SECURE SPHERE by EPITA

SECURE DEV

Recette, déploiement et intégration

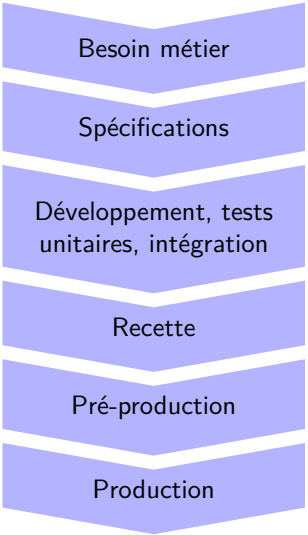
Quentin GROSYEUX

2023 - BNP

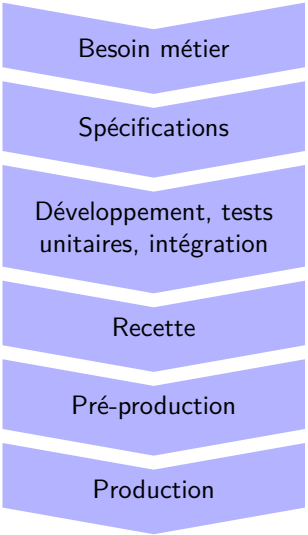
Table des matières

Table des matières

Processus de développement

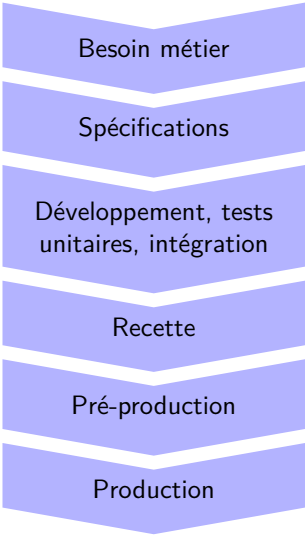


Processus de développement



Analyse de risque, concepts et principes de sécurité

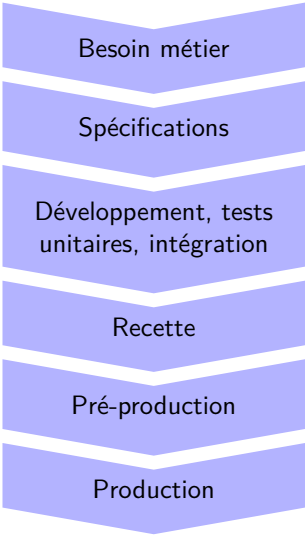
Processus de développement



Analyse de risque, concepts et principes de sécurité

Développement sécurisé, tests manuels et unitaires sécurité, analyse statique de code

Processus de développement

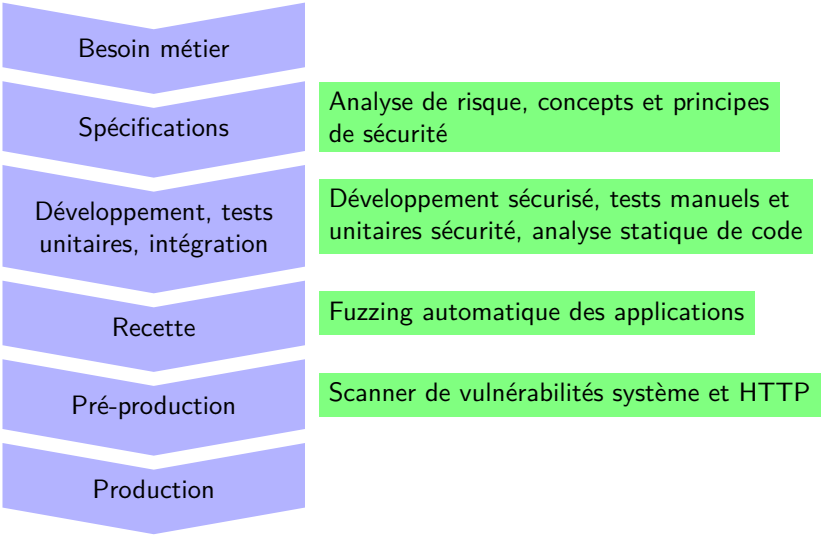


Analyse de risque, concepts et principes de sécurité

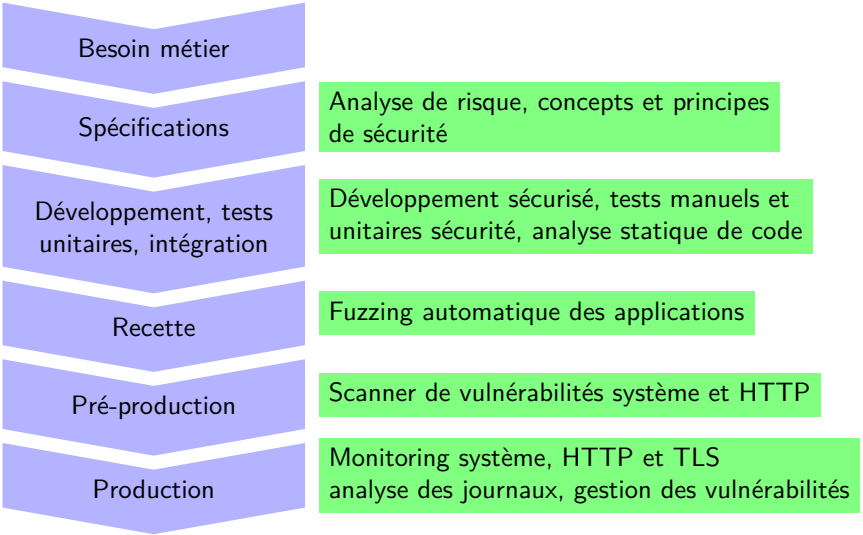
Développement sécurisé, tests manuels et unitaires sécurité, analyse statique de code

