CCN-CERT

BP/10

Recommandations de sécurité pour les CDN

RAPPORT DE BONNES PRATIQUES

FÉVRIER 2022

ccn-cert centro criptológico nacional

CCN centro criptológico nacional

Edita:

GOBIERNO DE ESPAÑA MINISTERIO DE DEFENSA

CCN centro criptológico nacional

Pº de la Castellana 109, 28046 Madrid

Date de sortie: février 2022

LIMITATION DE LA RESPONSABILITÉ

Ce document est fourni conformément aux termes contenus dans le présent document, rejetant expressément tout type de garantie implicite qui pourrait y être liée. En aucun cas, le Centre national de cryptologie ne peut être tenu responsable des dommages directs, indirects, fortuits ou extraordinaires dérivés de l'utilisation des informations et des logiciels indiqués, même s'il a été averti d'une telle possibilité.

AVIS JURIDIQUE

Index

1. À propos du CCN-CERT

Le CCN-CERT (www.ccn-cert.cni.es) est la capacité de réponse aux incidents de sécurité de l'information du Centre national de cryptologie, CCN (www.ccn.cni.es). Ce service a été créé en 2006 en tant que **CERT gouvernemental/national** espagnol et ses fonctions sont définies dans la loi 11/2002 réglementant le centre national d'intelligence, le RD 421/2004 réglementant le CCN et le RD 3/2010, du 8 janvier, réglementant le schéma de sécurité nationale (ENS), modifié par le RD 951/2015, du 23 octobre.

Selon eux, le CCN-CERT est chargé de gérer les cyberincidents qui affectent les **systèmes du secteur public, les entreprises et les organisations d'intérêt stratégique** pour le pays et tout système classifié. Sa mission est donc de contribuer à l'amélioration de la cybersécurité espagnole, en étant le centre national d'alerte et de réponse qui coopère et aide à répondre rapidement et efficacement aux cyberattaques et à faire face activement aux cybermenaces.

CCN-CERT est la capacité de réponse aux incidents de sécurité de l'information du Centre national de cryptologie, CCN.

2. Introduction aux CDN

Les réseaux de diffusion de contenu (CDN) sont des éléments de grande valeur, responsables de la diffusion de divers contenus avec lesquels l'utilisateur interagit quotidiennement. Ce type d'architecture est apparu principalement pour résoudre le problème de la latence, c'est-à-dire le délai qui se produit entre le moment où un site web est demandé et le moment où le contenu est livré et affiché à l'écran.

Ce processus est influencé par un certain nombre de facteurs, dont beaucoup sont spécifiques au type de contenu, de serveur et de site web demandé, le plus important étant la distance physique entre l'utilisateur et le serveur hébergeant le contenu.

De telles architectures ont été développées pour résoudre le problème de latence.

2. Introduction aux CDN

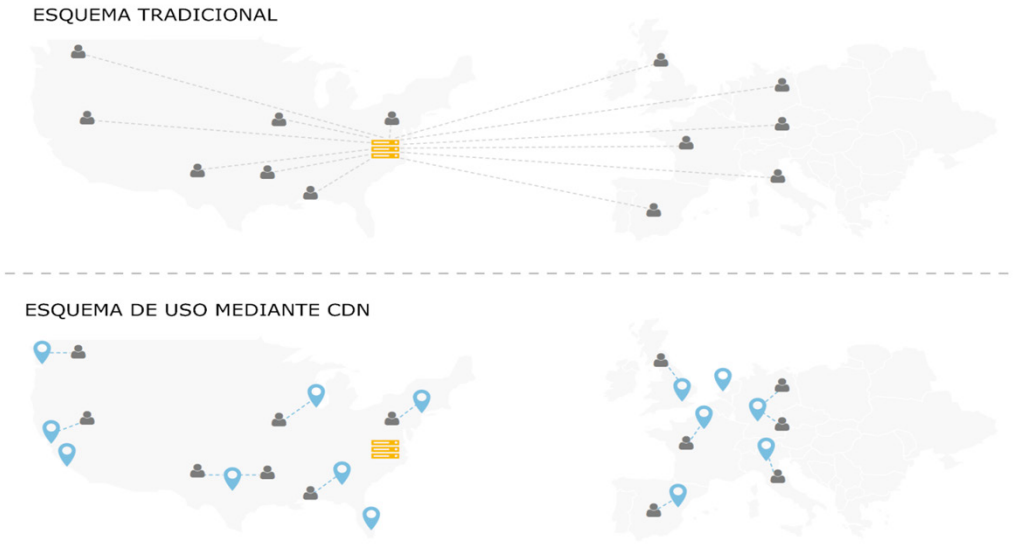


Figure 1 - Diagramme de fonctionnement d'un réseau CDN.

La principale mission d'un CDN est de réduire virtuellement cette distance physique, dans le but d'améliorer la vitesse et les performances. Pour ce faire, un CDN met en cache une version du contenu à servir dans plusieurs lieux géographiques, appelés points de *présence* (PoP). Chacun de ces PoP contient une série de serveurs de stockage chargés de fournir du contenu aux visiteurs géographiquement proches.

En plus de cette optimisation en termes de latence, un CDN offre un certain nombre d'autres avantages:

- Augmente la vitesse de chargement d'un site web.
- Bloque les bots, les *spammeurs* et autres outils nuisibles.
- Réduit la consommation de bande passante.
- Permet d'équilibrer la charge entre différents serveurs.
- Protège les sites web contre les attaques par déni de service distribué (DDoS).
- Il augmente le niveau de sécurité grâce à différentes règles et mécanismes de protection.

2. Introduction aux CDN

Pour que ces mécanismes soient applicables, il faudra, en règle générale, modifier le serveur racine DNS du domaine (par exemple, mydomain.com). Essentiellement, l'enregistrement A primaire du DNS devra être modifié pour pointer vers une adresse IP spécifique dans la gamme CDN, bien qu'il existe également des fournisseurs qui permettent la mise en œuvre au niveau CNAME sans avoir à modifier le serveur DNS racine. Dans les deux cas, l'utilisateur final sera redirigé vers le réseau CDN au lieu du serveur d'origine.

