SISR

CMS Hugo / WordPress

Dans ce document, nous aborderons la mise en place des CMS **Hugo** et **WordPress**. Nous détaillerons également les solutions de protection à mettre en œuvre, telles que **iptables**, **Fail2Ban** et **PortSentry**. Par ailleurs, des solutions de test seront intégrées pour évaluer la sécurité et la performance. Enfin, nous verrons comment configurer et utiliser **RSYNC** pour la synchronisation et la sauvegarde des données.

Saint Paul Bourdon Blanc

Table des matières

OBJECTIF : Cette section de la procédure vise expliquer la mise en place du CMS Hugo

MODE OPÉRATOIRE :

Configuration

Pour pouvoir installée la dernière version de Hugo vous devez premièrement installée git sur :

➔ Apt install git

```
root@debian12CLI:~# sudo apt install git
```

Ensuite, utilisez la commande `wget` pour vous rendre sur le site officiel de Hugo CMS ou sur leur dépôt GitHub. Une fois cela fait, copiez le lien du fichier que vous souhaitez télécharger :

➔ Wget -c https://github.com/gohugoio/hugo/releases/download/v0.135.0/hugo_extended_0.135.0_linux-amd64.debd

```
root@hugocms:~# wget -c https://github.com/gohugoio/hugo/releases/download/v0.135.0/hugo_extended_0.135.0_linux-amd64.deb
--2024-10-07 09:00:20--  https://github.com/gohugoio/hugo/releases/download/v0.135.0/hugo_extended_0.135.0_linux-amd64.deb
Résolution de github.com (github.com)… 140.82.121.3
```

Une fois cela fait taper la commande suivante :

➔ dpkg -i hugo_extended_0.135.0_linux-amd64.deb

```
root@hugocms:~# dpkg -i hugo_extended_0.135.0_linux-amd64.deb
```

Vous pouvez vérifier la version installée :

➔ hugo -v

```
root@hugocms:~# hugo -v
ERROR deprecated: --verbose was deprecated in Hugo v0.114.0 and will be removed in Hugo 0.136.0. use --logLevel info
Total in 1 ms
Error: Unable to locate config file or config directory. Perhaps you need to create a new site.
Run `hugo help new` for details
```

Une fois Hugo installé, vous pouvez créer un site avec la commande suivante :

➔ hugo new site monsite

```
root@hugocms:~# hugo new site monsite
*Congratulations! Your new Hugo site was created in /root/monsite.

Just a few more steps...

1. Change the current directory to /root/monsite.
2. Create or install a theme:
   - Create a new theme with the command "hugo new theme <THEMENAME>"
   - Or, install a theme from https://themes.gohugo.io/
3. Edit hugo.toml, setting the "theme" property to the theme name.
4. Create new content with the command "hugo new content <SECTIONNAME>/<FILENAME>.<FORMAT>".
5. Start the embedded web server with the command "hugo server --buildDrafts".

See documentation at https://gohugo.io/.
root@hugocms:~# *
```

➔ cd monsite

```
root@hugocms:~# cd monsite/
root@hugocms:~/monsite#
```

Pour pouvoir installer un thème depuis GitHub, suivez les commandes suivantes :

➔ Git init

```
root@hugocms:~/monsite# git init
astuce: Utilisation de 'master' comme nom de la branche initiale. Le nom de la branche
astuce: par défaut peut changer. Pour configurer le nom de la branche initiale
astuce: pour tous les nouveaux dépôts, et supprimer cet avertissement, lancez :
astuce:
astuce:         git config --global init.defaultBranch <nom>
astuce:
astuce: Les noms les plus utilisés à la place de 'master' sont 'main', 'trunk' et
astuce: 'development'. La branche nouvellement créée peut être rénommée avec :
astuce:
astuce:         git branch -m <nom>
Dépôt Git vide initialisé dans /root/monsite/.git/
root@hugocms:~/monsite#
```

➔ git submodule add https://github.com/google/docsy.git themes/docsy

```
root@hugocms:~/monsite# git submodule add https://github.com/google/docsy.git themes/docsy
Clonage dans '/root/monsite/themes/docsy'...
remote: Enumerating objects: 11741, done.
remote: Counting objects: 100% (1063/1063), done.
remote: Compressing objects: 100% (518/518), done.
remote: Total 11741 (delta 656), reused 821 (delta 466), pack-reused 10678 (from 1)
Réception d'objets: 100% (11741/11741), 9.34 Mio | 4.80 Mio/s, fait.
Résolution des deltas: 100% (7404/7404), fait.
root@hugocms:~/monsite#
```

➔ echo "theme = 'docsy'" >> hugo.toml

```
root@hugocms:~/monsite# echo "theme = 'docsy'" >> hugo.toml
root@hugocms:~/monsite#
```

➔ git submodule add https://github.com/twbs/bootstrap.git themes/github.com/twbs/bootstrap

```
root@hugocms:~/monsite# git submodule add https://github.com/twbs/bootstrap.git themes/github.com/twbs/bootstrap
Clonage dans '/root/monsite/themes/github.com/twbs/bootstrap'...
remote: Enumerating objects: 201873, done.
remote: Counting objects: 100% (691/691), done.
remote: Compressing objects: 100% (346/346), done.
Réception d'objets:  19% (38356/201873), 45.95 Mio | 22.97 Mio/s
```

➔ git submodule add https://github.com/FortAwesome/Font-Awesome.git themes/github.com/FortAwesome/Font-Awesome

```
root@hugocms:~/monsite# git submodule add https://github.com/FortAwesome/Font-Awesome.git  themes/github.com/FortAwesome/font-awesome
Clonage dans '/root/monsite/themes/github.com/FortAwesome/font-awesome'...
remote: Enumerating objects: 87765, done.
remote: Counting objects: 100% (12255/12255), done.
remote: Compressing objects: 100% (2346/2346), done.
Réception d'objets:  75% (66225/87765), 79.39 Mio | 17.54 Mio/s
```

➔ git submodule update --init –recursive

```
root@hugocms:~/monsite# git submodule update --init --recursive
root@hugocms:~/monsite#
```