Fuzzing automatique des applications

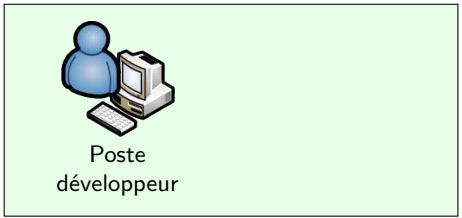
Processus de développement



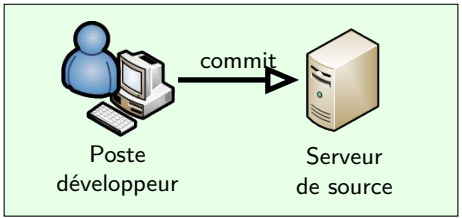
Processus de développement



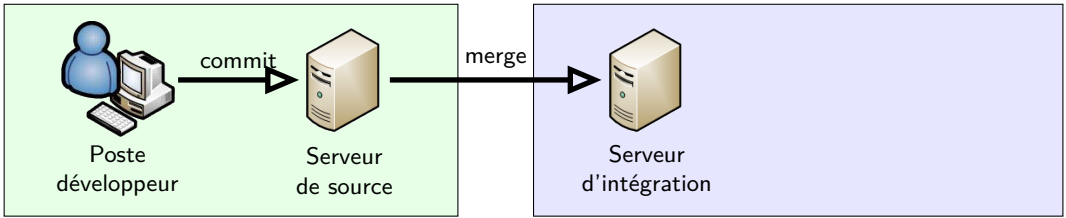
Software Development Life Cycle



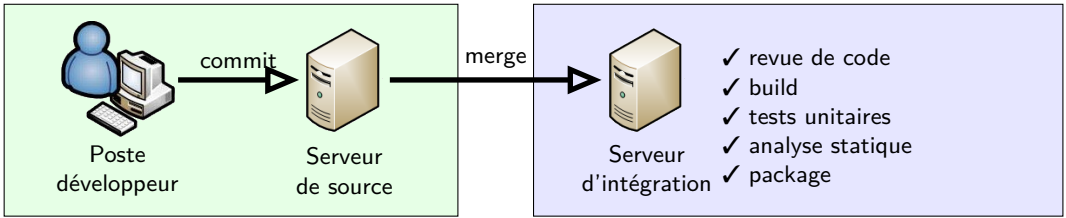
Software Development Life Cycle



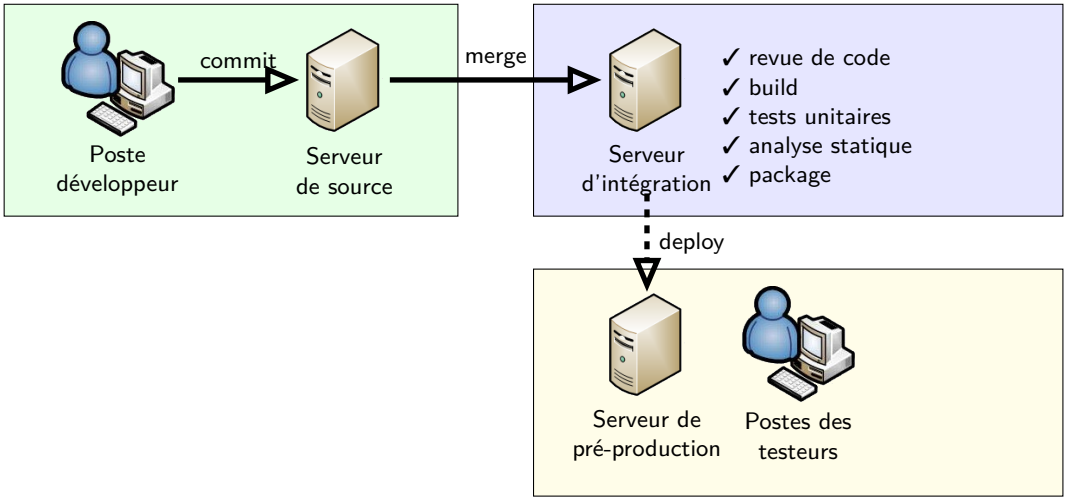
Software Development Life Cycle



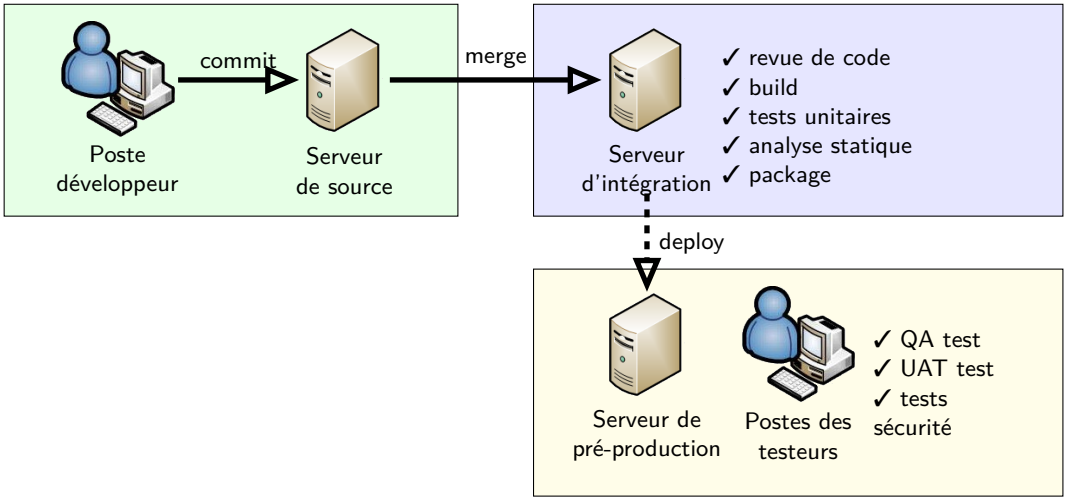
Software Development Life Cycle



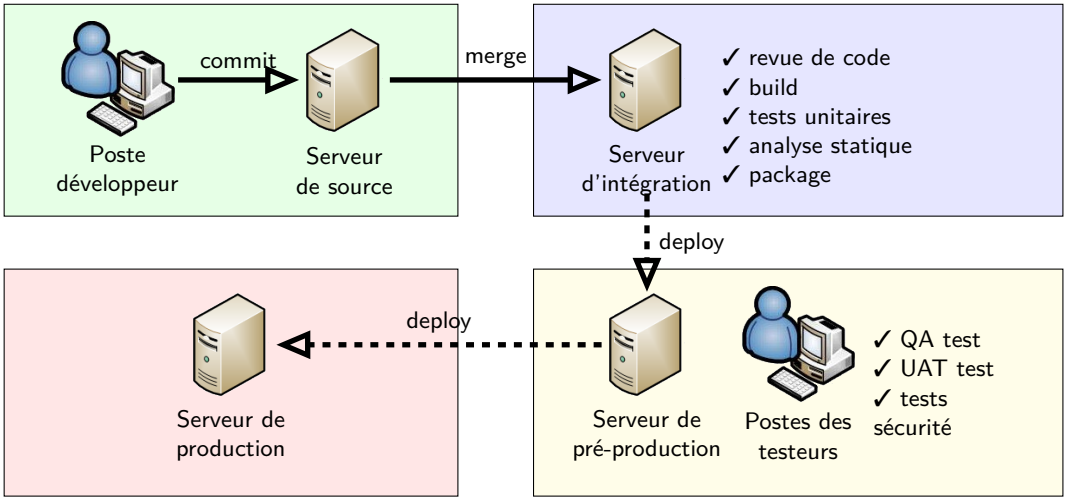
Software Development Life Cycle



Software Development Life Cycle



Software Development Life Cycle



Tests manuels

Phase de recette

- déterminer la surface d'attaque
- réaliser des tests techniques (tests d'intrusion) :
  - changer la taille des entrées, insérer des caractères spéciaux :
  - changer le type des entrées (valeur négatives, chaînes, etc.)
  - modifier les arguments (URL, ligne de commande, fichiers, etc.)
  - tester tous les cas d'erreur et de problèmes de configuration
- auditer les configurations des composants (socle système, services, applicatifs, etc.)
- faire éventuellement effectuer un audit (revue) de code source

Caractères spéciaux

Caractères à tester ?

Caractères spéciaux

Caractères à tester ?

`& > < ’ " ( ) | ; $ %00 * / \ - %x %n \t \n \r`

Dans les tests unitaires (`jest`, `pytest`, etc.) et dans les tests fonctionnels

Caractères spéciaux

Exercice

Utilisez l'extension Chrome `Bug Magnet` pour découvrir les problèmes dans le champs de recherche du site de SuperBouchons

OWASP ASVS

OWASP Application Security Verification Standard (v4) :

- 279 points à vérifier pour évaluer la sécurité d'une application Web

OWASP ASVS

OWASP Application Security Verification Standard (v4) :

- 279 points à vérifier pour évaluer la sécurité d'une application Web

Exercice

Récupérer le document sur le site de l'OWASP et le parcourir

Outils

CI/CD

Continuous Integration / Continuous Delivery :

- CI :