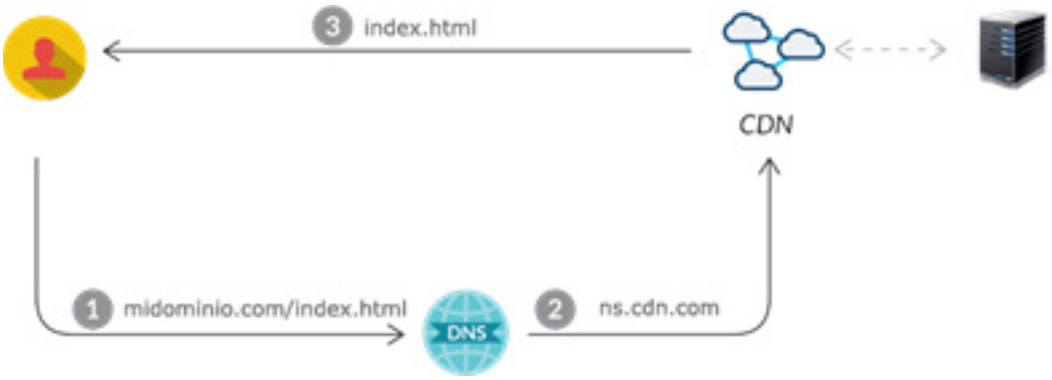


Figure 2 - Interaction entre l'utilisateur et le CDN

Il existe un grand nombre de fournisseurs de services CDN, avec des versions gratuites et payantes. Ce guide fournira des recommandations génériques pour chacun d'entre eux.

3. Comparaison générale des principaux fournisseurs de CDN

	CLOUDFLARE	Incapsula	Akamai
PoPs (points de présence)	120	42	2200
Durée minimale du contrat (mois)	1	1	12
Panneau de contrôle en ligne	✓	✓	✓
Enregistrement en ligne	✓	✓	✗
Support CNAME	✓	✓	✓
Support SPYDY	✗	✗	✓
Support HTTP/2	✓	✓	✓
Support IPV6	✓	✓	✓
Compression GZIP	✓	✓	✓
Protection contre le DDOS	✓	✓	✓
WAF	✓	✓	✓
Contrôle d'accès IP	✓	✓	✓

3. Comparaison générale des principaux fournisseurs de CDN

	CLOUDFLARE	Incapsula	Akamai
Propre certificat SSL	✓	✓	✓
Utilisation de l'API	✓	✓	✓
Accès aux journaux au format brut	✓	✗	✓
Vidéo à la demande (VOD)	✗	✗	✓
Options de stockage	✗	✗	✓
Statistiques en temps réel	✓	✓	✓

4. Techniques de protection des CDN contre les attaques DDOS

Une attaque par déni de service distribué (DDoS) est un type d'attaque qui vise à rendre un service en ligne indisponible, généralement en perturbant ou en suspendant temporairement les services fournis par le serveur. Les attaques proviennent d'appareils compromis, qu'il s'agisse d'ordinateurs personnels, de routeurs ou de dispositifs IoT, souvent distribués à l'échelle mondiale dans ce que l'on appelle un botnet.

Ces attaques diffèrent d'une attaque classique par déni de service (DoS), car les attaques DoS n'utilisent qu'un seul dispositif connecté à Internet (une connexion réseau) pour inonder une cible de trafic malveillant.

Une attaque DDoS est un type d'attaque visant à rendre un service en ligne indisponible, généralement en interrompant ou en suspendant les services fournis par le serveur.

4. Techniques de protection des CDN contre les attaques DDOS

Une fois que l'attaquant a pris le contrôle d'un *botnet*, il peut contrôler les machines en envoyant des instructions actualisées à chaque bot via un panneau de contrôle à distance. Lorsque le *botnet* cible l'adresse IP d'une victime, chaque *bot* répond en envoyant des requêtes à la cible, ce qui peut amener le serveur ou le réseau cible à dépasser sa capacité de réponse, entraînant une dégradation du service, voire une panne indéfinie.

Comme chaque *bot* est un dispositif Internet légitime, il n'est pas facile de séparer le trafic nuisible du trafic légitime.

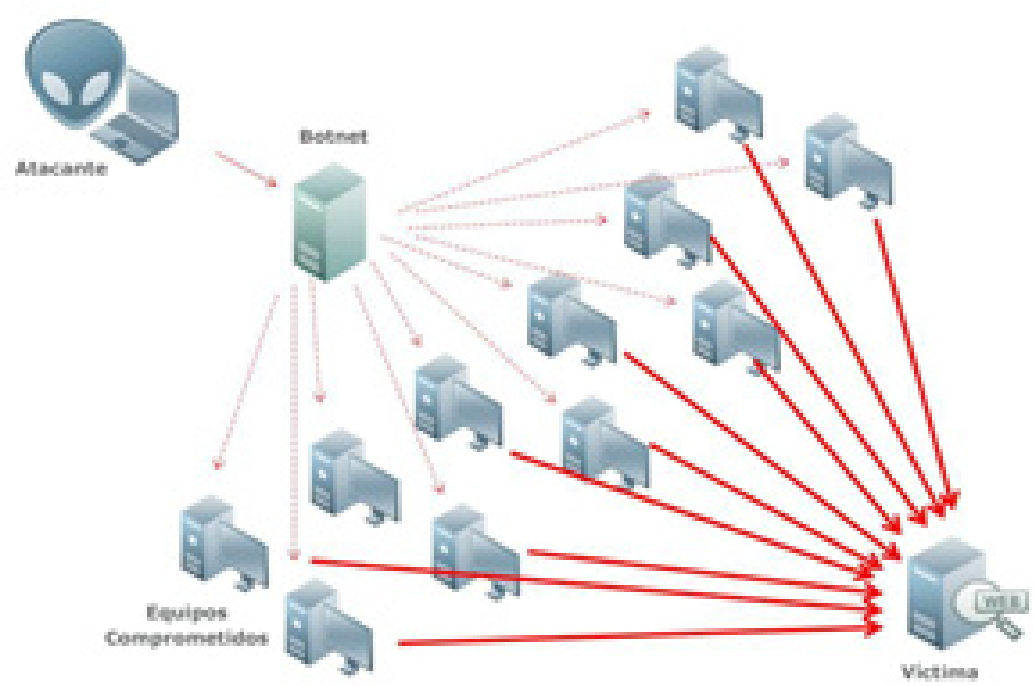


Figure 3 - Diagramme de fonctionnement d'un botnet.

4. Techniques de protection des CDN contre les attaques DDOS

De manière générale, les attaques DoS et DDoS peuvent être divisées en trois (3) types:

- **Attaques volumétriques:** comprend les inondations UDP, les inondations ICMP et autres inondations causées par des paquets générés artificiellement. L'objectif de l'attaque est de saturer la bande passante du site attaqué, et se mesure en bits par seconde (bps).
- **Attaques au niveau du protocole:** comprend les inondations SYN, les attaques par paquets fragmentés, etc. Ce type d'attaque consomme des ressources réelles du serveur ou des équipements de communication intermédiaires tels que les pare-feu et les équilibreurs de charge, et se mesure en paquets par seconde (pps).
- **Attaques de la couche 7 (au niveau de la couche applicative):** comprend des attaques telles que *slowloris*, les inondations GET/POST et les attaques visant les vulnérabilités des serveurs Apache, entre autres. En général, des requêtes apparemment légitimes sont utilisées et l'objectif de ces attaques est de faire tomber le serveur web. Ils sont mesurés en demandes par seconde (rps).