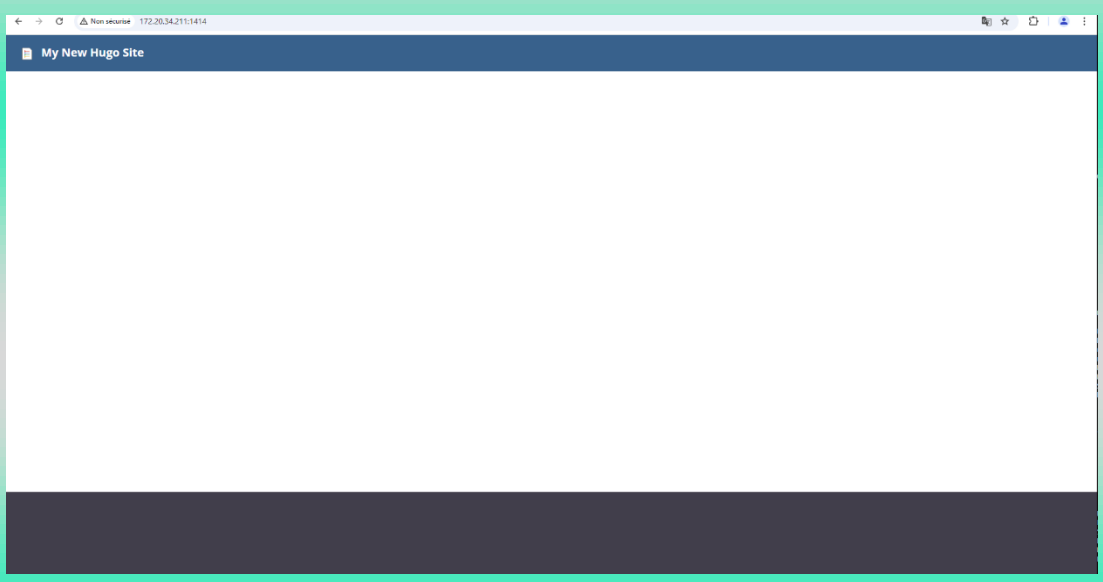
➔ hugo server --port=1414 --bind=0.0.0.0

```
root@hugocms:~/monsite# hugo server --port=1414 --bind=0.0.0.0
Watching for changes in /root/monsite/{archetypes,assets,content,data,i18n,layouts,static,themes}
Watching for config changes in /root/monsite/hugo.toml, /root/monsite/themes/docsy/hugo.yaml, /root/monsite/themes/github.com/twbs/bootstrap/hugo.yml
Start building sites …
hugo v0.135.0-f30603c47f5205e30ef83c70419f57d7eb7175ab+extended linux/amd64 BuildDate=2024-09-27T13:17:08Z VendorInfo=gohugoio


                   | EN
-------------------+-----
  Pages            |  8
  Paginator pages  |  0
  Non-page files   |  0
  Static files     | 30
  Processed images |  0
  Aliases          |  0
  Cleaned          |  0

Built in 1472 ms
Environment: "development"
Serving pages from disk
Running in Fast Render Mode. For full rebuilds on change: hugo server --disableFastRender
Web Server is available at http://localhost:1414/ (bind address 0.0.0.0)
```

Une fois votre thème installé, vous pouvez accéder à votre site web :

➔ http://VotreIp:1414/

Exemple :



Pour modifier votre site, vous pouvez accéder au répertoire `public` :

```
root@hugocms:~/monsite/public# ls
404.html  categories  css  favicons  index.html  index.xml  js  scss  sitemap.xml  tags  webfonts
root@hugocms:~/monsite/public#
```

OBJECTIF : Cette section de la procédure vise expliquer la mise en place de Wordpress

MODE OPÉRATOIRE :

Configuration

Installer **MariadB** :

https://mariadb.org/download/?t=repo-config&d=Debian+12+%22Bookworm%22&v=11.4&r_m=icam

- apt-get install apt-transport-https curl
- mkdir -p /etc/apt/keyrings
- curl -o /etc/apt/keyrings/mariadb-keyring.pgp 'https://mariadb.org/mariadb_release_signing_key.pgp'
- nano /etc/apt/sources.list.d/mariadb.sources

Contenu du fichier mariadb.sources.

- apt update
- apt install mariadb-server -y

Créer la **base de données** et l'**utilisateur** pour WordPress :

- mariadb -u root
- CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'wordpresspassword';
- CREATE DATABASE wordpress;
- GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'localhost';
- FLUSH PRIVILEGES;
- EXIT;

Note : Sécuriser l'accès à la base de données en mettant un mot de passe fort (Recommandation CNIL)

Installer **Nginx** :

Installer les paquets nécessaires :

- ➔ apt install curl gnupg2 ca-certificates lsb-release debian-archive-keyring

Télécharger et installer la clé de signature en exécutant :

- ➔ curl https://nginx.org/keys/nginx_signing.key | gpg --dearmor | tee /usr/share/key-rings/nginx-archive-keyring.gpg >/dev/null

Vérifier la clé téléchargée avec la commande suivante :

- ➔ gpg --dry-run --quiet --no-keyring --import --import-options import-show /usr/share/keyrings/nginx-archive-keyring.gpg

Résultat attendu :

pub rsa2048 2011-08-19 [SC] [expires: 2027-05-24]

573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62

uid nginx signing key <signing-key@nginx.com>

Notez que la sortie peut contenir d'autres clés utilisées pour signer les paquets.

Pour utiliser la version stable de Nginx, exécuter :

- ➔ echo "deb [signed-by=/usr/share/keyrings/nginx-archive-keyring.gpg] http://nginx.org/packages/debian lsb_release -cs nginx" | tee /etc/apt/sources.list.d/nginx.list

Pour utiliser la version mainline, exécuter :

- ➔ echo "deb [signed-by=/usr/share/keyrings/nginx-archive-keyring.gpg] http://nginx.org/packages/mainline/debian lsb_release -cs nginx" | tee /etc/apt/sources.list.d/nginx.list

Configurer le système pour préférer les paquets du dépôt nginx.org avec la commande :

- ➔ echo -e "Package: *\nPin: origin nginx.org\nPin: release o=nginx\nPin-Priority: 900\n" | tee /etc/apt/preferences.d/99nginx

Mettre à jour la liste des paquets :

- ➔ apt update

Installer Nginx :

- ➔ apt install nginx

Vérifier la version installée de Nginx :

- ➔ nginx -v

Installer **WordPress** pour **Nginx** :

- ➔ mkdir -p /usr/share/nginx/html/WP

Sur le site officiel de WordPress, vous pouvez télécharger les dernières versions de Word-Press. Téléchargez l'archive dans le répertoire /tmp. Vous pouvez accéder à ce répertoire en utilisant cd, et télécharger l'archive en utilisant wget :

- ➔ cd /tmp
- ➔ wget https://wordpress.org/latest.tar.gz

Extraire cette archive dans le répertoire créé précédemment. Ceci peut être fait en utilisant :

- ➔ tar xf latest.tar.gz

Déplacer le contenu de WordPress dans « html » de nginx :

- ➔ mv /tmp/wordpress/* /usr/share/nginx/html/WP/

Configurer la connexion à la base de données pour WordPress. Copier le fichier de base :

- ➔ cp /usr/share/nginx/html/WP/wp-config-sample.php /usr/share/nginx/html/WP/wp-config.php

Editer le fichier.

➔ nano /usr/share/nginx/html/WP/wp-config.php

Remplacer la connexion à la BDD par l'utilisateur créer précédemment.

```
define( 'DB_NAME', 'wordpress' );

define( 'DB_USER', 'wordpress' );

define( 'DB_PASSWORD', 'wordpresspassword' );

define( 'DB_HOST', 'localhost' );
```

Le serveur web aura besoin d'un accès complet à ces fichiers. Modifier l'autorisation en utilisant :

➔ chown -R nginx /usr/share/nginx/html/WP
➔ find /usr/share/nginx/html -type d -exec chmod 755 {} \;
➔ find /usr/share/nginx/html -type f -exec chmod 644 {} \;

Notez que Nginx fonctionne avec l'utilisateur et le groupe nginx, mais PAS PHP. PHP fonctionne avec l'utilisateur et le groupe www-data.