  - ensemble de pratiques (commits unitaires, *push* fréquents sur le serveur de sources, etc.)
  - techniques consistentes et automatisées de construction, de packaging et de test des applications
- CD :
  - méthode automatique pour transmettre les changements du code dans les différents environnements logiciels, jusqu'à la production (déploiement du serveur ou mise à disposition des livrables)

CI/CD


CONTINUOUS INTEGRATION
CONTINUOUS DELIVERY
CONTINUOUS DEPLOYMENT
BUILD
TEST
MERGE
AUTOMATICALLY RELEASE TO REPOSITORY
AUTOMATICALLY DEPLOY TO PRODUCTION

Exemple de pipeline CD

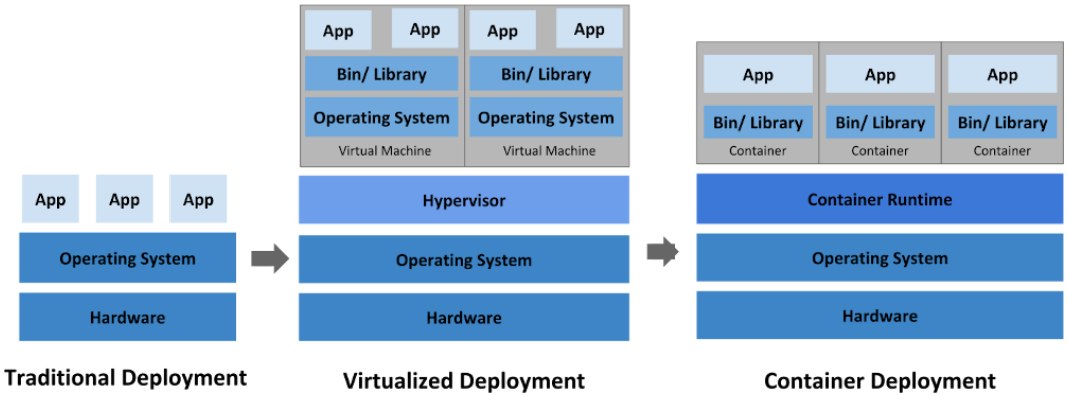
1. récupération du code depuis le serveur de source, import des dépendances et construction de la solution
2. exécution de commandes pour configurer l'environnement de test
3. copie de la solution dans l'environnement de test
4. création des variables d'environnement nécessaires pour l'exécution
5. déploiement des composants nécessaires pour l'exécution (services Web, API, bases de données, etc.)
6. démarrage des services et initialisation (données minimales)
7. exécution de la suite de test et récupération des résultats
8. restauration de l'environnement
9. fourniture des résultats et des alertes sur l'état de la solution

Conteneurs

- souvent liés à des pipelines CI/CD (dans le cloud ou pas) ou pour faciliter le déploiement :
  - environnement vierge et dédié à une solution : aucun effet de bord entre les différentes exécutions
  - destruction immédiate : pas besoin de restauration de l'environnement
- moyen de fournir une solution clé en main dans un format standard et portable (*infrastructure as code*), déploiements homogènes et maîtrisés chez les clients
- isolation entre les services (réduction de la conséquence d'une vulnérabilité dans un composant)
- brique de gestion de la montée en charge (*horizontal scaling*) avec des instances strictement identiques

Docker, Podman, rkt, etc.
Docker compose, Kubernetes, Amazon Elastic Container Service, etc.

