Une meilleure sécurité dans Drupal

Par Marine Gandy & Nicolas Loye

Bienvenue!

Un atelier (en anglais) est disponible ici :

Cloner le dépôt de code pour commencer.

Les étapes de l'atelier et les prérequis sont listés dans le fichier README.

https://gitlab.com/nicoloye/drupal-security-101

NOTRE PRÉSENTATION

Marine Gandy
Devrel - Platform.sh

Twitter - https://twitter.com/mupsigraphy
LinkedIn - https://www.linkedin.com/in/marinegandy
Drupal - https://www.drupal.org/u/mupsi

Nicolas Loye
CTO - Smile

Twitter - https://twitter.com/nicoloye
LinkedIn - https://www.linkedin.com/in/nicolas-loye
Drupal - https://www.drupal.org/u/nicoloye

AVERTISSEMENT

Ceci est une session d'introduction avancée
Il y aura des QR codes et des mots clés en masse
Pas de recette magique ou par défaut

VOUS MAINTENANT

VOUS QUAND ON AURA FINI

REFUS
PAR DÉFAUT

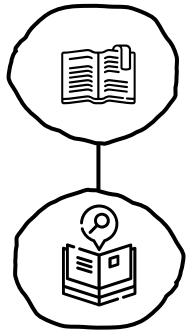
AUCUNE CONFIANCE
DES SAISIES UTILISATEUR

01 - LES BASES

Connaissances de base à propos de
la sécurité dans la communauté

DICTIONNAIRES & LISTES

MITRE



CVE

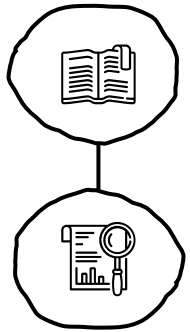
Common Vulnerabilities and Exposures
Dictionnaire d'informations publiques relatives aux vulnérabilités de sécurité

CWE

Common Weakness Enumeration
Liste des vulnérabilités répertoriées dans les logiciels

DICTIONNAIRES & LISTES

ANSSI



CERT-FR

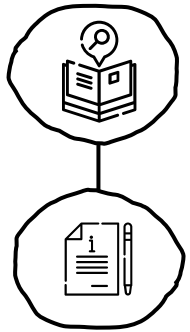
Dictionnaire d'informations publiques relatives aux vulnérabilités de sécurité

Guides

Documents utiles de bonnes pratiques et d'aide à la sécurité dans les SI

NORMES & STANDARDS

OWASP



TOP 10

Liste les menaces les plus fréquentes dans les CWE. Globalement inchangée depuis des années

ASVS

Application Security Verification Standard
3 niveaux de conformité et leurs critères

NORMES & STANDARDS

NIST | CMSS

Common Misuse Scoring System

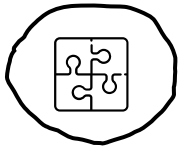
Score entre 0 et 25, 25 étant le plus critique

Point de mesure	Valeur
Access complexity	Basic
Authentication	User
Confidentiality impact	Some
Integrity impact	Some
Exploit (Zero-day impact)	Proof
Target distribution	All
Score CMSS : 15	

- 0 à 4 : Non critique
- 5 à 9 : Peu critique
- 10 à 14 : Modérément critique
- 15 à 19 : Critique
- 20 à 25 : Très critique

AC:Basic/A:User/CI:Some/II:Some/E:Proof/TD:All

SECURITY UPDATES CYCLE



CONTRIB

Mise à jour de sécurité tous les mercredis

CORE

Le troisième mercredi du mois

02 - CONFIGURATIONS

Des bonnes pratiques très efficaces
et simples à mettre en oeuvre

FICHIERS & CONFIGS

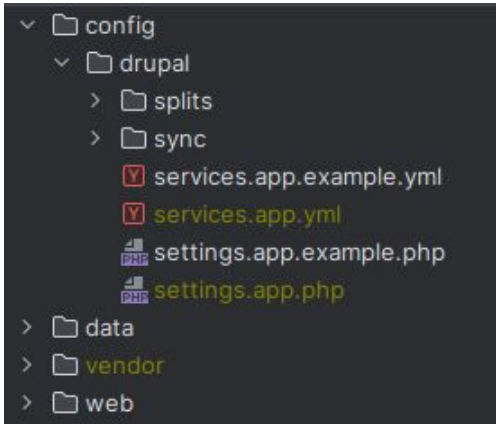
https://gitlab.com/nicoloye/drupal-security-101