Los atacantes están motivados principalmente por:

- **Idéologie:** les "hacktivistes" utilisent les attaques DDoS pour attaquer les sites Web avec lesquels ils sont en désaccord idéologique.
- **Conflits commerciaux:** les entreprises peuvent utiliser les attaques DDoS pour mettre stratégiquement hors service les sites web de leurs concurrents, par exemple pour les empêcher de participer à un événement important (par exemple, le Black Friday).
- **Vandalisme:** les script-kiddies utilisent des outils accessibles au public sur Internet pour lancer des attaques DDoS sans motivation claire ni intention réelle de réaliser un bénéfice en retour.
- **Extorsion:** les attaquants utilisent les attaques DDoS ou la menace d'attaques DDoS comme moyen d'extorquer de l'argent à leurs victimes.
- **Guerre numérique:** les attaques DDoS autorisées par les gouvernements peuvent être utilisées contre l'infrastructure d'un pays ennemi.

L'atténuation d'une attaque DDoS nécessite une stratégie qui dépend du type d'attaque et du fait qu'il s'agisse d'un ou de plusieurs types d'attaques en même temps. En règle générale, plus l'attaque est complexe, plus il est difficile d'identifier le trafic nuisible, car l'objectif de l'attaquant est de faire en sorte que le trafic légitime paraisse légitime en rendant l'atténuation aussi inefficace que possible.

C'est pourquoi **pratiquement tous les services CDN proposent des services pour atténuer ce type d'attaque.** En fonction du niveau de sécurité que l'utilisateur a sélectionné, généralement *Attaque faible, moyenne* ou *élevée,* le CDN analysera et filtrera les connexions, en mettant en œuvre une série de mesures de sécurité supplémentaires pour filtrer le trafic.

4.1 Les défis du Javascript

Il s'agit d'un type de défi qui est utilisé pour filtrer les attaquants qui utilisent des outils automatisés des clients légitimes. Le défi est basé sur l'envoi à chaque client, attaquant ou utilisateur légitime, d'un code *JavaScript* qui comprend une sorte de défi. Pratiquement tous les navigateurs actuels disposent d'un moteur *JavaScript* et comprendront et résoudront facilement le défi de manière transparente (sans interaction avec l'utilisateur), tandis que les outils DDoS automatisés ne sont généralement pas équipés de telles fonctionnalités *JavaScript* et ne seront donc pas en mesure de surmonter le défi et d'établir la connexion avec le serveur final.

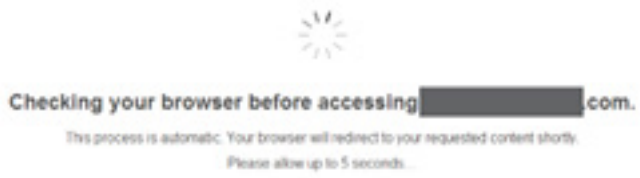


Figure 4 - Défi JavaScript fourni par CloudFlare

4.2 Tests sur les paquets SYN TCP

Avec cette méthode, l'objectif est de vérifier que la pile TCP du client est valide et correctement implémentée, en recherchant une réponse correcte à certains paquets construits dans des conditions inhabituelles, ce qui permet de détecter les paquets provenant d'adresses IP sources usurpées et les paquets *générés par les sockets à l'aide d'*un outil DDoS.

Les tactiques courantes vont du renvoi d'un paquet RST dans le premier SYN reçu (en espérant que le client le renvoie) à l'envoi délibéré d'un SYN-ACK avec un numéro de séquence incorrect en espérant que le client renvoie un RST puis réessaie.

La façon la plus simple de répondre à ce type de tests est de permettre au système d'exploitation de répondre à ces paquets, ce qui permet de savoir s'il s'agit d'une connexion légitime.

Il existe principalement deux (2) techniques :

a. **Réinitialisation TCP:** le CDN envoie un paquet avec le *drapeau* RESET (RST) actif pour réinitialiser les connexions TCP établies (celles qui ont terminé avec succès la *poignée de main*). Il s'agit de la méthode de vérification la plus courante, car les outils DDoS et les *bots* conçus n'ont pas cette logique implémentée, contrairement à une connexion via un vrai navigateur.

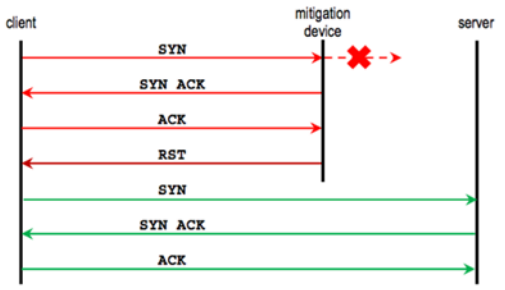


Figure 5 - Fonctionnement de l'atténuation par réinitialisation TCP

b. **TCP hors séquence:** contrairement à la méthode précédente, le CDN peut délibérément défier le client en envoyant des réponses SYN-ACK avec un numéro hors séquence comme le montre la figure ci-dessous. Comme le numéro de séquence est incorrect, le client est censé rétablir la connexion TCP et établir à nouveau la connexion. Là encore, un *bot* ou un outil DDoS ne résoudrait pas ces cas.

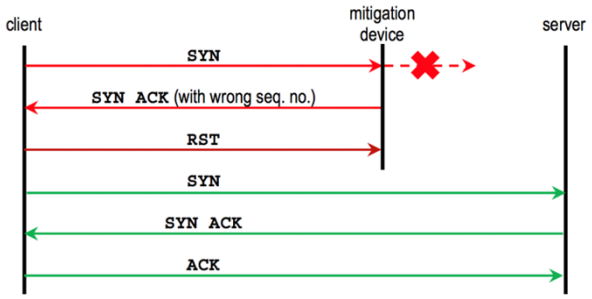


Figure 6 - Fonctionnement de l'atténuation par TCP hors-séquence

4.3 Filtrage des connexions SSL

Aujourd'hui, en raison du renforcement des mesures de sécurité contre les techniques DDoS classiques, nous commençons à observer une augmentation des inondations dommageables de connexions SSL.