Conteneurs



Source : documentation Kubernetes

Images docker

10 février 2020

Misconfigured Docker Registries Expose Thousands of Repositories

Thousands of code repositories exposed in over one hundred Docker registries accessible from the Internet without authentication, 92 allowing the push operation, and 7 the delete operation

Images docker

Exercice

Lancer dans la VM une instance de l'image docker de mongodb

Images docker

Exercice

Lancer dans la VM une instance de l'image docker de mongodb

`https://hub.docker.com/_/mongo` (`docker search mongodb`)

Images docker

Exercice

Lancer dans la VM une instance de l'image docker de mongodb

`https://hub.docker.com/_/mongo` (`docker search mongodb`)

```
docker run --name mongo mongo:bionic
```

Conteneurs

Solutions

- ✓ dédier un hyperviseur ou serveur de conteneurs à chaque environnement (développement, test, production, etc.)
- ✓ créer des templates de conteneurs/images de VM sécurisés
- ✓ n'utiliser que des images Docker officielles ou créer ses propres images (`FROM scratch` ou avec un système de fichiers de bootstrap de la distribution)
- ✓ ne pas exécuter d'applications en root dans les conteneurs (ou utiliser les *user namespaces*), ne pas créer des conteneurs priviligiés (*capabilities* ou accès direct aux périphériques)
- ✓ ne partager que les répertoires strictement nécessaires entre l'hôte et les VM/conteneurs

Gestion des secrets CI/CD

Protection des secrets liés au déploiement continu (mots de passe des services, tokens d'accès à la plate-forme, clés SSH, etc.), du moins au plus sécurisé :

- dans des variables d'environment
- dans des fichiers de configuration
- dans un outil de gestion des secrets (HashiCorp Vault, etc.)