FICHIERS .HTACCESS

Vigilance en dehors d'Apache
Vigilance sur les hébergeurs / PaaS

CONFIGURATIONS & SERVICES EN DEHORS DE LA RACINE (DOCROOT)

```php
if (file_exists($app_root . '/../config/drupal/settings.app.php')) {
 include($app_root . '/../config/drupal/settings.app.php');
}
```



Utiliser **drupal-paranoia**

CONFIGURATIONS ET SERVICES ADAPTÉS CHARGÉS EN PROD

Pas de settings.local.php
Pas de services.local.yml

LA CONFIGURATION DE PROD EST CELLE PAR DÉFAUT

Config Split pour les surcharges

NE PAS EXPORTER / VERSIONNER LES VALEURS SENSIBLES

Stocker des placeholders
Surcharger les valeurs depuis le fichier de settings

STOCKER LES VALEURS SENSIBLES DE FAÇON SÉCURISÉE

Usage du module **Dotenv**
Ne pas versionner le .env

Utiliser des secrets dans la CI

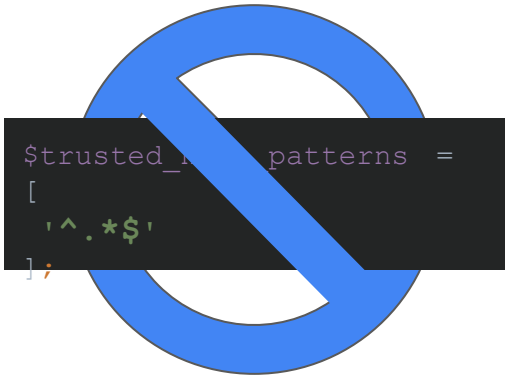
GitLab

STOCKER LES VALEURS SENSIBLES DE FAÇON SÉCURISÉE

Usage du module **Key**

Usage du module **Vault**

TRUSTED HOSTS PATTERNS RESTRICTIFS




```
$trusted_host_patterns = [
 '^www\.exemple\.fr$',
 '^www\.exemple\.com$',
];
```

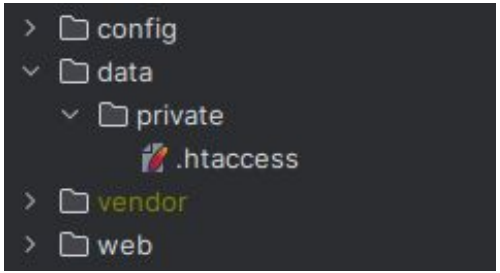

DOSSIERS
https://gitlab.com/nicoloye/drupal-security-101

GESTION DES MÉDIAS

Des fichiers non publics doivent être stockés
dans le système de stockage privé :

/admin/config/media/file-system

MÉDIAS PRIVÉS
EN DEHORS DE LA RACINE (DOCROOT)



https://gitlab.com/nicoloye/drupal-security-101

DOSSIER TEMPORAIRE
EN DEHORS DU DOCROOT

https://gitlab.com/nicoloye/drupal-security-101

AUCUN DUMP VERSIONNÉ

POLITIQUES DE SÉCURITÉ

https://gitlab.com/nicoloye/drupal-security-101

HTTPS PARTOUT !