Pour vérifier les services faire :

➔ ps aux

USER : L'utilisateur qui exécute le processus.

PID : L'ID du processus.

Installer **PHP** :

➔ apt install php8.2 php8.2-cli php8.2-fpm php8.2-mysql php8.2-curl -y

Note : Dépôt utile LEMP/LAMP https://deb.sury.org/

Changer l'utilisateur et le groupe pour PHP8.2 :

➔ nano /etc/php/8.2/fpm/pool.d/www.conf

Modifier dans le fichier les lignes suivante :

user = nginx

group = nginx

listen.user = nginx

listen.group = nginx

Redémarrer le service PHPfpm.

➔ systemctl restart php8.2-fpm.service

Configurer **Nginx** pour **WordPress** :

Si l'utilisateur de nginx n'est pas « nginx » alors le changer :

➔ nano /etc/nginx/nginx.conf

```
user  nginx;
worker_processes  auto;
```

Pour configurer Nginx pour WordPress, nous devons créer un nouveau bloc serveur pour notre installation WordPress :

➔ nano /etc/nginx/conf.d/wordpress.conf

Si vous utilisez https importer ce fichier de configuration, sinon en http celui-ci.

Renommer le fichier de configuration par défaut :

➔ mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf.disabled

Ajouter le dossier et le fichier de configuration de fastcgi pour Nginx :

➔ mkdir -p /etc/nginx/snippets
➔ nano /etc/nginx/snippets/fastcgi-php.conf

Contenu du fichier « fastcgi-php.conf ».

Valider la configuration de Nginx en utilisant :

➔ nginx -t

Redémarrer Nginx :

➔ systemctl restart nginx

Question 1 :

Jekyll est un générateur de sites statiques conçu pour transformer des fichiers de texte brut, comme ceux en Markdown, en pages HTML prêtes à être hébergées. Il est souvent utilisé pour créer des blogs, des portefeuilles, ou des sites de documentation technique. Contrairement aux sites dynamiques, Jekyll ne nécessite pas de base de données ni de serveur pour exécuter des scripts. Il génère des fichiers HTML statiques qui sont rapides à charger et faciles à déployer.

Un des grands avantages de Jekyll est sa simplicité de déploiement. Il s'intègre parfaitement avec GitHub Pages, offrant un hébergement gratuit et automatique dès que vous mettez à jour votre dépôt Git. De plus, les sites statiques étant plus sécurisés, Jekyll réduit les risques de failles souvent liées aux CMS dynamiques comme WordPress. Il offre également un contrôle total sur la personnalisation, avec la possibilité de créer des layoutssur mesure en utilisant le langage de templating Liquid.

Cependant, Jekyll présente aussi des limites. Sa nature statique signifie qu'il ne peut pas gérer nativement des fonctionnalités dynamiques comme des commentaires ou des formulaires interactifs. Cela impose de recourir à des services externes pour ces besoins. De plus, son utilisation peut être complexe pour des non-développeurs, car elle repose sur des outils comme Git et la ligne de commande. Enfin, pour des sites volumineux, le temps de compilation peut devenir long, ce qui ralentit les mises à jour.

Ainsi, Jekyll est un outil puissant et efficace pour des projets où la simplicité, la rapidité et la sécurité sont primordiales, mais il peut être moins adapté aux sites nécessitant beaucoup de contenu dynamique.

Question 2 :

Expliquer le rôle de ces logiciels Eleventy et Gatsby avec leurs avantages/inconvénients puis comparez-les avec le logiciel retenu HUGO :

Eleventy est un générateur de site statique qui permet de créer des sites web à partir de modèles et de contenus. Il est très flexible et s'adapte à différents formats de fichiers, tels que Markdown, HTML, et bien d'autres.

Avantages :

- **Simplicité :** Facile à configurer et à utiliser, idéal pour les petits projets.
- **Flexibilité :** Supporte plusieurs formats de fichiers, ce qui permet une grande liberté dans la structure du contenu.
- **Pas de dépendances lourdes :** Ne nécessite pas de framework JavaScript lourd.

Inconvénients :

- **Moins de fonctionnalités avancées :** Par rapport à d'autres générateurs comme Gatsby, il peut manquer de certaines intégrations et fonctionnalités avancées.

Gatsby est un générateur de site statique basé sur React. Il est optimisé pour créer des applications web modernes et rapides, souvent utilisées pour des sites plus complexes.

Avantages :

- **Performance :** Génère des sites très rapides grâce à un rendu côté serveur et à l'optimisation automatique des ressources.
- **Écosystème riche :** Large choix de plugins et d'intégrations avec des CMS et des API tierces.
- **Réactivité :** Idéal pour les sites nécessitant une interactivité grâce à son utilisation de React.

Inconvénients :

- **Complexité :** Peut-être plus difficile à prendre en main pour les développeurs novices.

Hugo se distingue des autres générateurs de site statique comme Eleventy et Gatsby par sa rapidité et sa simplicité d'utilisation. Contrairement à Gatsby qui est basé sur React et peut être complexe à configurer Hugo permet une prise en main rapide. Bien qu'Eleventy offre une flexibilité intéressante avec plusieurs formats de fichiers, Hugo excelle par sa performance, même pour les sites de grande taille. Alors qu'Hugo se concentre sur la génération de sites statiques rapides, Gatsby est meilleur dans les projets nécessitant une interactivité poussée.

Question 3 :

A votre avis, pourquoi c'est le logiciel HUGO qui a été retenu plutôt que Eleventy ou Gatsby ? Justifiez :

Le logiciel HUGO a été retenu par rapport à Eleventy et Gatsby, car Corporate LM vise à offrir des outils de gestion informatique "simples, performants et adaptés". HUGO répond parfaitement à ces exigences de simplicité et d'efficacité sans nécessiter de compétences spécialisées dans des frameworks comme React. De plus, l'entreprise utilise Ubuntu pour ses équipes de développeurs, et l'intégration de HUGO dans un environnement virtualisé sous Linux est fluide et naturelle, ce qui en fait un choix plus pertinent. Par ailleurs, l'adaptation de HUGO sur un serveur Debian est également facilitée.

Dans le contexte de Corporate LM, l'utilisation de Jekyll sur le serveur montre que l'entreprise est déjà familière avec les générateurs de sites statiques. HUGO est souvent considéré comme une alternative plus performante à Jekyll, particulièrement pour les développeurs cherchant à explorer de nouveaux outils de génération statique. Ainsi, l'adoption de HUGO s'inscrit naturellement dans la continuité de l'expérience acquise avec Jekyll.

Question 6 :

Qu'appelle-t-on une infrastructure LEMP :

Une infrastructure **LEMP** est un ensemble de logiciels open source utilisés pour héberger des applications web, souvent des sites dynamiques ou des systèmes de gestion de contenu (CMS) comme WordPress. L'acronyme LEMP représente les éléments principaux de cette infrastructure :

L – **Linux** : Le système d'exploitation sur lequel l'infrastructure est installée. Linux est un choix populaire pour les serveurs en raison de sa stabilité, de sa sécurité et de son faible coût.