Ces inondations SSL contournent un grand nombre de dispositifs de sécurité, tels que les pare-feu ou les systèmes de protection contre les intrusions (IPS). En outre, il existe des variantes actuelles telles que les attaques par renégociation SSL, où, contrairement aux attaques traditionnelles, l'attaquant n'a besoin que d'un dixième de la puissance de calcul du serveur non protégé pour prendre le contrôle de ses ressources et perturber le trafic légitime vers celles-ci.

Un fournisseur de CDN supprime simplement les connexions SSL vides ou mauvaises, protégeant ainsi les ressources qu'elles contiennent.

4.4 Redirection 302 HTTP

L'idée de base est qu'un navigateur légitime respectera les redirections HTTP 302. Pour cette raison, les redirections générées dynamiquement sont testées pour s'assurer que le visiteur est un utilisateur légitime, qui peut interpréter ces actions par le biais du navigateur.

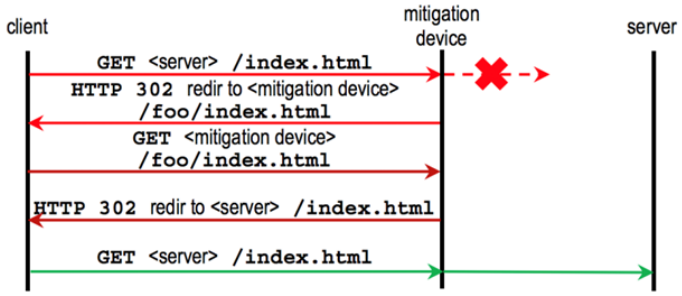


Figure 7 - Fonctionnement de l'atténuation par redirection HTTP

4.5 Cookies HTTP

Cette technique est généralement liée à la précédente et permet d'identifier le trafic malveillant en injectant dynamiquement un cookie dans la connexion entre le client supposé légitime et le CDN. Le trafic qui n'est pas en mesure d'interpréter cette nouvelle situation, car il s'agit probablement de trafic malveillant, sera écarté.

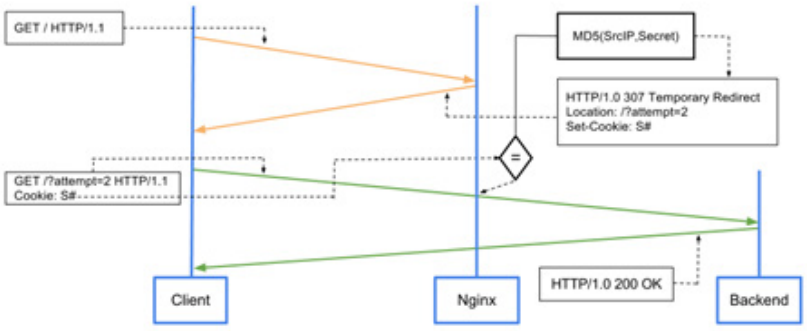


Figure 8 - Fonctionnement de l'atténuation des cookies HTTP dans un module Nginx

4.6 CAPTCHA

Un CAPTCHA *(Completely Automated Public Turing test to tell Computers and Humans Apart)* est un test de défi-réponse utilisé pour déterminer si l'utilisateur est humain ou non.

Cette technique est probablement la plus répandue, la plus simple et la plus sûre à utiliser, car elle implique une intervention humaine directe. Si le client réussit à le résoudre, il sera maintenu sur la liste blanche pendant un certain temps ou pour une certaine quantité de trafic, après quoi il devra s'authentifier à nouveau.

Figure 9 - Exemple de fonctionnement d'un système de défi-réponse basé sur CAPTCHA.

Cette méthode est, en soi, assez intrusive et, en pratique, elle est généralement utilisée lors de la configuration du CDN en mode *"Under Attack"*.

5. Recommandations de sécurité dans l'utilisation des CDN

5.1 Configurer SSL/TLS sur la connexion entre l'utilisateur et le CDN

La communication SSL/TLS entre le CDN et le serveur d'origine doit être utilisée autant que possible. *Transport Layer Security* (TLS) est un protocole de cryptage des données envoyées sur Internet, qui a évolué à partir de *Secure Sockets Layer* (SSL), afin de corriger la plupart des défauts de sécurité de ce protocole (l'industrie utilise encore ces termes de manière interchangeable pour des raisons historiques).

De cette façon, TLS est conçu pour fournir:

Authentification

Cryptage

Intégrité

Afin d'activer TLS sur la configuration, le site web aura besoin d'un certificat SSL et de sa clé privée correspondante. Il existe un certain nombre d'outils, ainsi qu'une méthode d'évaluation simple permettant aux administrateurs d'évaluer la configuration du serveur SSL en tenant compte de facteurs tels que la validité et la confiance des certificats, la prise en charge des protocoles, l'échange de clés et le cryptage.

En combinant les résultats de ces tests, avec une note comprise entre 0 et 100, on obtient une note globale qui est convertie en une lettre, de A (note la plus élevée) à F (note la plus basse).

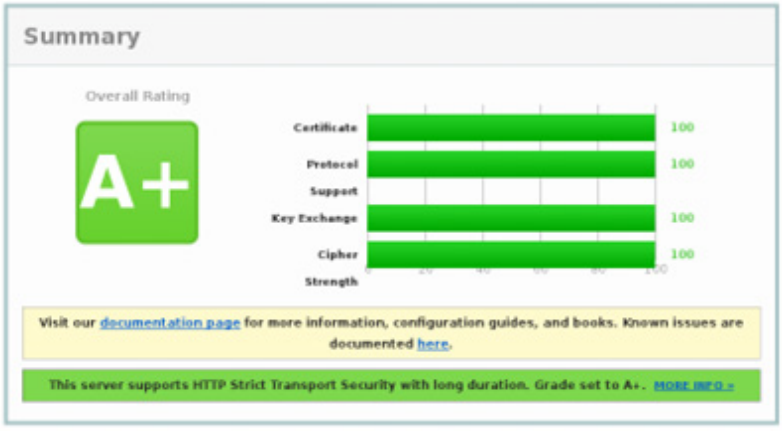


Figure 10 - Exemple de l'outil de notation SSL Server Test de Qualys

L'activation de SSL/TLS sur la configuration du CDN a l'avantage de fournir une sécurité aux visiteurs en utilisant un certificat autogénéré.

Étant donné que les visiteurs se connectent uniquement au CDN, un certificat moins sûr (par exemple, classé C) utilisé entre le serveur d'origine et le CDN n'affectera pas cette expérience, car celui proposé par le CDN aura probablement un meilleur score, sans qu'il soit nécessaire de modifier la configuration sur le serveur d'origine.