Redirection vers HTTPS
Pas de mixed contents

EN-TÊTE HSTS

Usage de **seckit** (D8+)

CORRS

CORS

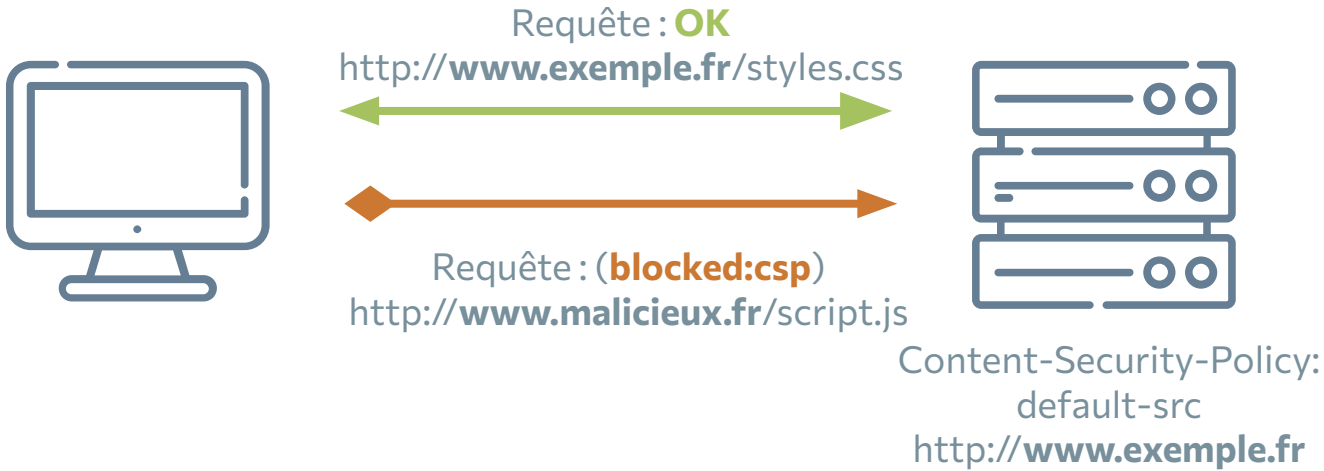
Configuration dans le fichier yml de services

CORS

```
cors.config:
 enabled: true
 # Specify allowed headers, like 'x-allowed-header'.
 allowedHeaders:
['x-csrf-token','authorization','content-type','accept','origin','x-requested-with']
 # Specify allowed request methods, specify ['*'] to allow all possible ones.
 allowedMethods: ['*']
 # Configure requests allowed from specific origins.
 allowedOrigins: ['http://www.mon-site-internet.fr']
 # Sets the Access-Control-Expose-Headers header.
 exposedHeaders: false
 # Sets the Access-Control-Max-Age header.
 maxAge: 1000
 # Sets the Access-Control-Allow-Credentials header.
 supportsCredentials: false
```

EN-TÊTE Content-Security-Policy

EN-TÊTE Content-Security-Policy



EN-TÊTE Content-Security-Policy

Usage de **CSP**

Usage de **Seckit**

EN-TÊTE X-Frame-Options

Usage de **Seckit**

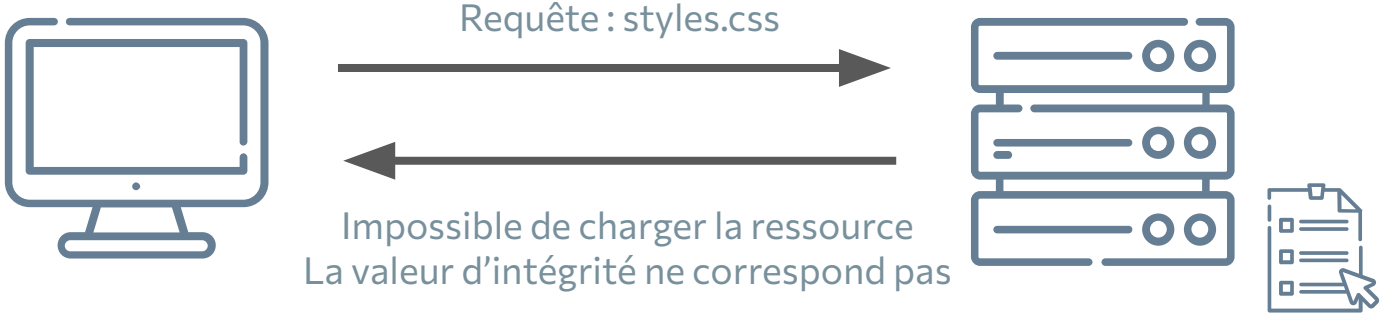
EN-TÊTE Referrer-Policy

Usage de **Seckit**

En cours d'intégration au core
(tagué pour 11.x-dev)