E – **Nginx** (prononcé "Engine-X") : Le serveur web qui remplace souvent Apache dans cette configuration. Nginx est apprécié pour sa capacité à gérer un grand nombre de connexions simultanées tout en consommant peu de ressources, le rendant efficace pour des environnements à fort trafic.

M – **MySQL** (ou MariaDB) : Le système de gestion de bases de données relationnelles, utilisé pour stocker et gérer les données des applications web. MySQL est très utilisé dans le développement web et s'intègre bien avec des outils comme WordPress.

P – **PHP** : Le langage de programmation utilisé pour traiter les requêtes côté serveur et générer des pages web dynamiques. PHP est très largement utilisé dans les CMS et applications web.

Question 7 :

Quels sont les avantages/inconvénients de Nginx par rapport à Apache :

Avantages de Nginx :

1. **Meilleure gestion des connexions simultanées** : Grâce à son architecture événementielle, Nginx gère mieux un grand nombre de connexions simultanées qu'Apache.
2. **Faible consommation de ressources** : Il utilise moins de mémoire et de CPU, ce qui le rend plus adapté aux sites à fort trafic.
3. **Excellent serveur proxy** : Idéal pour la répartition de charge et la mise en cache dans des architectures complexes.
4. **Performance pour le contenu statique** : Nginx est plus rapide pour servir des fichiers statiques (images, CSS, etc.).
5. **Configuration simple** : Les fichiers de configuration de Nginx sont plus concis et faciles à comprendre.

Inconvénients de Nginx :

1. **Moins de modules** : Nginx offre moins de modules natifs qu'Apache, ce qui peut limiter les fonctionnalités avancées.
2. **Pas de gestion des fichiers .htaccess** : Nginx ne supporte pas .htaccess, donc la configuration doit être centralisée.
3. **Gestion dynamique moins directe** : Nginx nécessite un serveur externe (comme PHP-FPM) pour exécuter des scripts dynamiques.
4. **Documentation et support** : Apache bénéficie d'une documentation plus complète et d'une communauté plus établie.

Conclusion :

- **Nginx** est idéal pour des sites à fort trafic avec beaucoup de contenu statique et un besoin de performance élevée.
- **Apache** est plus adapté aux configurations complexes, aux applications dynamiques et aux environnements nécessitant une flexibilité avec .htaccess.

Question 10 :

Qu'est-ce que l'OWASP (Open Web Application Security Project) :

L'**OWASP** (Open Web Application Security Project) est une organisation internationale à but non lucratif dédiée à la **sécurité des applications web**. Son objectif principal est de sensibiliser les développeurs, les entreprises et les utilisateurs à l'importance de la sécurité dans le développement des applications et de fournir des ressources, des outils et des bonnes pratiques pour protéger les applications contre les menaces en ligne.

Question 11 :

Quel est le but du projet nommé : Top Ten OWASP :

Le **but du projet OWASP Top Ten** est de sensibiliser et d'éduquer les développeurs, les professionnels de la sécurité et les organisations sur les principales **vulnérabilités de sécurité** dans les applications web. Le projet fournit une **liste des 10 failles de sécurité les plus critiques** que l'on retrouve fréquemment dans les applications web, accompagnée de recommandations pour prévenir ces vulnérabilités.

Question 12 :

Quel est le but de ces deux projets :

OWASP WebScarab NG :

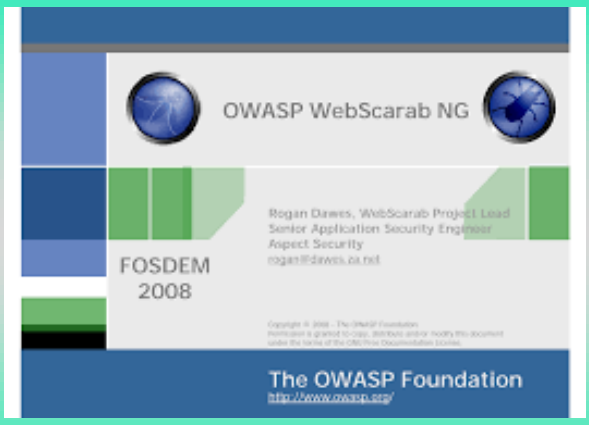
Le projet **OWASP WebScarab NG** est un outil conçu pour analyser et manipuler les communications HTTP/HTTPS entre un navigateur et un serveur web. Son but principal est d'aider les professionnels de la sécurité à identifier et corriger les failles potentielles dans les applications web. WebScarab NG permet de capturer et d'inspecter les requêtes et réponses échangées, et même de les modifier en temps réel pour tester la résistance de l'application face à différentes attaques, comme les injections SQL ou la manipulation des sessions. Cet outil est particulièrement utile pour réaliser des **tests manuels et interactifs**, offrant une flexibilité maximale pour auditer et renforcer la sécurité des échanges web. Grâce à sa capacité à modifier le trafic web, WebScarab NG est un atout important dans le processus de sécurisation des applications.

OWASP WebGoat :

Le projet **OWASP WebGoat** est une plateforme éducative sous forme d'application web volontairement vulnérable, spécialement conçue pour **apprendre la sécurité des applications web**. WebGoat permet aux développeurs et aux experts en sécurité de s'exercer à identifier, exploiter, puis corriger des vulnérabilités dans un environnement contrôlé. À travers des **scénarios réalistes d'attaques**, les utilisateurs peuvent pratiquer des techniques offensives et défensives sur des failles courantes comme l'injection SQL, le Cross-Site Scripting (XSS), et bien d'autres. Cette approche interactive permet aux participants de comprendre l'impact des vulnérabilités tout en apprenant à les corriger, faisant de WebGoat une ressource précieuse pour la formation en cybersécurité et la sensibilisation aux risques web.

Lien Vers WebGoat :

➔ https://owasp.org/www-project-webgoat/



Question 13 :

Qu’est-ce que HackerOne :

HackerOne propose des programmes de bug bounty, permettant aux entreprises de récompenser financièrement les hackers éthiques pour la découverte de vulnérabilités. La plateforme offre un environnement sécurisé pour soumettre et examiner ces découvertes, facilitant la communication entre les hackers et les entreprises. Les rapports soumis incluent des détails sur les vulnérabilités, aidant les équipes de sécurité à les traiter efficacement. HackerOne regroupe une communauté diversifiée de hackers et fournit des outils d'analyse pour suivre Consultez :

➔ HackerOne

Question 14 :

A ce propos, qu’est-ce qu’un livre blanc :

Un **livre blanc** est un document qui présente une analyse approfondie d'un sujet, une technologie ou une tendance spécifique. Il est souvent utilisé pour partager des connaissances, expliquer des solutions techniques, ou mettre en avant des produits et services. En fournissant des données, des études de cas et des recommandations, un livre blanc vise à informer les lecteurs et à établir la crédibilité de l'entreprise ou de l'organisation qui le publie. C'est un outil essentiel pour le marketing de contenu et la génération de leads.