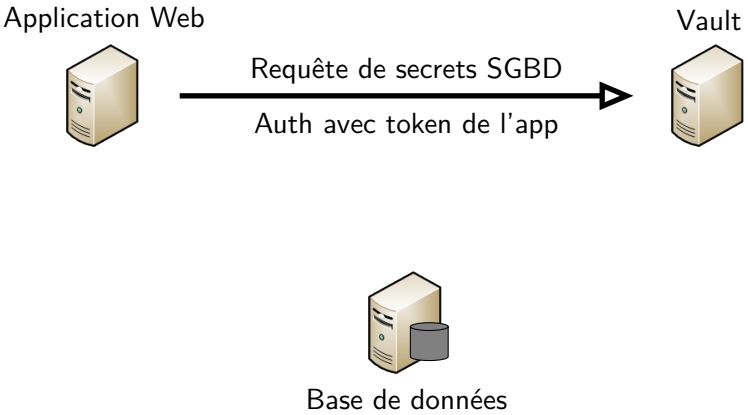
Secrets as a Service

Application Web

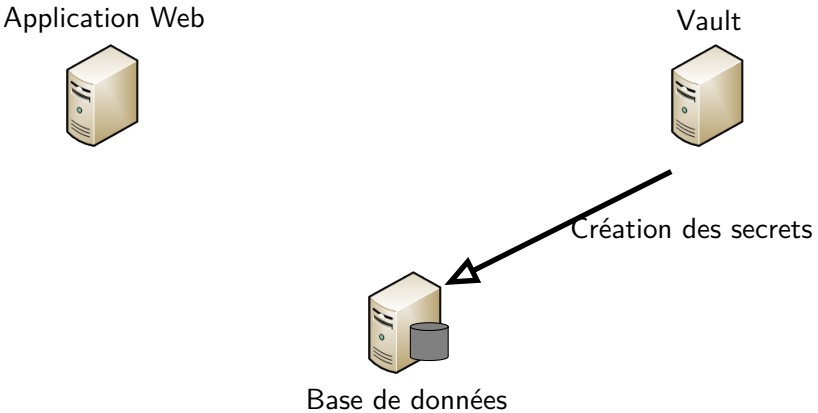
Vault

Base de données

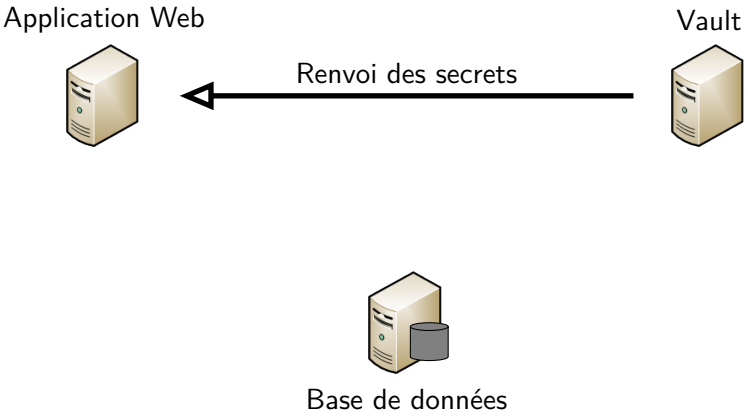
Secrets as a Service



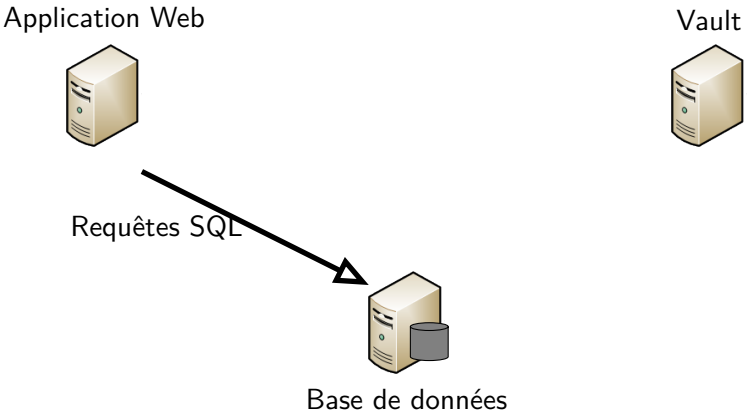
Secrets as a Service



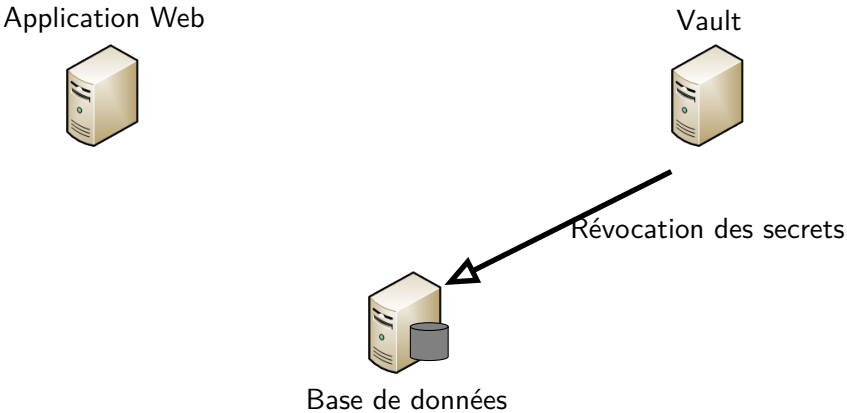
Secrets as a Service



Secrets as a Service



Secrets as a Service



Analyse automatisée

- outils servant à automatiser les tests
- pas d'exhaustivité, pas de garantie d'absence de vulnérabilité
- ne remplace pas un vrai audit de sécurité/test d'intrusion

Analyse des en-têtes HTTP

Exercice

Utilisez `https://observatory.mozilla.org` pour analyser les améliorations possibles du site publique de votre société

Fuzzing

Découverte automatisée des vulnérabilités de type injection (SQLi, XSS, exécution de commandes, etc.) et de configuration de sites Web

Exercice

Utilisez `wapiti` et ZAProxy pour découvrir les failles de la VM.

```
$ wapiti -u http://127.0.0.1:8080/ -f txt
$ zap.sh -quickurl \
    http://127.0.0.1:8080/ -quickout report.xml -cmd
```

Fuzzing API

Découverte automatisée des vulnérabilités de type injection (SQLi, XSS, exécution de commandes, etc.) et de configuration des API HTTP

Exercice

Utilisez `wfuzz` et `artilleryio/artillery-plugin-fuzzer` sur l'API de SuperBouchons

Analyse automatisée d'applications Web

- `https://github.com/mozilla/observatory-cli` : en-têtes HTTP
- `testssl` : configuration TLS et certificat HTTPS
- `wfuzz` : brute-force de répertoires
- `burp` et `ZAP` : relai Web
- `w3af`, `wapiti`, `ZAP` : fuzzing d'applications automatiques
- `wfuzz`, `artillery-plugin-fuzzer` : fuzzing d'applications à configurer

Analyse statique de code

- analyse du code source sans exécution
- simple `grep` ou compréhension du flot d'exécution
- beaucoup de faux positifs (à trier) et de faux négatifs
- disparité de support entre langages
- Wikipedia : *List of tools for static code analysis*

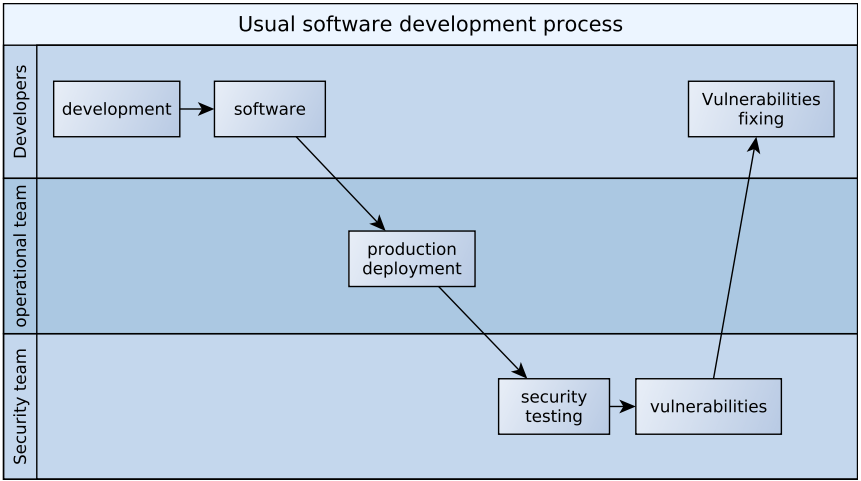
Analyse statique de code Node.js

Outils gratuits :

- *linter* : `jslint`, `eslint`
- sécurité : ajinabraham/NodeJsScan (paquet python nodejsscan...), insidersec/Insider (en Go...)

DevOps

Usual Suspects



Concepts

- évolution du concept "agile" (2009) : fourniture rapide de logiciels de qualité
- mêmes personnes qui développent, testent le fonctionnement et déploient les logiciels (plusieurs casquettes et compétences)
- repose sur l'automatisation du déploiement : utilisation massive de mécanismes d'installation rapide de systèmes (Ansible, Puppet, Docker, etc.)
- la sécurité était un peu laissée de côté

DevSecOps

- évolution du concept DevOps :-)
- mise en œuvre du concept de *secure by design* : la sécurité est intégrée à chaque étape de vie d'un logiciel
- mêmes personnes qui développement, sécurisent le code, testent le fonctionnement, testent la sécurité, déploient et surveillent la sécurité des logiciels
- la sécurité est réalisée par les développeurs, pas par une équipe dédiée et séparée

DevSecOps

- nécessite une formation à la sécurité pour toute l'équipe DevOps
- présence d'un "security champion" par équipe, avec une certaine expertise en sécurité (architecture logicielle, fonctions de sécurité, etc.)
- ajout d'outils de sécurité dans la chaîne DevOps : tests unitaires de sécurité, analyse statique de code source, surveillance de la configuration, scans de vulnérabilité, etc.