SUBRESOURCE INTEGRITY

```
<link rel="stylesheet" href="styles.css" integrity="sha384-foo1c6b" crossorigin="anonymous">
```



SUBRESOURCE INTEGRITY

```
ma-lib/styles.css: {
 type: "external",
 attributes: {
   integrity: "sha384-foo1c6b",
   crossorigin: "anonymous"
 }
}
```

SUBRESOURCE INTEGRITY

En cours d'intégration au core*
(tagué pour 11.x-dev)

* pour les éléments agrégés uniquement

EN-TÊTE X-Content-Type-Options

https://gitlab.com/nicoloye/drupal-security-101

EN-TÊTE X-XSS-Protection

Usage de **Seckit**

SUPPRIMER LES EN-TÊTES DRUPAL

Usage de **Remove_http_headers**

UTILISATEURS

https://gitlab.com/nicoloye/drupal-security-101

CHAQUE PERSONNE A UN COMPTE UTILISATEUR INDIVIDUEL

DÉSACTIVER LE COMPTE UTILISATEUR 1 (EN PRODUCTION)

POLITIQUE DE MOT DE PASSE & AUTRES SÉCURITÉS DE CONNEXION

Usage du module
Two-Factor Authentication

POLITIQUE DE MOT DE PASSE & AUTRES SÉCURITÉS DE CONNEXION

Usage du module **Password Policy**

Une politique cohérente
(un peu ancienne)

QUI PEUT CRÉER DES COMPTES UTILISATEURS ?

Vérifier que la politique de création de compte
est conforme aux besoins du métier :

/admin/config/people/accounts

VÉRIFIER LES PERMISSIONS DES RÔLES / UTILISATEURS

Mieux vaut pas assez que trop
Se méfier du rôle anonyme

INDEX SEARCH_API

Toujours activer le processeur “Content Access”

ERREURS & JOURNAUX

https://gitlab.com/nicoloye/drupal-security-101

MASQUER LES MESSAGES D'ERREUR

Masquer tous les messages :

/admin/config/development/logging

```
$config['system.logging']['error_level'] =
'hide';
```

CONFIGURER LA JOURNALISATION

Durée de rétention des données cohérente

CONFIGURER LA JOURNALISATION

Idéalement désactiver **DBlog**, privilégier
Syslog ou le module **Monolog**

CONFIGURER LA JOURNALISATION

Sentry

SENTRY

COEUR & MODULES

https://gitlab.com/nicoloye/drupal-security-101

MISES À JOUR DE SÉCURITÉ EFFECTUÉES & TABLEAU DE BORD À JOUR

Vérifier régulièrement le statut avec **Update Manager**

Il n'y a pas d'erreur ou de vigilance à traiter
(dans le contexte du projet) :

/admin/reports/status

MODULES / LIBS DE DEV À LEUR PLACE

Les dépendances de dev

```
"require-dev" : {
    "behat/mink" : "^1.8",
    "behat/mink-goutte-driver" : "^1.2",
    "drupal/devel" : "^4.1",
    "drupal/stage_file_proxy" : "^1.1",
    "phpunit/phpunit" : "^9.5"
},
```

```
"devDependencies"  : {
  "browser-sync" : "^2.18.12",
  "node-normalize-scss" : "^3.0.0",
  "stylelint" : "^7.12.0",
  "stylelint-config-standard" : "^16.0.0",
  "stylelint-order" : "^0.5.0"
}
```

PAS DE MODULE OU COMPOSANT INUTILISÉ

Les dépendances de dev ne sont pas appelées
Concerne tous les package managers
(composer, npm, etc)

https://gitlab.com/nicoloye/drupal-security-101

PAS DE COMPOSANTS OBSOLÈTES

Usage du plugin composer
Security Advisories

PAS DE COMPOSANTS OBSOLÈTES

Usage de **Dependabot**

Usage de **Renovate**

TOUJOURS À L'AISE ?