Question 15 :

Qu’est-ce que le Syntec Informatique :

Syntec Informatique est une organisation professionnelle en France qui regroupe des entreprises du secteur des technologies de l'information et de la communication (TIC). Elle représente les intérêts de ses membres auprès des institutions, des pouvoirs publics et des autres acteurs économiques. Syntec Informatique œuvre également pour le développement et la promotion de bonnes pratiques dans le secteur, notamment à travers des publications, des études et des groupes de travail sur des thématiques telles que le **Green IT** et l'innovation technologique.

Question 16 :

Qu’est-ce qu’un livre vert :

Un **livre vert** est un document qui présente des réflexions, des analyses ou des propositions sur un sujet particulier, souvent dans le but de susciter des débats ou de recueillir des avis. Dans le contexte des politiques publiques ou des initiatives sectorielles, il sert à informer et à consulter les parties prenantes sur des questions importantes. Contrairement à un livre blanc, qui est plus orienté vers la promotion de solutions, un livre vert a un caractère plus exploratoire et participatif.

Question 17 :

Qu’appelle-t-on le Green IT :

Le **Green IT** désigne l'ensemble des pratiques et des technologies visant à réduire l'impact environnemental des technologies de l'information. Cela inclut l'efficacité énergétique des centres de données, l'utilisation de matériels écologiques, la gestion responsable des déchets électroniques et la promotion de solutions numériques durables. L'objectif du Green IT est de minimiser la consommation d'énergie et les émissions de gaz à effet de serre tout en maximisant l'efficacité des systèmes informatiques. C'est une approche essentielle pour intégrer la durabilité dans le secteur technologique.

Question 18 :

Qu’est-ce qu’un livre bleu :

Un **livre bleu** est un document de référence qui synthétise les recommandations, analyses et bonnes pratiques sur un sujet spécifique, souvent lié à la sécurité numérique. Dans le cadre des **Assises de la Sécurité**, ces livres bleus visent à informer les entreprises et les décideurs sur les enjeux de la cybersécurité et à proposer des solutions concrètes pour améliorer la sécurité des systèmes d'information. Ils servent également de guide pour la mise en œuvre de stratégies de sécurité efficaces.

Consultez :

➔ Livre Bleu 2011.

Question 19 :

Qu’est-ce que les Assises ?

Les **Assises de la Sécurité** sont un événement annuel en France qui rassemble des professionnels de la cybersécurité, des entreprises et des décideurs pour discuter des enjeux et des défis liés à la sécurité numérique. Ce forum permet d'échanger des idées, de partager des expériences et de présenter des solutions innovantes pour améliorer la protection des systèmes d'information. Les Assises sont également l'occasion de publier des recommandations et des bonnes pratiques, souvent présentées dans des **livres bleus**.

Question 20 :

Expliquez l'intérêt et le rôle de ces 3 outils dans votre infrastructure :

Fail2Ban	Iptables	Portsentry
Fail2Ban est un outil de sécurité qui surveille les fichiers journaux et bloque les adresses IP suspectes après un certain nombre de tentatives de connexion infructueuses. Il protège ainsi votre site WordPress contre les attaques par force brute en ajoutant temporairement des règles dans le pare-feu pour interdire les IP malveillantes.	Iptables est un outil de filtrage de paquets qui permet de configurer les règles de pare-feu sur un serveur Linux. Il permet de contrôler le trafic entrant et sortant, en définissant des règles spécifiques pour autoriser ou bloquer certaines connexions. Cela renforce la sécurité de votre serveur en limitant les accès non autorisés.	Portsentry est un outil de détection d'intrusion qui surveille les ports ouverts sur votre serveur. En analysant les connexions suspectes, il peut automatiquement bloquer les adresses IP qui tentent de scanner ou d'accéder à des ports non sécurisés. Cela aide à prévenir les attaques en signalant et en neutralisant les activités malveillantes sur votre serveur.

Question 21 :

Mettre en œuvre ces 3 outils avec la documentation technique associée :

Pour pouvoir installer iptables vous devez tout d'abord mettre en place IP table avec :

➔ Apt install iptable

Pour pouvoir mettre en place Portsentry :

➔ Apt install portsentry

Pour pouvoir autoriser les ports :

➔ nano /etc/portsentry/portsentry.conf

Puis ajouter les ports que vous souhaitez :

➔

```
#
# Use these if you just want to be aware:
TCP_PORTS="80"
UDP_PORTS="80"
#
```

Par défaut, Portsentry ajoute l'adresse IP d'un attaquant dans `/etc/hosts.deny`. Assurez-vous que l'option suivante est activée pour bloquer l'accès :

➔ KILL_ROUTE="/sbin/iptables -I INPUT -s $TARGET$ -j DROP"

```
root@debian12CLI:~# KILL_ROUTE="/sbin/iptables -I INPUT -s $TARGET$ -j DROP"
root@debian12CLI:~#
```

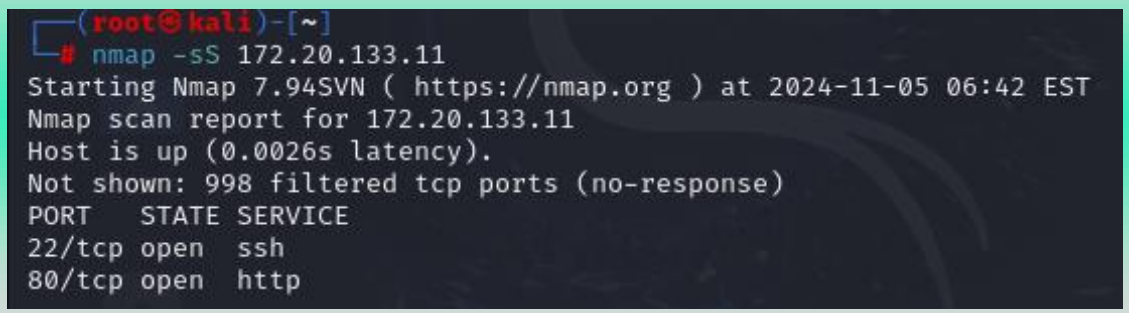
Pour pouvoir ingorer les ip de votre réseau local :

➔ nano /etc/portsentry/portsentry.ignore

```
# /etc/portsentry/portsentry.ignore: Contains all IPs portsentry(8)
#                                    will never block.
#
# This file was generated by /usr/lib/portsentry/portsentry-build-ignore-fi
# DO NOT EDIT - edit /etc/portsentry/portsentry.ignore.static instead and u
# "/etc/init.d/portsentry restart" to reload the configuration.

# IPs from /etc/portsentry/portsentry.ignore.static:
127.0.0.1/32
0.0.0.0
```

Une fois cela fait vous pouvez vérifier avec l'outil nmap que seul les ports 80 et 22 sont ouverts :

➔



Pour pouvoir installer **Fail2Ban** :

➔ apt install fail2ban

Configurer fail2ban pour protéger wordpress, créer un filtre pour WordPress :

➔ nano /etc/fail2ban/filter.d/wordpress.conf

Contenu du fichier pour détecter les tentatives de connexion échouées via wp-login.php.