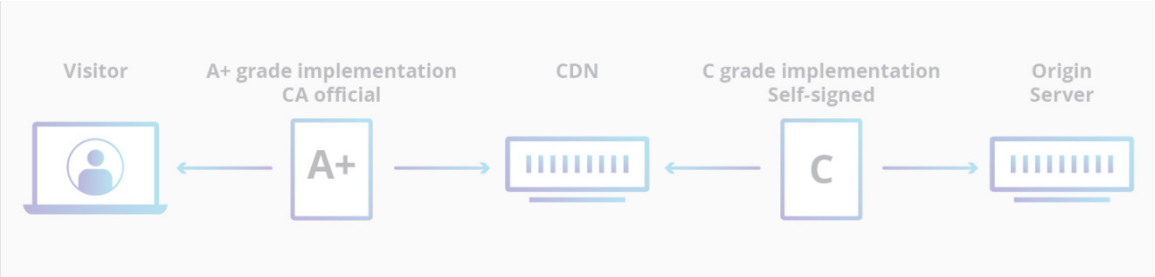


Figure 11 - Mise en œuvre et sécurisation du certificat CDN

5.2 Configuration pour les services sous connexion HTTP

Dans certains cas, le serveur d'origine n'est pas configuré pour offrir des connexions sécurisées sous SSL, soit en raison d'une administration incorrecte, soit parce qu'il s'agit d'un service qui ne peut être configuré de manière sécurisée.

Dans ces cas, il sera nécessaire de configurer la connexion du CDN sur SSL pour le trafic entre le CDN et les visiteurs, tandis que la connexion entre le CDN et le site web est maintenue sur http sans nécessiter de configuration supplémentaire sur le serveur web d'origine.

En outre, nous devons tenir compte du fait que l'absence de mise en œuvre de SSL peut avoir des résultats négatifs sur d'autres aspects, tels que le positionnement de certains moteurs de recherche comme Google.

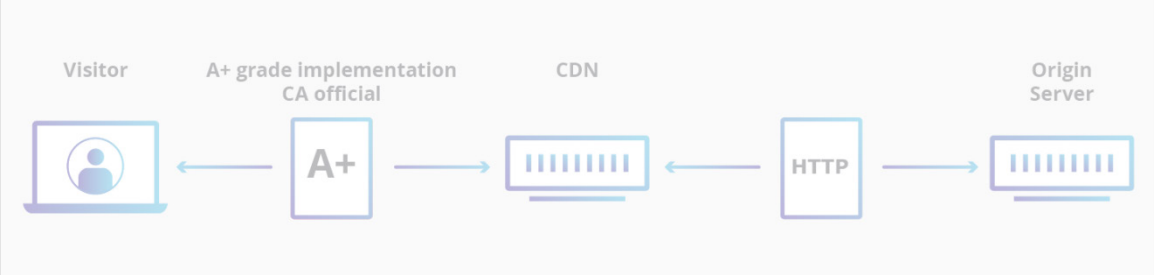


Figure 12 - Mise en œuvre de SSL sur les serveurs http

De plus, le CDN lui-même choisira les algorithmes de cryptage les plus sûrs, quelle que soit la configuration du site web, et mettra en œuvre d'autres options telles que le *Forward Secrecy*.

5.3 Activer l'utilisation des HSTS

HSTS *(HTTP Strict Transport Security)* est une technologie de politique de sécurité web, conçue pour aider à protéger les serveurs HTTPS contre les attaques par dégradation.

Ce type d'attaques par dégradation (connues sous le nom d'*attaques par arrachage SSL*) permet des attaques MiTM *(Man-in-the-middle)* où un attaquant potentiel pourrait rediriger le navigateur d'un serveur web HTTPS correctement configuré vers son propre serveur via un canal HTTP. Une fois cette redirection effectuée, les données échangées, telles que les *cookies*, peuvent être compromises.

Les navigateurs qui prennent en charge le HSTS sont les suivants:

Google Chrome à partir de la version 4.0.211.0.

Google Chrome pour Android depuis la version 18.

Firefox et Firefox Mobile depuis la version 4.

Fonctionne à partir de la version 12.

Safari depuis la version 7.

Navigateur Android depuis la version 4.4 d'Android.

Internet Explorer prévoit de l'implémenter dans la version 12 de son navigateur.

La plupart des CDN vous permettent de configurer et d'activer cette option, en ajoutant des en-têtes où vous pouvez définir la durée de cette politique, inclure différents sous-domaines, etc.

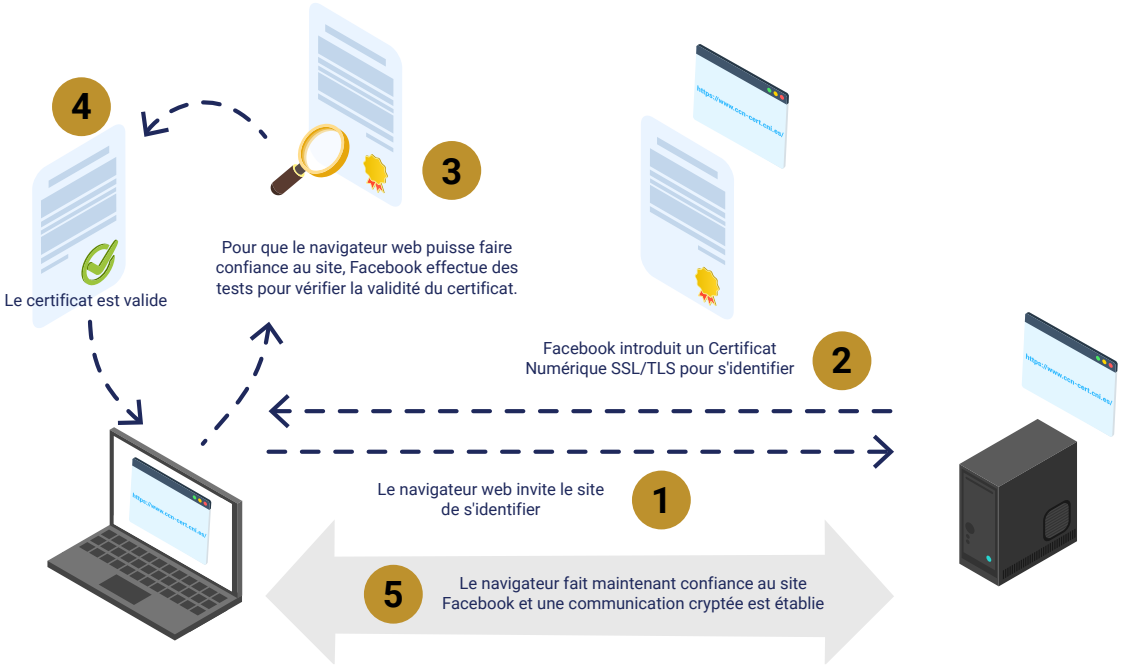
5.4 Utilisation du certificat du client

TLS (la version moderne de SSL) permet à un client de vérifier l'identité du serveur avec lequel il établit une connexion. Normalement, une *poignée de main* TLS est à sens unique, c'est-à-dire que le client peut vérifier l'identité du serveur, mais le serveur ne peut pas vérifier l'identité du client. Dans une *poignée de main* TLS authentifiée par le client, les deux parties fournissent un certificat pour vérifier leur identité.