NOTRE PRÉSENTATION

03 - CODE

Des vigilances au quotidien

AUCUNE CONFIANCE
DES SAISIES UTILISATEUR

DE QUOI SE MÉFIER ?

- Soumissions de formulaire
- Arguments d'URL (query parameters)
- Chemins d'URL
- Enregistrements de bases de données
- Uploads d'utilisateurs
- E-mails entrants
- Cookies

- Headers HTTP
- Enregistrements DNS
- Enregistrements WHOIS
- Variables d'environnement
- Configurations en place
- Configurations par défaut
- … tout ce qui vient de l'extérieur !

SOLUTION COMPLÈTE

Beaucoup des vulnérabilités du TOP 10 sont couvertes par l'emploi de ces 3 règles simples et pour lesquelles Drupal met à disposition des APIs

3 RÈGLES DE BASE

01

VALIDER

Refuser le traitement dès la saisie

02

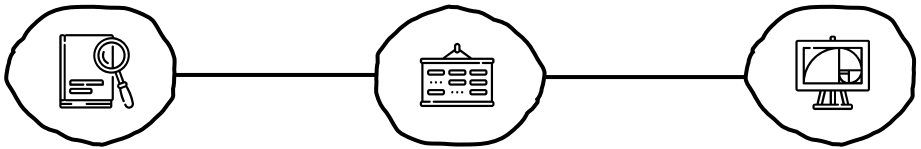
NETTOYER

Supprimer tout ce qui semble malicieux

03

ÉCHAPPER

Neutraliser les caractères sensibles



VALIDER

Valider les données saisies par l'utilisateur permet de refuser le traitement de la saisie dès le point d'entrée, tant que celle-ci ne correspond pas à un standard établi.

Validation d'un format e-mail :
✅ **example@example.com**
⛔ **Example-example**
⛔ **https//example.com**

Drupal met à disposition plusieurs services de validation permettant de s'assurer de la validité des données saisies par un utilisateur.

VALIDER

De plus, l'usage des composants Symfony dans le cœur permet également d'hériter d'un ensemble de validators très complets.

```
\Drupal::service('email.validator')->isValid($mail);
```

En dernier recours PHP met également à disposition des validators.

```
filter_var($mail, FILTER_VALIDATE_EMAIL);
```

NETTOYER & ÉCHAPPER

Nettoyer les données saisies par l'utilisateur permet de supprimer tout ce qui pourrait être considéré comme malicieux dans la saisie avant usage dans la logique applicative. Le nettoyage peut être une suppression des tags HTML, de caractères spécifiques, etc.

⛔ Entrées malicieuses	✅ Entrées nettoyées
How to <script>alert('xss');</script>	How to
How to <script>alert('xss');</script>	How to alert('xss');
my-song-***.mp3	my-song-___.mp3
class1>class2	class1_class2

NETTOYER & ÉCHAPPER

Échapper les données permet de neutraliser les caractères sensibles d'une chaîne de caractères potentiellement malicieuse en limitant l'apparence finale de la chaîne.

⛔ **How to <script>alert('xss');</script>**
✅ **How to <script>alert(\'xss\');</script>**

NETTOYER & ÉCHAPPER

Drupal met à disposition des méthodes de nettoyage et d'échappement qui permettent de réaliser des traitements sur des chaînes de caractères

```php
t('Comments to @type posts.', ['@type' => $string]);
\Drupal::translation()->formatPlural($string, '1 comment', '@count comments');
\Html::escape($string);
\Xss::filter($string);
\Xss::filterAdmin($string);
\UrlHelper::stripDangerousProtocols($uri);
\UrlHelper::filterBadProtocol($string);
```

NETTOYER & ÉCHAPPER

De la même façon, il est possible d'échapper des chaînes dans les requêtes SQL grâce aux tableaux de placeholders

```
$query = $connection->query("SELECT * FROM {posts}
    WHERE title = :title", [':title' => $query]);
```