Créer le fichier jail.local pour configurer une jail spécifique à WordPress via les logs Nginx :

➔ nano /etc/fail2ban/jail.local

Contenu du fichier pour définir la jail WordPress et adapter le chemin des logs.

Explications :

enabled : Active la jail.

filter : Utilise le filtre wordpress.conf créé plus haut.

action : Spécifie qu'on utilisera iptables pour bloquer les IP.

logpath : Chemin des logs d'accès Nginx spécifiques à WordPress (adaptez si besoin).

maxretry : Nombre de tentatives avant qu'une IP soit bloquée.

bantime : Durée du bannissement en secondes (ici, 1 heure).

Note : On désactive fail2ban pour SSH (sshd) car ce n'est pas notre but dans cette documentation.

Appliquer les permissions sur le fichier :

➔ chmod 755 /var/log/nginx
➔ chown root:adm /var/log/nginx/wordpress.access.log

Redémarrer Fail2Ban :

➔ systemctl restart fail2ban

Vérifiez que Fail2Ban surveille bien la jail WordPress avec :

➔ fail2ban-client status wordpress

Cela vous montrera le statut de la jail, le nombre d'IP bannies, etc.

Pour voir les logs de Fail2Ban et surveiller les bannissements, utiliser :

➔ tail -f /var/log/fail2ban.log

Pour débannir une adresse IP dans la jail WordPress utiliser :

➔ fail2ban-client set wordpress unbanip 192.168.1.100

Syntaxe : fail2ban-client set <jail-name> unbanip <IP>

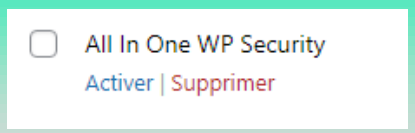
Question 22 :

Parcourez le « catalogue » des extensions Wordpress et mettez en place le WAF de votre choix sous forme de plugin Wordpress.

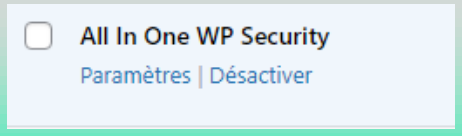
Utilisation du WAF **All In One Security**.

Pour cela, installer le plugin All In One Security sur votre WordPress.

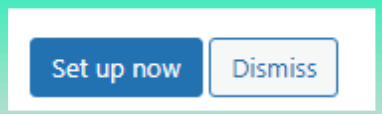
Activer l'extension.



Allez dans paramètres.



Mettre en place le pare-feu basé sur PHP en cliquant sur « Set up now ».



Le pare-feu de base est activé avec les paramètres recommandés. Cependant, ce niveau de protection est générique et vous pouvez encore affiner la configuration pour une sécurité optimale.

Bien que le pare-feu soit activé par défaut, il est conseiller configurer d'autres options pour renforcer davantage la sécurité du site.

1. Paramètres de sécurité utilisateur :

Allez dans WP Security > User Security :

Activez la protection Login Lockout pour bloquer les adresses IP après plusieurs tentatives de connexion échouées.

2. Protection contre les attaques par force brute

Allez dans WP Security > Brute Force :

Activez l'option Rename Login Page URL pour éviter que les attaquants accèdent à votre page de connexion par défaut (wp-login.php).

3. Vérification des fichiers critiques

Allez dans WP Security > File Security :

Utilisez cet outil pour vérifier les permissions de fichiers et de dossiers critiques.

Corrigez les autorisations si nécessaire (par exemple, wp-config.php doit être en 440 ou 400 pour une sécurité maximale).

Note : Dans notre cas il doit être en 640.

Il se peut que AIOS n'est pas les permissions, alors dans ce cas le faire manuellement :

➔ chmod 640 /usr/share/nginx/html/WP/wp-config.php

4. Activer la protection des bases de données

Allez dans WP Security > Database Security :

Modifiez le préfixe par défaut de votre base de données (wp_) pour un préfixe personnalisé afin de compliquer les attaques par injection SQL.

5. Scanner votre site pour détecter les vulnérabilités

Allez dans WP Security > Scanner :

Utilisez l'outil Scanner intégré pour rechercher des malwares, des fichiers infectés, ou des modifications suspectes.

*A noter que ce n'est pas une liste exhaustive, sur chacune des fonctionnalités proposées par AIOS, il est disposé un bouton « **More info** » pour l'explication de l'option.*

Question 23 :

Quel est le but de l'outil nmap ? Que permet-il dans ce contexte ?

Nmap (Network Mapper) est un outil open source essentiel pour l'analyse et la sécurité des réseaux. Il permet de scanner les réseaux pour identifier les hôtes actifs, les ports ouverts, et les services en cours d'exécution, offrant une vue d'ensemble de l'infrastructure réseau. Utilisé en cybersécurité, Nmap aide à détecter les vulnérabilités en identifiant les ports et services non sécurisés, permettant ainsi de prévenir les cyberattaques potentielles. Cet outil est aussi employé dans les tests de pénétration, où il simule des attaques pour évaluer la robustesse des systèmes. En résumé, Nmap est une ressource clé pour les administrateurs et les analystes de sécurité, permettant de protéger les réseaux en renforçant leur sécurité.

Il permet :

- Effectuer **des audits de sécurité proactive** : en identifiant les failles potentielles avant qu'elles ne soient exploitées.
- Surveiller **l'activité du réseau** : en maintenant un inventaire des hôtes et des services actifs

Question 24 :

Quel est le but de l'outil Dirbuster ? Que permet-il dans ce contexte ?

Dirbuster est un outil de sécurité qui sert à découvrir des répertoires et des fichiers cachés sur un serveur web en utilisant une technique de force brute. Il envoie des requêtes HTTP en testant des noms de fichiers et de dossiers communs listés dans une wordlist, afin d'identifier des chemins d'accès non publiés. Dans un contexte de cybersécurité, Dirbuster permet de repérer des répertoires sensibles, comme ceux d'administration ou de sauvegarde, souvent laissés accessibles par erreur. Cet outil est particulièrement utile pour les tests de pénétration, car il aide à simuler des intrusions et à identifier des failles potentielles, renforçant ainsi la sécurité du serveur web.

Question 25 :

Quel est le but de l'outil Nikto ? Que permet-il dans ce contexte ?

Nikto est un outil de scan de vulnérabilités conçu pour analyser les serveurs web et détecter les failles de sécurité. Il identifie les configurations vulnérables, les fichiers sensibles et les versions obsolètes de logiciels qui pourraient être exploités par des attaquants. Grâce à une base de données régulièrement mise à jour, Nikto est capable de repérer les vulnérabilités connues rapidement. Dans un contexte de cybersécurité, il est largement utilisé pour effectuer des tests de pénétration, permettant aux professionnels de la sécurité de simuler des attaques et d'évaluer la sécurité des serveurs web. Cet outil aide donc à anticiper et corriger les failles de sécurité.

Question 26 :

Mettre en œuvre ces trois outils avec les tests fonctionnels. Rédigez la documentation technique associée.