Si le serveur d'origine est configuré pour accepter les demandes utilisant un certificat client valide du CDN, le trafic qui ne passe pas par le réseau du CDN sera écarté, car il ne possède pas le certificat correct.

Cela signifie que les attaquants ne peuvent pas contourner les fonctionnalités du CDN telles que le WAF, les règles de limitation des connexions ou la protection contre les attaques DDoS. Même si un attaquant potentiel parvenait à usurper l'adresse IP source en usurpant n'importe quelle adresse IP du pool CDN, il ne serait pas en mesure d'établir une connexion avec le serveur final.

Par conséquent, la mise en œuvre et la configuration du certificat client du fournisseur de CDN ajouteront un nouveau niveau de sécurité au service web, ainsi que la prévention des fuites de données potentielles et l'élimination d'une grande partie du risque des outils d'analyse de masse.



5.5 Changer l'adresse IP d'origine associée au serveur

Si le site web n'a pas été initialement configuré avec le CDN, il se peut qu'il y ait eu une période où le DNS pointait directement sur l'adresse IP source. Grâce à des outils disponibles en ligne, il est possible d'interroger facilement l'historique des enregistrements DNS, ce qui peut révéler les adresses IP utilisées avant l'activation du CDN et qui peuvent encore être utilisées dans le même but.

Ce risque ne peut être atténué directement, car les informations disponibles sont historiques. Cependant, il est toujours possible de demander une nouvelle adresse IP dans une plage différente de celle qui aurait été maintenue lors de la mise en œuvre du CDN.

Vous trouverez ci-dessous un exemple d'un tel outil montrant les informations historiques qu'il conserve.

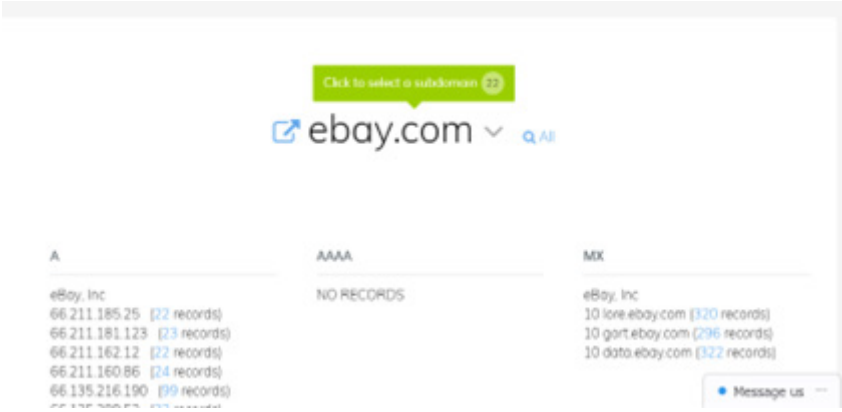


Figure 13 - Obtention d'informations historiques sur un domaine

5.6 Autoriser uniquement l'accès au pool d'adresses IP du CDN

L'une des principales tâches à effectuer lors de la mise en place d'un CDN est de mettre sur une liste blanche toutes les adresses IP spécifiées par le fournisseur pour autoriser les connexions, en bloquant toute autre requête externe.

Cela doit être fait au niveau d'IPv4 et d'IPv6, s'ils sont utilisés, et dans le cas de Linux, *iptables* peut être utilisé pour le faire de manière simple. Par exemple, pour autoriser la connexion à partir de l'adresse IP 1.1.1.1.1, vous pouvez exécuter la commande suivante:

```
$ iptables -I INPUT -p tcp -m multiport --dports http,https -s "1.1.1.1"
-j ACCEPT
```

Une dernière règle sera ensuite ajoutée pour bloquer toute tentative de connexion à partir d'adresses IP qui n'ont pas été spécifiées avec la commande précédente:

```
$ iptables -I INPUT -p tcp -m multiport --dports http,https -j DROP
```

Il est à noter que, selon le système d'exploitation utilisé, il sera également nécessaire de sauvegarder cette table afin de la récupérer en cas de redémarrage du système. Dans le cas des systèmes Linux, il est possible de sauvegarder l'état des règles iptables avec les commandes suivantes:

- **Debian/Ubuntu:** iptables-save > /etc/iptables/rules.v4
- **RHEL/CentOS:** iptables-save > /etc/sysconfig/iptables
- **Debian/Ubuntu:** ip6tables-save > /etc/iptables/rules.v6
- **RHEL/CentOS:** ip6tables-save > /etc/sysconfig/ip6tables

5.7 Protéger contre les attaques par force brute et limiter les connexions

Une autre caractéristique utile des CDN est qu'ils permettent à l'administrateur de fixer des limites de connexion pour protéger le site web contre les attaques par déni de service, les tentatives de connexion par force brute contre un panneau d'accès ou d'administration, et d'autres comportements abusifs dirigés vers la couche applicative.

Figure 14 - Fonctionnement d'une attaque par force brute

En général, les modules de connexion sont effectués par le biais de requêtes POST où un nom d'utilisateur et un mot de passe sont envoyés au serveur web. Lors de la préparation d'une attaque par force brute, un attaquant capture ces demandes de connexion et les utilise pour générer une attaque automatisée, lançant des demandes de manière consécutive, au rythme le plus élevé possible, afin d'obtenir une combinaison valide de nom d'utilisateur et de mot de passe qui permettra l'accès.

Par exemple, le système peut être configuré pour empêcher un éventuel attaquant de réaliser une attaque par force brute contre le panneau d'administration d'un site Wordpress, en créant un blocage lorsqu'un certain nombre de tentatives infructueuses est atteint ou, de manière générique, lorsqu'une erreur 401 (accès non autorisé) ou 403 (accès interdit) survient, en détectant un crawler ou un bot nuisible et en générant une action lorsque, par exemple, il accède à des pages qui ne sont pas trouvées (codes 404), etc.