▼

```
SELECT * FROM posts WHERE title = 'Best \'; DROP TABLE posts; --'
```

NETTOYER & ÉCHAPPER

Comme pour les validations, il est possible d'utiliser directement des sanitizers PHP

```php
filter_var('###foo@bar.com', FILTER_SANITIZE_EMAIL);
```

Il existe également différentes méthodes d'échappement.

```php
filter_var('test <script>alert("xss");</script>', FILTER_SANITIZE_FULL_SPECIAL_CHARS);
htmlspecialchars('test <script>alert("xss");</script>', ENT_QUOTES, 'UTF-8');
```

UTILISER LES API FOURNIES

- Consistance dans le code
- Sécurité
- Mise en cache
- Mises à jour communautaires

PERMISSIONS: REFUS PAR DÉFAUT

- Ne donner que les permissions **nécessaires**
- Penser “defensive programming”

```php
if (!\Drupal::currentUser()->hasPermission('access secret data')) {
    return;
}
```

ATTENTION AUX CACHES !

Utiliser la Cache API, avec les bons tags et contextes

Par exemple : un bloc qui change en fonction de l'utilisateur connecté ne doit pas garder en cache les informations de la personne précédente !

COOKIES CUSTOM

Vérifier les attributs **Secure** et **HttpOnly** des cookies déposés

```
Set-Cookie: id=dConEur; Expires=Tue, 17 Oct 2023 08:30:00 GMT; Secure; HttpOnly
```

04 - RESOURCES

Des aides à la mise en oeuvre

LOGICIELS

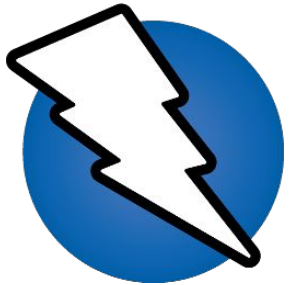
https://gitlab.com/nicoloye/drupal-security-101

SAST

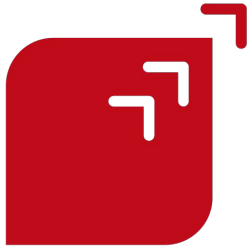
GitLab

sonar qube

DAST

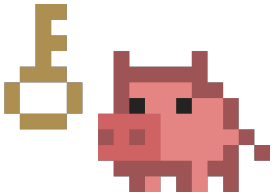


OWASP ZAP



OWASP Benchmark
(fork de Fluidattacks)

Scan de valeurs sensibles



GitHub

Git-secret

Trufflehog

Autres solutions

SAST / DAST

Liste d'outils OWASP

Observatory

moz://a


DOCUMENTATION
https://gitlab.com/nicoloye/drupal-security-101


moz://a

LINKS

Cheat sheets officiels de l'OWASP

- https://cheatsheetseries.owasp.org

Série de cours en ligne avec exercices pratiques

- https://portswigger.net/web-security

Bulletins d'alertes de sécurité

- https://www.drupal.org/security
- https://cve.mitre.org
- https://cwe.mitre.org/
- https://nvd.nist.gov/
- https://www.cert.ssi.gouv.fr/

LINKS

Méthodologie OWASP Software Assurance Maturity Model

- https://owaspsamm.org

Guide de l'identité numérique

- https://pages.nist.gov/800-63-3/sp800-63b.html#sec5

Guide de gestion d'incident

- https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-61r2.pdf

Passwords Seclists

- https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/10-million-password-list-top-1000000.txt

LINKS

Outils de contrôle des dépendances

- https://owasp.org/www-project-dependency-check/
- https://owasp.org/www-project-dependency-track/
- https://github.com/retirejs/retire.js/

Référentiel Opquast - Sécurité

- https://checklists.opquast.com/en/web-quality-assurance/detail/

your high-digital place Open

AXESS
Digital & Business Experts

SMILE

Talan

Merci

pour votre écoute !

DRUPALCAMP ROAZHON
Rennes - 28 au 30 mars 2024

https://gitlab.com/nicoloye/drupal-security-101