Sur Kali Linux, l'outil Nmap est déjà installé par défaut. Vous pouvez donc directement utiliser les commandes associées :

Effectue un scan de base sur la cible :

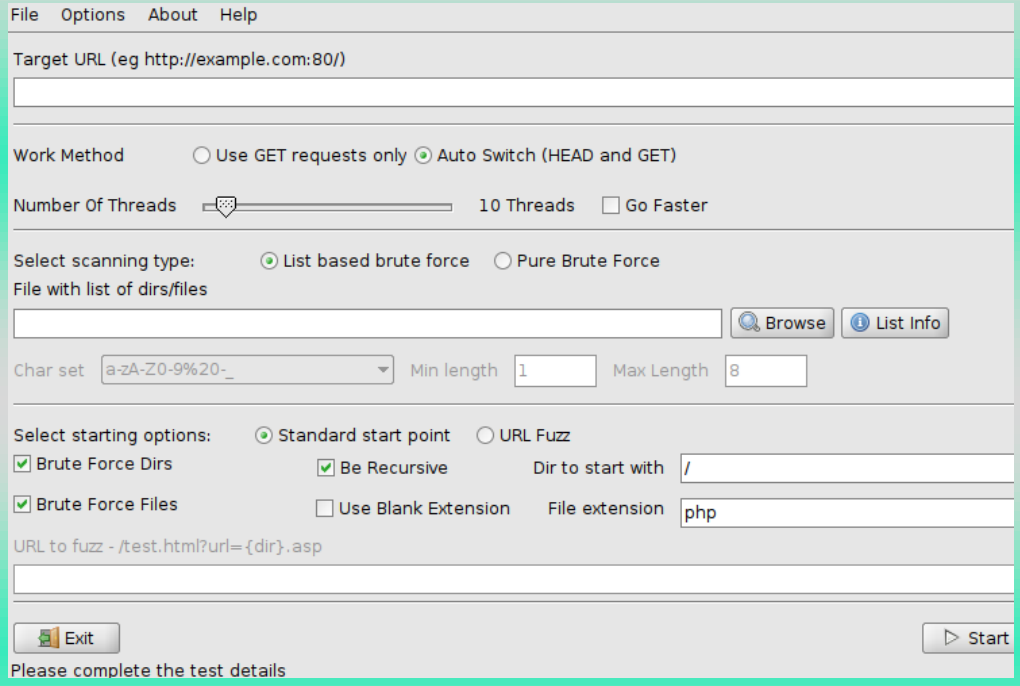
➔ nmap -sn stpaulbb.org

```
┌──(root㉿kali)-[~]
└─# nmap stpaulbb.org
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-04 02:31 EST
Nmap scan report for stpaulbb.org (188.165.61.82)
Host is up (0.0039s latency).
rDNS record for 188.165.61.82: cluster024.hosting.ovh.net
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 5.07 seconds

┌──(root㉿kali)-[~]
└─# nmap -sn stpaulbb.org
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-04 02:46 EST
Nmap scan report for stpaulbb.org (188.165.61.82)
Host is up (0.0016s latency).
rDNS record for 188.165.61.82: cluster024.hosting.ovh.net
Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds
```

Tout comme Nmap, Dirbuster est généralement préinstallé sur Kali Linux.

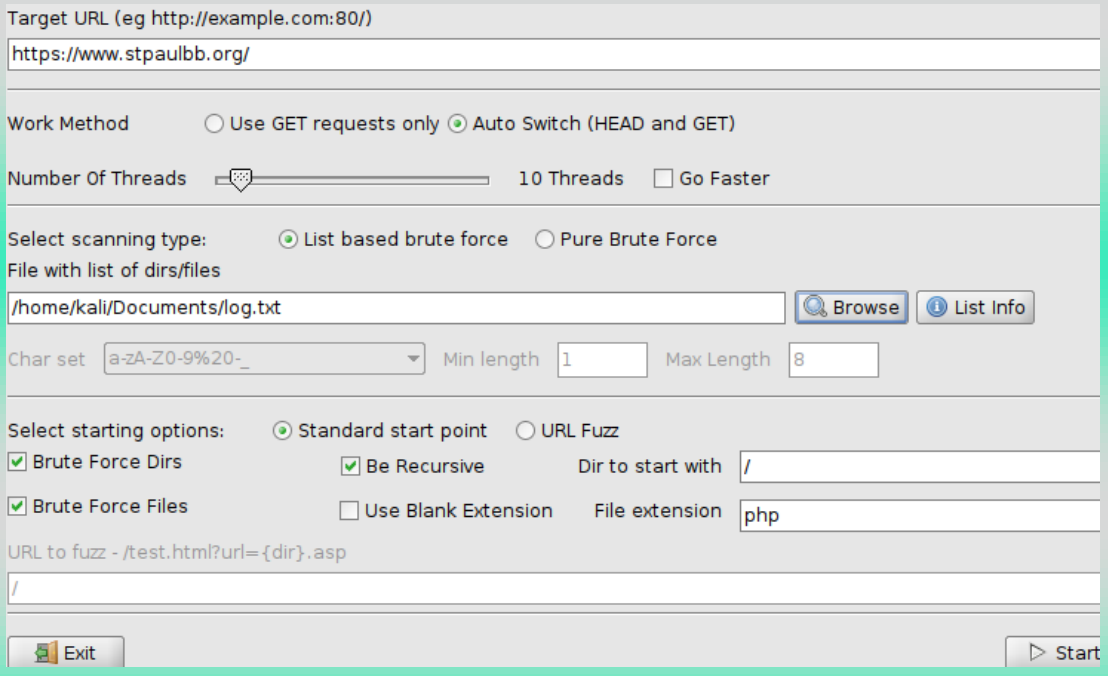
Une fois lancé, vous devriez voir une interface ressemblant à ceci :

