Procédure Mise à jour de sécurité Drupal 8

|  |  |  |
| --- | --- | --- |
| N° Version  | Auteur  | Modification |
| 1.0 | Morad BELLOT | Version initiale |

# 1. Objet

Cette procédure définit les dispositions à prendre pour maintenir les sites web développés avec Drupal 8 :

* Mise à jour de sécurité
* Test de fonctionnement et non régression
* Test d’intrusion

# 2. Pré-requis

Avant de démarrer la mise à jour de sécurité il faut disposer des éléments suivants :

|  |  |
| --- | --- |
|  | Accès cpanel serveur de production |
|  | Accès cpanel serveur de préproduction  |
|  | Accès git avec droit de commit sur la master |
|  | Accès admin BO Drupal |
|  | Environnement de développement installé :* Stack PHP/MYSQL avec les bonnes version
* Capistrano
* Drush
 |
|  | Contact client : email et N° de téléphone optionnel  |

#

# 3. Processus de MAJ Sécurité

[](https://www.draw.io/#G1Jd2F4-4UUlmJnK7s7CWfvDXAasnKMDOF)

# 4. Préparation des environnements

1. Récupérer la base de données production via le BO (Backup and migrate) à partir de la production
2. Faire un git clone du projet en local :
	1. **git clone « url de dépôt »**
	2. **git checkout master**
	3. **git pull**
3. Créer un branche maj\_secu\_ « jjmmaa » à partir de master
	1. **git checkout -b maj\_secu\_ « jjmmaa »**
4. Configurer les setting.local.php
5. Restaurer la base de données en local

# 5. Application des MAJ de sécurité en local

1. Rechercher les MAJ de sécurité : composer outdated drupal/\* (Maj en général) ou bien drush ups (securité)
2. Vider le cache : **drush cr**
3. Mettre à jour le core DRUPAL
	1. **composer require drupal/core:version --update-with-dependencies**
	2. **drush updb –y**
	3. **drush cr**
4. Annuler/compléter les modification de ces 2 fichiers :
	1. .htaccess : si modification reprendre l’ancienne version (redirection, performance ..) et la compléter avec les ajouts de la MAJ Drupal.
	2. .gitignore : si modification reprendre l’ancienne version (redirection, performance ..) et la compléter avec les ajouts de la MAJ Drupal.
5. Faire un test pour valider la MAJ du core
6. Commiter la MAJ du core :
	1. **git commit –m "core securtiy update to N° de la nouvelle version “**
7. Vider le cache : **drush cr**
8. Mettre à jour les modules contrib
	1. **composer require ‘drupal/module :^version’ –-update-with-dependecies**
	2. **drush updb –y**
9. Faire un test pour valider la MAJ des modules
10. Commiter la MAJ des modules
	1. **Commit –m " modules securtiy update to N° de la nouvelle version “**
11. Push le code : **git push**

# 6. Déploiement en preprod

1. Restaurer une copie de la base de données prod sur la preprod
2. Déploiement avec capistrano
	* Fichier : capistrano/config/preprod.rb
	* Changer set branch avec le nom de la branche crée pour la MAJ de sécurité
	* Ne pas comiter le changement de ce fichier
3. cd capistrano
4. cap preprod deploy
5. après déploiement il faut exécuter **drush updb –y** sur la preprod si nécessaire

# 7. Test de non régression

Faire un test visuel en naviguant sur les différentes pages du site et vérifier que les pages continuent à s’afficher correctement.

Vérifier également ces fonctionnalités :

* Connexion utilisateur
* Formulaire de soumission
* Filtre des pages listing
* Affichage traduction
* Contribution sur le backoffice

# 8. Déploiement en production

Avant de procéder à la mise à jour de la production, il faut faire une sauvegarde de la base de données pour pouvoir faire un rollback au cas d’échec du déploiement.

1. Merger maj\_secu\_ « jjmmaa » dans master en local et pusher le code
	1. git checkout master
	2. git pull
	3. git merge maj\_secu\_ « jjmmaa »
	4. git push
2. Sauvegarder la base de données production avec backup and migrate
3. Installer la nouvelle version en production : cap production deploy
4. Après déploiement il faut exécuter **drush updb –y** s’il y en a

#

# 9. Annexe

## 9.2. Templates des emails

### 9.2.1. Planning de MAJ de sécurité

Un email informatif doit être envoyé au client 1 semaine avant le jour de la MAJ de sécurité.

|  |
| --- |
| Bonjour,Nous vous informons que l’équipe technique VOID va procéder à une mise à jour de sécurité de votre site web le xx/xx/xxxx.L’objectif de cette intervention est de maintenir votre site à jour avec les derniers correctifs pour corriger d’éventuelle faille de sécurité.Nous vous tenons informés une fois la procédure est terminée avec un rapport des actions menées.Equipe Support & Maintenance |

### 9.2.2. Absence de mise à jour de sécurité

Un email informatif doit être envoyé au client le jour de la MAJ de sécurité

|  |
| --- |
| Bonjour,Nous vous informons que l’équipe technique VOID a procédé à un contrôle de version de votre site web le xx/xxx/xxx. Votre version est à jour et il n y a aucune mise à jour de sécurité à déployer.En effet la version de votre CMS est 8.xx qui correspond à la dernière mise à jour de sécurité.Nous vous tiendrons informés une fois une nouvelle mise à jour existe.Equipe Support & Maintenance |

### 9.2.3. Rapport fin mise à jour de sécurité

 Un rapport de la MAJ de sécurité à envoyer au client.

|  |
| --- |
| Bonjour,Nous vous informons que l’équipe technique VOID a procédé à une mise à jour de sécurité de votre site web le xx/xx/xxxx à hhHMM. Il s’agit d’une mise à jour technique des modules suivants pour renforcer la sécurité de votre site :- Core CMS Drupal nouvelle N° de version- Module xxx nouvelle N° de version- Module yyy nouvelle N° de version Après déploiement de ces mises à jour, nous avons lancé un scan de sécurité Acunetix pour vérifier qu’il n y pas de vulnérabilités sur votre site web. Notre équipe de recette a procédé également à un test de non régression pour confirmer le bon fonctionnement de votre site web, à savoir :- Affichage des pages- Connexion utilisateur- Formulaire de soumission- Contribution back office- Gestion des médias Pour toute remarque ou question, n’hésitez pas à prendre contact avec nos équipes via notre plateforme de support http://redmine3.void.fr afin d’en assurer un meilleur suivi. Equipe Support & Maintenance |

#

#