the last 3600 secs - Tue Mar 4 16:57:50 2025
175 218.92.0.217 # lfd: (sshd) Failed SSH login from 218.92.0.217 (CN/China/-): 5 distributed sshd attacks on account [root] in the last 3600 secs - Tue Mar 4 16:59:30 2025
176 78.187.21.105 # lfd: (sshd) Failed SSH login from 78.187.21.105 (TR/Turkey/78.187.21.105.dynamic.ttnet.com.tr): 5 in the last 3600 secs - Tue Mar 4 18:10:57 2025
177 193.32.162.136 # lfd: (sshd) Failed SSH login from 193.32.162.136 (NL/The Netherlands/-): 5 in the last 3600 secs - Tue May 20 13:20:05 2025
178 45.148.10.240 # lfd: (sshd) Failed SSH login from 45.148.10.240 (AD/Andorra/-): 5 in the last 3600 secs - Tue May 20 16:41:38 2025
179 119.195.45.147 # lfd: (sshd) Failed SSH login from 119.195.45.147 (KR/South Korea/-): 5 in the last 3600 secs - Tue May 20 16:47:14 2025
180 36.189.207.209 # lfd: (sshd) Failed SSH login from 36.189.207.209 (CN/China/-): 7 distributed sshd attacks on account [root] in the last 3600 secs - Tue May 20 16:50:24 2025
181 68.183.102.75 # lfd: (sshd) Failed SSH login from 68.183.102.75 (US/United States/-): 5 in the last 3600 secs - Tue May 20 16:54:49 2025
182 138.124.30.225 # lfd: (sshd) Failed SSH login from 138.124.30.225 (NL/The Netherlands/-): 5 in the last 3600 secs - Tue May 20 19:46:03 2025
183 92.118.39.61 # lfd: (sshd) Failed SSH login from 92.118.39.61 (NL/The Netherlands/-): 5 in the last 3600 secs - Tue May 20 20:43:44 2025
184 103.20.102.234 # lfd: (sshd) Failed SSH login from 103.20.102.234 (VN/Vietnam/-): 5 in the last 3600 secs - Tue May 20 23:53:49 2025
185 140.240.222.253 # lfd: (sshd) Failed SSH login from 140.240.222.253 (CN/China/-): 5 in the last 3600 secs - Wed May 21 01:44:08 2025
186 0.222.203.73 # lfd: (sshd) Failed SSH login from 0.222.203.73 (SG/Singapore/-): 7 distributed sshd attacks on account [root] in the last 3600 secs - Wed May 21 02:22:51 2025
187 193.32.162.157 # lfd: (sshd) Failed SSH login from 193.32.162.157 (NL/The Netherlands/-): 5 in the last 3600 secs - Wed May 21 02:40:21 2025
188 93.123.16.63 # lfd: (sshd) Failed SSH login from 93.123.16.63 (BG/Bulgaria/momus.ghost.bg): 5 in the last 3600 secs - Wed May 21 05:16:50 2025
189 209.30.83.177 # lfd: (sshd) Failed SSH login from 209.30.83.177 (AU/Australia/-): 5 in the last 3600 secs - Wed May 21 06:34:07 2025
190 134.122.9.52 # lfd: (sshd) Failed SSH login from 134.122.9.52 (US/United States/-): 5 distributed sshd attacks on account [root] in the last 3600 secs - Wed May 21 06:55:50 2025
191 59.53.92.190 # lfd: (sshd) Failed SSH login from 59.53.92.190 (CN/China/-): 5 in the last 3600 secs - Wed May 21 08:19:55 2025
192 196.251.83.136 # lfd: (sshd) Failed SSH login from 196.251.83.136 (SC/Seychelles/undefined.hostname.localhost): 5 in the last 3600 secs - Wed May 21 09:37:27 2025
193 220.81.148.101 # lfd: (sshd) Failed SSH login from 220.81.148.101 (KR/South Korea/-): 5 in the last 3600 secs - Wed May 21 10:15:48 2025
194

```

Bonnes pratiques à mettre en place pour empêcher ces attaques

1. Faire les updates fréquemment sur les serveurs

```

brslvn761@v1:~$ sudo -i
[sudo] password for brslvn761:
root@v1:~# apt update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Get:2 http://security.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:3 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Hit:4 https://packages.sury.org/php bookworm InRelease
Hit:5 https://software.virtualmin.com/vm/7/gpl/apt virtualmin InRelease
Fetched 103 kB in 1s (106 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@v1:~#

```

2. Changer le SSH standard (22) pour un autre

brslvn761@v1:~\$ nano /etc/ssh/sshd_config

```

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun May 4 16:38:04 2025 from 171.6.151.197
brslvn761@v1:~$ nano /etc/ssh/sshd_config
brslvn761@v1:~$

```



```
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

3. Créer un utilisateur sudo admin et bloquer root login

To disable root login via SSH, you'll need to edit the SSH configuration file (sshd_config) and change the **PermitRootLogin setting from yes to no**. Then, you need to restart the SSH service for the changes to take effect.

Detailed Steps:

Edit the SSH configuration file: Open the /etc/ssh/sshd_config file with a text editor (e.g., vi, nano).

Locate the PermitRootLogin setting: Find the line that says PermitRootLogin yes.

Change the value: Modify the line to PermitRootLogin no.

Save the changes: Save the file and exit the text editor.

Restart the SSH service: Use the appropriate command for your Linux distribution (e.g., systemctl restart sshd or service ssh restart).

Test the change: Try to SSH in as root to verify that the change has been applied

ou avec webmin panel

Allow authentication by password
☒ Yes ☐ No

Permit logins with empty passwords
☐ Yes ☒ No

Allow login by root
No, disable root login completely ▾

Allow public key authentication
☒ Yes ☐ No

Check permissions on key files
☒ Yes ☐ No

Display /etc/motd at login
☐ Yes ☒ No

✓ Save

Webmin est l'interface utiliser dans cet exemple

<https://webmin.com/>

Sources:

<https://www.cloudflare.com/fr-fr/learning/bots/what-is-a-bot-attack/>

<https://www.cloudflare.com/fr-fr/learning/ddos/what-is-a-ddos-attack/>

<https://www.cloudflare.com/fr-fr/learning/security/threats/sql-injection/>

<https://www.kaspersky.fr/resource-center/definitions/sql-injection>

<https://datadome.co/fr/bot-management-protection/comment-prevenir-les-attaques-par-injection-sql/>

https://tutoriels.lws.fr/wordpress/injections-sql#4_Securiser_WordPress_avec_la_bonne_configuration