➔

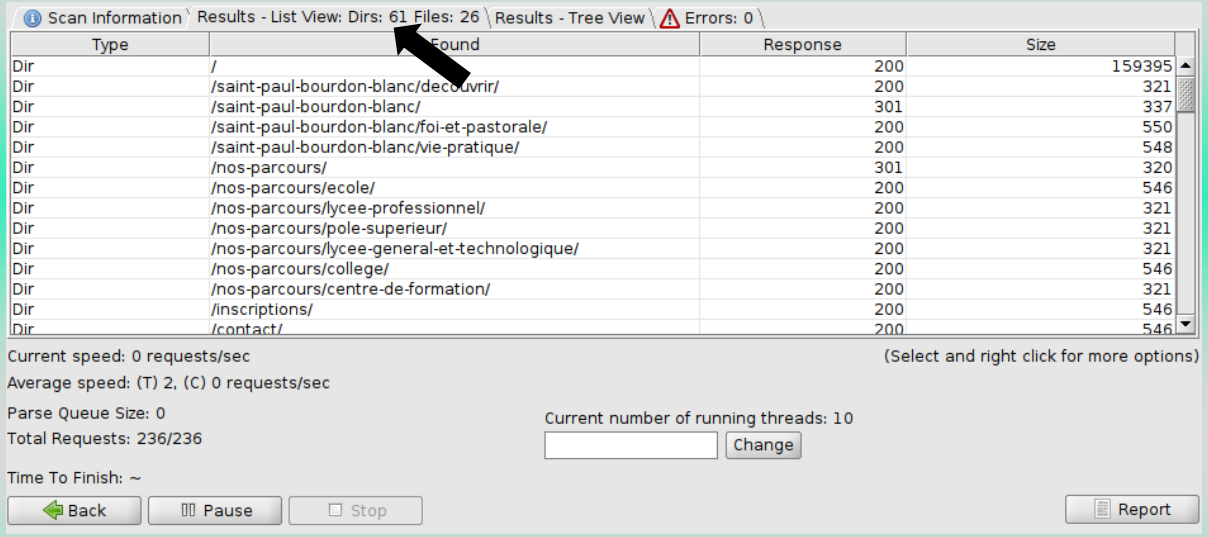


Vous pouvez alors sélectionner votre site web pour pouvoir commencer, avant de lancer l'outil, l'interface devrait ressembler à ceci :

➔



Comme un passage de l'outil peut être assez long, il est possible et très pratique de voir les résultats de recherche live pendant que celle-ci est en cours :



Pour pouvoir utiliser Nikto :

Pour scanner un site :

➔ nikto -h http://example.com

Vous pouvez spécifier un port en rajoutant l'argument :

➔ nikto -h example.com -p 8080

Spécifier le type de scan (-T) :

➔ nikto -h example.com -T 123

-T permet de choisir le type de tests à effectuer :

- 1 : Tests sur les répertoires
- 2 : Tests sur les fichiers
- 3 : Tests sur les vulnérabilités générales

Exemple -T 123 pour effectuer tous les types de tests.

Spécifier un délai entre les requêtes :

➔ nikto -h example.com -pause 2

Afficher les options avancées :

➔ nikto -Help

Question 27 :

On constate qu'avec Nmap, les ports 80 et 443 sont ouverts, ce qui indique que le serveur accepte les connexions HTTP et HTTPS. En utilisant Dirbuster, nous obtenons des informations supplémentaires sur la structure du site, ainsi que sur les répertoires et fichiers accessibles. Dirbuster permet également de détecter certains ports ouverts.

Cependant, pour renforcer la sécurité, des outils comme **Fail2Ban**, **iptables**, et **Portsentry** ont été mis en place. Ces outils rendent plus difficile l'exécution de scans de ports. En effet, **Fail2Ban** bloque automatiquement les adresses IP après plusieurs tentatives de connexion échouées, ce qui empêche les attaquants de continuer à essayer de deviner les mots de passe. De plus, **iptables** permet de filtrer les connexions et de limiter les accès non autorisés, tandis que **Portsentry** détecte et réagit aux tentatives de scan de ports, contribuant ainsi à protéger le système contre les attaques automatisées.

Question 28 :

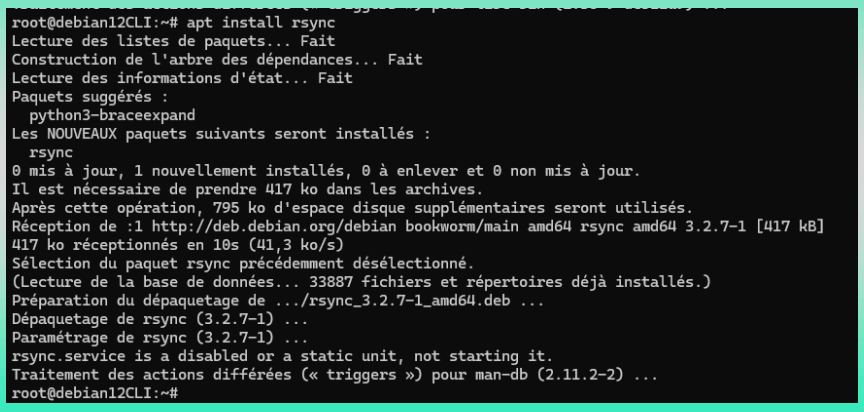
Dans le cadre de la sécurisation de ce serveur il est crucial de sauvegarder plusieurs éléments pour assurer une protection continue et permettre une récupération rapide en cas d'incident. Tout d'abord les fichiers de configuration de sécurité notamment ceux de Fail2Ban, iptables, et Portsentry doivent être régulièrement sauvegardés. Ces fichiers contiennent des règles essentielles pour bloquer les accès non autorisés et détecter les scans de ports ; leur perte ou corruption affaiblirait considérablement la sécurité du serveur. Ensuite, les journaux système et de sécurité sont également des éléments critiques à conserver. Les logs tels que ceux générés par Fail2Ban ou Portsentry permettent de retracer les tentatives de connexion suspectes et de détecter d'éventuelles activités malveillantes. En sauvegardant ces fichiers on garantit la possibilité de mener des analyses en cas d'attaque et de comprendre l'origine et le comportement de potentiels intrus. Pour ce faire on peut utiliser des outils de sauvegarde automatique chiffrés et externalisés afin de sécuriser ces données sur un serveur de backup ou dans un environnement cloud protégé.

Question 29 : RSYNC

Pour pouvoir installer RSYNC suivez les instructions suivantes :

- ➔ apt update
- ➔ apt upgrade
- ➔ apt install rsync

(Penser à vérifier si les pacquet d'installation sont a jour sinon prenner la version la plus récente)



Pour copier des fichiers vers un autre serveur ou récupérer des fichiers distants, configurez une connexion SSH sécurisée :

- ➔ rsync -av --delete -e ssh /usr/share/nginx/html/WP/ user@serveur_distant:/backup/documents/

Vous avez également la possibilité d'automatiser les sauvegardes en utilisant des scripts pour les planifier :

Créez un script de sauvegarde :

- ➔ sudo nano /usr/local/bin/backup_script.sh

Ajoutez le contenu suivant au script

- ➔ rsync -av --delete /home/user/documents/ user@serveur_distant:/backup/documents/

Enregistrez et donnez les permissions d'exécution au script :

- ➔ sudo chmod +x /usr/local/bin/backup_script.sh

Enregistrez et donnez les permissions d'exécution au script :

➔ sudo chmod +x /usr/local/bin/backup_script.sh

Ouvrez le fichier cron pour l'édition :

➔ crontab -e

```
root@debian12CLI:~# crontab -e
no crontab for root - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny

Choose 1-2 [1]:
```

Ajoutez la ligne suivante pour exécuter le script tous les jours à 2h du matin :

➔ 0 2 * * * /usr/local/bin/backup_script.sh

Editée par	Tom COELHO / Cylian MENAGE-BIMBEMENT	
Révisée par :	Tom COELHO / Cylian MENAGE-BIMBEMENT	
Suivie par :	Tom COELHO / Cylian MENAGE-BIMBEMENT	
Validée par :	Tom COELHO / Cylian MENAGE-BIMBEMENT	
Date : 11 / 11 / 2024	(Saint Paul Bourdon Blanc	Version : 1