Cette limitation permet, plus généralement, de fixer des seuils de temps, de définir le type de réponse sur des URL spécifiques du site web, voire de réduire la bande passante utilisée et de protéger le serveur final.

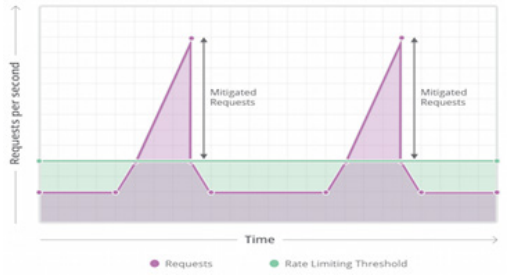


Figure 15 - Fonctionnement de la limitation du débit

5.8 Vérifier la configuration des enregistrements DNS

Aucun des enregistrements DNS ne doit contenir la moindre mention de l'adresse IP source. Les enregistrements DNS suivants doivent être vérifiés minutieusement pour s'assurer qu'ils ne contiennent aucune information sur la source :

SPF: SPF (acronyme de Sender Policy Framework) est une protection contre l'usurpation d'adresse lors de l'envoi de courriers électroniques. Il identifie les serveurs de messagerie SMTP autorisés pour le transport des messages via les enregistrements DNS.

TXT: Un enregistrement TXT est un type d'enregistrement DNS qui contient des informations textuelles provenant de sources externes à un domaine et qui est ajouté à la configuration du domaine.

En outre, il convient d'accorder une attention particulière à la configuration du reste des sous-domaines hébergés, car certains d'entre eux pourraient encore être configurés directement sur l'adresse IP du serveur d'origine, l'exposant ainsi aux attaquants et annulant les mesures d'atténuation du CDN contre différentes attaques.

5.9 Héberger le courrier sur un autre serveur

Si le serveur de messagerie est hébergé sur la même adresse IP que le serveur web d'origine, un attaquant pourrait trouver l'adresse IP dans un courriel sortant.

Par exemple, un attaquant pourrait envoyer un courriel à une adresse inexistante pour générer un rebond, où l'adresse IP source du serveur pourrait être exposée dans les en-têtes (le rebond peut contenir l'adresse IP de votre serveur dans l'un de ses champs). Il existe même

des cas où il n'est pas nécessaire d'envoyer un courriel si la plateforme dispose, par exemple, d'un système d'enregistrement des utilisateurs ou pour obtenir le mot de passe au cas où l'utilisateur l'aurait oublié.

En outre, les enregistrements MX indiquent toujours le serveur de messagerie réel. Par conséquent, si le courrier d'un domaine passe par le même serveur web, vous pouvez facilement trouver l'adresse IP grâce aux enregistrements MX.

L'image suivante montre un exemple, où la véritable adresse IP du serveur (52.x.x.x) a été exposée par l'utilisation d'un e-mail:

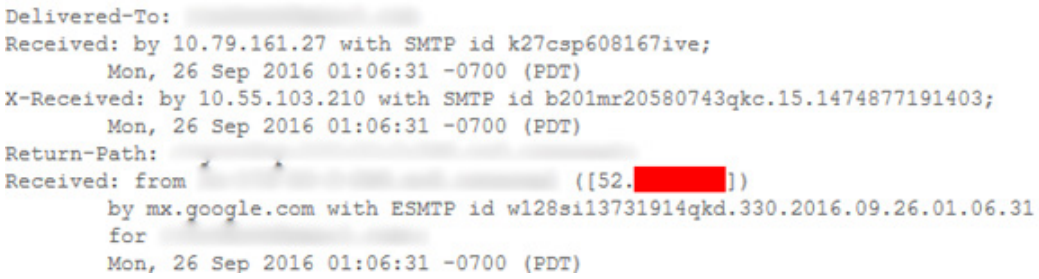


Figure 16 - Découverte de l'IP source par le service de courrier électronique

Une autre option pour éviter ce type de fuite d'informations serait d'utiliser le service par l'intermédiaire d'un tiers pour envoyer les courriers au nom de son propre domaine.

5.10 Désactiver l'inclusion dynamique de fichiers

Un grand nombre de services web offrent à l'utilisateur la possibilité de définir, par exemple, une image personnalisée comme avatar de l'utilisateur. Dans la plupart des cas, ils offrent la possibilité non seulement de télécharger directement une image sur le serveur, mais aussi de fournir une URL distante à partir de laquelle récupérer l'image.

Si le serveur télécharge ce fichier, cela ouvre la voie à un nouveau vecteur d'attaque, permettant à un attaquant potentiel d'obtenir facilement la véritable adresse IP.

Avec quelques lignes de PHP et un fichier *.htaccess* correctement choisi, il est possible de créer un lien vers une ressource (par exemple une image) qui enregistre l'adresse IP de toute personne qui tente de la récupérer:

```
# Contenu du fichier .htaccess
AddHandler application/x-httpd-php5 .jpg

# Fichier d'exemple file.php
    <?php

    $fh =@fopen("log.txt", "a+");
    $timestamp = date('l jS \of F Y h:i:s A');
    $hostname = @gethostbyaddr($_SERVER['REMOTE_ADDR']);

    @fwrite($fh, "\r\n$timestamp\r\n");
    @fwrite($fh, 'REMOTE_ADDR: '.$_SERVER['REMOTE_ADDR']."\r\n");
    @fwrite($fh, 'Host Name: '."$hostname\r\n");
    @fwrite($fh, 'HTTP_CLIENT_IP: '.$_SERVER['HTTP_CLIENT_IP']."\r\n");
    @fwrite($fh, 'HTTP_USER_AGENT: '.$_SERVER['HTTP_USER_AGENT']."\r\n");

    fclose($fh);
    // bait.png est l'image à afficher lors de la capture de l'ip
    // donnez 755 permissions à bait.png
    $im = imagecreatefrompng("bait.png");
    header('Content-Type: image/jpeg');
    imagepng($im);
    imagedestroy($im);
    ?>
```

Un attaquant potentiel pourrait utiliser un certain nombre d'autres services en ligne pour atteindre le même objectif, sans avoir besoin de son propre serveur web, avec les mêmes résultats et un niveau d'anonymat encore plus élevé.

5.11 Configuration du WAF et de la protection au niveau des applications

Dans certains cas, les attaquants utilisent les attaques DDoS pour affaiblir les défenses du périmètre ou bloquer les dispositifs de sécurité. Pour cette raison, de nombreux CDN offrent la possibilité d'utiliser un système WAF *(Web Application Firewall)*, qui protégera le client contre un grand nombre d'attaques.