CI/CS/CD ;-)

Impact des pratiques DevOps

- les développeurs sont désormais responsables de certains systèmes en production et de leur sécurité (déploiement, durcissement, supervision, etc.)
- des comptes privilégiés sur les systèmes sont nécessaires pour le déploiement : attention à leur usage !
- une supervision de ces comptes et une bonne gestion (renouvellent des mots de passe, suppression, etc.) sont nécessaires
- un système centralisé et sécurisé de gestion des secrets est souvent utilisé (ex : Vault)

Table des matières

Analyse de risque sur le développement

Exercice

Application de l'analyse de risque sur votre SI pour déterminer les mesures de sécurité à mettre en place.

Biens essentiels et biens supports

Quels sont les éléments importants à protéger ?

Exemple 1

05/01/2021 : SolarWinds : Les attaquants ont eu accès au code source de Microsoft.

Microsoft a déclaré jeudi que les auteurs de la cyberattaque ayant touché SolarWinds ont réussi à accéder à son réseau interne, où ils ont eu accès à quelques comptes internes, qu'ils ont utilisés pour accéder aux dépôts de code source de Microsoft.

Le fabricant de Windows rassure : les cyberattaquants n'ont pas pu apporter de modification aux dépôts auxquels ils ont eu accès car les comptes compromis n'avaient pas la permission de modifier le code, seulement de le visualiser.

Exemple 1

05/01/2021 : SolarWinds : Les attaquants ont eu accès au code source de Microsoft.

Microsoft a déclaré jeudi que les auteurs de la cyberattaque ayant touché SolarWinds ont réussi à accéder à son réseau interne, où ils ont eu accès à quelques comptes internes, qu'ils ont utilisés pour accéder aux dépôts de code source de Microsoft.

Le fabricant de Windows rassure : les cyberattaquants n'ont pas pu apporter de modification aux dépôts auxquels ils ont eu accès car les comptes compromis n'avaient pas la permission de modifier le code, seulement de le visualiser.

Accès aux postes/comptes des développeurs $\Rightarrow$ récupération de code source, voire piégeage pour attaquer les clients

Exemple 2

29/03/2021 : Le serveur Git du projet PHP piraté, l'attaquant a ajouté une backdoor dans le code source.

Deux commits ont été ajouté au dépot `php-src` sur le serveur `git.php.net`, usurpant l'identité du créateur de PHP et d'un autre développeur dans les messages de commit, sous un prétexte de "*fix typo*". Ils permettent une exécution de commande arbitraire si un serveur Web utilise cette version de PHP et qu'un client utilise un user-agent spécial.

Les commits ont été détectés en quelques heures, grâce au code review post-commit. PHP a décidé d'utiliser GitHub suite à cela.

Exemple 3

Rançongiciels sur des entreprises françaises :

- CMA-CGM (10/2020)
- M. Bricolage (10/2020)
- Siplec (pétrole E. Leclerc) (11/2020)
- Sopra Steria (11/2020) : impact financier entre 40 et 50 millions d'euros
- CHU de Rouen (11/2020)
- Hôpital de Dax (02/2021)
- Hôpital de Villefranche (02/2021)
- Hôpital d'Oloron-Sainte-Marie (03/2021)

Exemple 3

Rançongiciels sur des entreprises françaises :

- CMA-CGM (10/2020)
- M. Bricolage (10/2020)
- Siplec (pétrole E. Leclerc) (11/2020)
- Sopra Steria (11/2020) : impact financier entre 40 et 50 millions d'euros
- CHU de Rouen (11/2020)
- Hôpital de Dax (02/2021)
- Hôpital de Villefranche (02/2021)
- Hôpital d'Oloron-Sainte-Marie (03/2021)

Cryptolocker sur postes et serveurs $\Rightarrow$ perte de la dernière version des sources

Exemple 4

Compromission du site Internet de Linux Mint en février 2016. Les attaquants ont remplacé les liens pointant normalement vers le téléchargement d'une version légitime de la distribution Linux Mint par des liens vers une distribution contenant une porte dérobée. Linux Mint aurait rapidement supprimé les liens illégitimes de son site Internet, mais recommande aux victimes potentielles de supprimer les ISO téléchargées, de formater leurs disques durs et de changer tous leurs mots de passe potentiellement exposés.

Exemple 4

Compromission du site Internet de Linux Mint en février 2016. Les attaquants ont remplacé les liens pointant normalement vers le téléchargement d'une version légitime de la distribution Linux Mint par des liens vers une distribution contenant une porte dérobée. Linux Mint aurait rapidement supprimé les liens illégitimes de son site Internet, mais recommande aux victimes potentielles de supprimer les ISO téléchargées, de formater leurs disques durs et de changer tous leurs mots de passe potentiellement exposés.

Compromission du serveur de distribution $\Rightarrow$ modification des binaires et des codes source publiques

Exemple 5

- outil d'attaque Cobalt Strike signé par clé privée de NVIDIA (2022)
- code malveillant Duqu 2 (2015) signé par clé privée de Foxconn
- codes malveillants signés par Adobe (2012)
- plus de 500 certificats malveillants signés par l'autorité de certification DigiNotar (2011), utilisés notamment pour effectuer un *man-in-the-middle* pour les serveurs de Google, de Yahoo, de Tor, etc.
- 9 signatures de certificats effectuées par la clé privée de Comodo (2011)

Exemple 5

- outil d'attaque Cobalt Strike signé par clé privée de NVIDIA (2022)
- code malveillant Duqu 2 (2015) signé par clé privée de Foxconn
- codes malveillants signés par Adobe (2012)
- plus de 500 certificats malveillants signés par l'autorité de certification DigiNotar (2011), utilisés notamment pour effectuer un *man-in-the-middle* pour les serveurs de Google, de Yahoo, de Tor, etc.
- 9 signatures de certificats effectuées par la clé privée de Comodo (2011)

Compromission d'un serveur interne $\Rightarrow$ récupération de la clé privée de signature de binaire et utilisation pour signer du code malveillant

Exemple 6

Fin 2013, la société américaine Target annonce le vol de 110 millions de données personnelles de clients.
Les attaquants ont attaqué par *phishing* les employés du prestataire de HVAC (chauffage, ventilation et climatisation) 2 mois auparavant, puis ont rebondit sur un serveur interne du SI de Target à travers l'interconnexion liée à la facturation, gestion des contrats et des projets.