Un pare-feu d'application web est un pare-feu qui surveille la sécurité web et filtre ou bloque le trafic vers et depuis une application web. Il peut ainsi filtrer le contenu des applications web, tandis qu'un pare-feu réseau protège le trafic uniquement entre les serveurs.

En règle générale, un WAF contient les règles qui s'appliquent aux 10 principales vulnérabilités de l'OWASP:

- **Injection.**
- **Authentification et gestion des sessions.**
- **Cross-Site Scripting (XSS).**
- **Références d'objets non sécurisées.**
- **Faibles paramètres de sécurité.**
- **Fuite d'informations confidentielles.**
- **Absence de contrôle d'accès.**
- **Falsification de requête intersite (CSRF).**

- **Utilisez des composants dont les vulnérabilités sont connues.**
- **Redirections et soumissions non validées.**

En outre, une attention particulière doit être accordée à la configuration du reste des sous-domaines hébergés, car certains d'entre eux pourraient encore être configurés directement à l'adresse IP du serveur d'origine, l'exposant ainsi à d'éventuels attaquants et annulant l'atténuation du CDN contre différentes attaques.

Figure 17 - Système de protection WAF basé sur le CDN

5.12 Éviter les moteurs de recherche de services

Il existe un certain nombre d'outils de recherche de services et de serveurs disponibles en ligne qui peuvent donner accès aux informations relatives à l'adresse IP source du service nouvellement configuré.

L'un des plus connus est **Shodan**, un moteur de recherche qui permet à l'utilisateur, par le biais de divers filtres, de trouver des équipements spécifiques (routeurs, serveurs, etc.) connectés à Internet. Il peut également être compris comme un moteur de recherche de bannières de service, qui sont des métadonnées que le serveur renvoie au client.

Ces informations peuvent contenir les versions du logiciel du serveur, les options prises en charge par le service, un message de bienvenue ou tout autre élément dont le client peut avoir connaissance avant d'interagir avec le serveur et qui peuvent être utilisés pour localiser le serveur d'origine se trouvant derrière un service CDN.

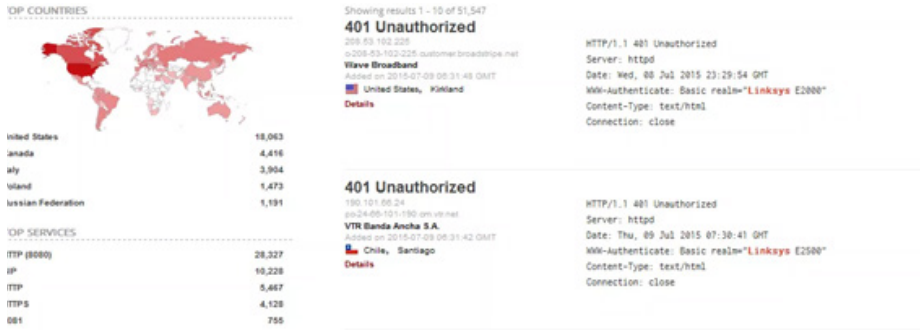


Figure 18 - Exemple de recherche dans Shodan

Un autre service largement utilisé est **Censys**, qui collecte quotidiennement des données sur les serveurs, les services et les sites web, en balayant tout le spectre des adresses IPv4. Par exemple, sachant que le domaine moz.com est protégé par un CDN, vous pouvez lancer une recherche sur Censys pour essayer d'obtenir plus d'informations et obtenir ce qui suit:

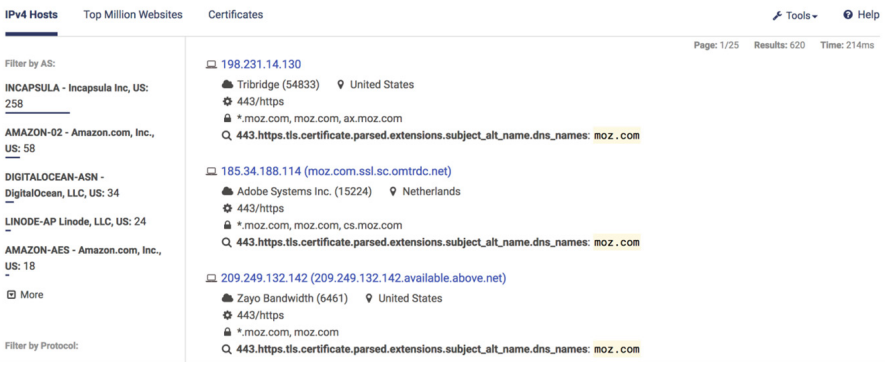


Figure 19 - Recherche Censys du domaine moz.com

Comme on peut le voir dans la capture d'écran, il y a un grand nombre de champs qui peuvent être utilisés pour effectuer ces recherches, comme le CN (Common Name), le SAN (Subject Alternative Name), le calcul SHA256 du certificat, le code de réponse HTTP (dans ce cas 200), etc. jusqu'à trouver l'adresse IP source du domaine, qui dans ce cas appartient à la plage 209.249.132.0/24.

6. Guide de sécurité de base

Ce guide des bonnes pratiques vise à poser les bases des mesures de sécurité à prendre en compte lors de la migration d'un service vers un fournisseur CDN.

Decalogue de securite de base

1. Configuration SSL/TLS, y compris HSTS, sur la connexion entre l'utilisateur final et le CDN.
2. Utiliser des certificats clients entre le CDN et le serveur d'origine.
3. Modifier l'adresse IP du serveur d'origine si elle a été exposée auparavant.
4. Autoriser uniquement l'accès au serveur d'origine par une adresse IP du pool CDN.
5. Protéger le service web contre les attaques par force brute et mettre en place une limitation du nombre de connexions clients.
6. Vérifier la configuration des enregistrements DNS.
7. Migrer le service de courrier électronique si nécessaire.
8. Désactiver, le cas échéant, l'inclusion de fichiers dynamiques dans le service web correspondant.
9. Mettre en place un WAF et appliquer des mesures de sécurité au niveau des applications.
10. Vérifier que les données du serveur d'origine ne figurent pas dans les moteurs de recherche Internet tels que Shodan ou Censys.

Références

- http://hopandfork.org/2016/11/16/cloudflare-ip-unveil.html
- https://www.defcon.org/images/defcon-21/dc-21-presentations/Mui-Lee/DEFCON-21-Miu-Lee-Kill-em-All-DDoS-Protection-Total-Annihilation-WP-Updated.pdf

centro criptológico nacional

centro criptológico nacional

www.ccn.cni.es

www.ccn-cert.cni.es

oc.ccn.cni.es