Exemple 6

Fin 2013, la société américaine Target annonce le vol de 110 millions de données personnelles de clients.
Les attaquants ont attaqué par *phishing* les employés du prestataire de HVAC (chauffage, ventilation et climatisation) 2 mois auparavant, puis ont rebondit sur un serveur interne du SI de Target à travers l'interconnexion liée à la facturation, gestion des contrats et des projets.

Compromission d'un prestataire externe $\Rightarrow$ utilisation des accès (privilégiés ou non) chez les clients

Mesures de protection

Dépend du contexte :

Solutions

- ✓ isolation réseau des postes des développeurs et des serveurs de développement
- ✓ gestion fine des droits d'accès aux dépôts de code source (en particulier les droits d'écriture)
- ✓ signature du code sur une machine hors ligne
- ✓ protection du serveur de distribution et restriction sur les accès des administrateurs (plages IP, authentification forte, etc.)
- ✓ vérification très fréquente de l'intégrité des données du serveur de distribution
- ✓ archivage hors ligne des binaires et de la source de chaque version

Mesures de protection

Solutions

- ✓ ne pas mettre de vraies données métier dans les environnements de développement et de pré-production (anonymisation)
- ✓ ne partager des mots de passe ou des secrets entre les environnements

Table des matières

SuperBouchons

Exercice

Quels sont les problèmes relatifs à la configuration ?

SuperBouchons

Exercice

Quels sont les problèmes relatifs à la configuration ?

- en-tête `X-Powered-By` ajoutée par Express
- Nom du cookie de session : `connect.sid`

Corriger l'application

Exercice

Correction (`sb.js`) :

- `app.disable("x-powered-by");` ou grâce à `helmet`
- `app.use(session(key: "PHPSESSIONID"));`

Helmet

Solutions

✓ utiliser `helmet` pour les applications Express : précise les en-têtes HTTP relatives à la sécurité (CSP, HSTS, `X-Powered-By`, XSS, *clickjacking*)

Protections réseau

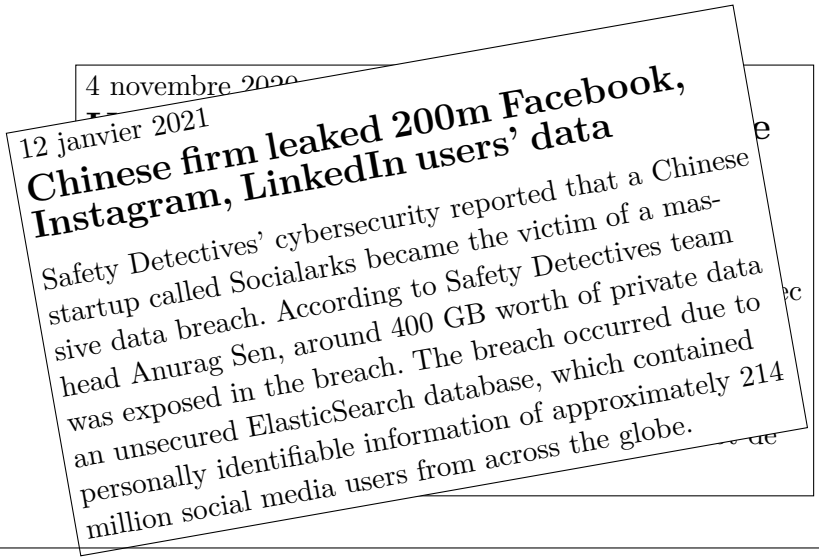
Interfaces d'administration

4 novembre 2020

Un problème de configuration expose les mots de passe de deux millions de cultivateurs de marijuana

Une communauté en ligne où les cultivateurs de marijuana peuvent bloguer sur leurs plantes et interagir avec d'autres cultivateurs, GrowDiaries, a subi une faille de sécurité en septembre de cette année. La violation de données s'est produite après que la société a laissé deux applications Kibana exposées sur internet sans mot de passe.

Interfaces d'administration



4 novembre 2020

12 janvier 2021

Chinese firm leaked 200m Facebook, Instagram, LinkedIn users' data

Safety Detectives' cybersecurity reported that a Chinese startup called Socialarks became the victim of a massive data breach. According to Safety Detectives team head Anurag Sen, around 400 GB worth of private data was exposed in the breach. The breach occurred due to an unsecured ElasticSearch database, which contained personally identifiable information of approximately 214 million social media users from across the globe.

Filtrage

Solutions

- ✓ ne laisser accessibles aux utilisateurs que les flux strictement nécessaires : pare-feu
- ✓ bloquer l'accès aux interfaces d'administration depuis les réseaux autres que ceux des administrateurs : pare-feu, relai-inverse filtrant (*reverse proxy*) ou configuration du service Web

NIDS

- système de détection d'intrusion réseau
- détecte des attaques et émet des alertes (ne bloque pas les flux malveillants) en analysant les flux réseau (TAP réseau, *port mirroring*)
- nécessite de reconstruire les flux applicatifs à partir des paquets réseau
- base de signatures d'attaque (motifs d'attaque) : ne détecte que ce qui est connu
- recherche d'anomalies : nécessite d'apprendre les flux légitimes

Snort

Exercice

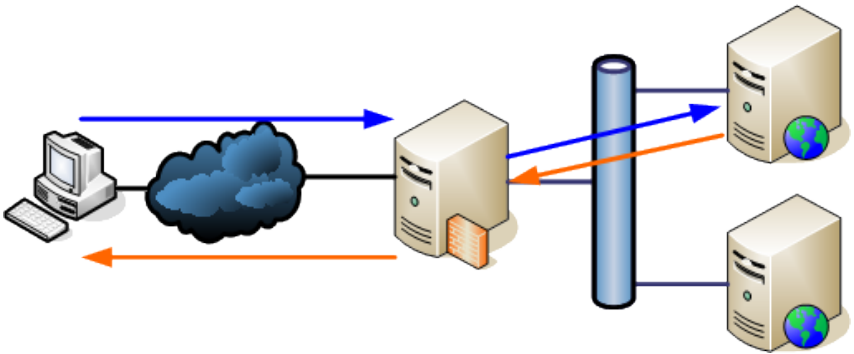
Analyser quelques règles de Snort :
`https://www.snort.org/downloads/community/community-rules.tar.gz`

Limites des NIDS

- beaucoup de faux positifs
- mise à jour continuelle des signatures
- ne peut détecter certaines catégories d'attaques (contrôle d'accès défaillant, etc.)
- techniques d'évasion au niveau réseau (défragmentation différente selon les OS)
- aveugle face au chiffrement des flux (TLS)
- charge importante pour l'analyse des protocoles dynamiques

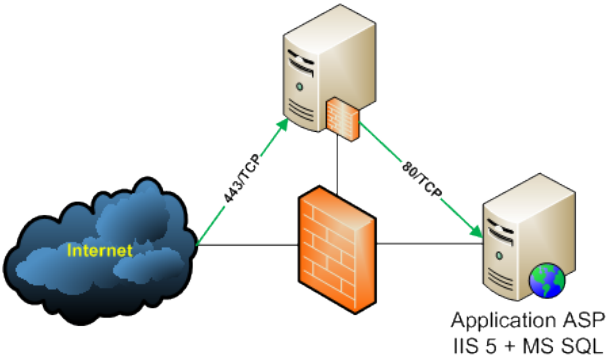
Positionnement des relais-inverses

- en entrée de réseau : relais-inverse, traitant les requêtes de clients non identifiés, à destination de ressources identifiées (ex : relais HTTP filtrant en entrée)



⇒ protection des serveurs

Relais inverse HTTP



- pare-feu autorisant 443/TCP vers le relais-inverse et 80/TCP du relais-inverse vers le serveur
- relais-inverse HTTPS :
  - tête de tunnel HTTPS
  - normalisation des URL, filtrage et journalisation (GET, POST)

WAF

Web application firewall :

- client $\leftrightarrow$ WAF (DNS) $\leftrightarrow$ serveur Web
- motifs (expressions rationnelles) standards de détection d'attaques (XSS, injection SQL, exécution de commande, erreurs de configuration du serveur, etc.)
- permet de réagir en urgence de manière centralisée en cas de problèmes (0day sur un framework ou application Web)
- limitations :
  - ne peut pas détecter toutes les classes d'attaques (CSRF, erreurs de conception, divulgations d'informations sensibles, gestion des droits, etc.)
  - les motifs ne sont pas exhaustifs pour chaque classe d'attaque
  - gestion des *websockets* et du *webRTC* (partage vidéo et son en peer-to-peer)
- défense en profondeur, ne remplace pas le développement sécurisé !

Apache/mod_proxy

Relais-inverse HTTP : apache et mod_proxy.

```
ProxyPass /appli1 http://192.168.0.1/appli
ProxyPass /appli2 http://192.168.0.2/
ProxyPassReverse /appli1 http://192.168.0.1/appli
ProxyPassReverse /appli2 http://192.168.0.2/
```

Filtrage et journalisation : mod_security.

Apache/mod_security

```
SecRuleEngine On
[...]
SecAuditEngine On
SecAuditLog /var/log/apache/audit_log

SecFilterScanPost On
SecFilterCheckURLEncoding On
SecFilterCheckUnicodeEncoding Off
SecFilterSelective THE_REQUEST \
                      "!^[\x0a\x0d\x20-\x7f\xa0-\xff]+$"

SecFilterDefaultAction "deny,log,status:500"
ErrorDocument 500 /error.html
SecFilterSelective HTTP_Transfer-Encoding "!^$"

SecFilter "\.\./"
```

Apache mod_security

Exercice

Analyser quelques règles de mod_security :
`https://github.com/coreruleset/coreruleset/`

Service Web

Durcissement générique (Apache)

Solutions

- ✓ mettre à jour
- ✓ configurer correctement TLS (*Mozilla SSL Configuration Generator*)
- ✓ désactiver les modules inutiles (`userdir`, `suexec`, `cgi`/`cgid`, `include`, `autoindex`)
- ✓ désactiver le support de HTTP 1.0
- ✓ limiter le nombre de requêtes par fils (`MaxRequestsPerChild`) et le timeout (`Timeout`)
- ✓ désactiver la méthode TRACE (`TraceEnable`)
- ✓ désactiver le suivi des liens symboliques (`Options -FollowSymLinks`)
- ✓ masquer les bannières (`ServerTokens` et `ServerSignature`) et l'en-tête ETag (`FileETag`)

Durcissement générique (Apache)

Solutions

- ✓ activer `mod_security`
- ✓ supprimer les répertoires par défaut dans la racine et les règles de réécriture par défaut (`cgi-bin`, etc.)
- ✓ activer les en-têtes de protection des sites : `X-XSS-Protection`, `X-Frame-Options`, etc.

SGBD

Introduction

SGBD = Système de Gestion de Base de Données
Une cible de choix pour les attaquants :

Introduction

SGBD = Système de Gestion de Base de Données
Une cible de choix pour les attaquants :

- souvent mal configurés (pas de durcissement)
- administration complexe nécessitant des compétences spécifiques
- difficulté à redémarrer le service, donc peu à jour
- contiennent des données métier intéressantes

Brute-Force

Video : brute-force en ligne d'un service MariaDB et extraction des empreintes des mots de passe

Durcissement générique

Solutions

- ✓ dédier un serveur/conteneur
- ✓ maintenir à jour !
- ✓ utiliser un compte non privilégié sur l'hôte
- ✓ n'autoriser les connexions entrantes que depuis des origines maîtrisées (administrateurs DBA)
- ✓ chiffrer les communications si supporté
- ✓ désactiver toutes les fonctionnalités et services inutiles
- ✓ désactiver les comptes par défaut ou changer leur mot de passe
- ✓ activer et configurer la journalisation (authentifications, activités, etc.)
- ✓ définir le besoin de disponibilité et une architecture correspondante (load balancing, procédures de sauvegardes et de restauration)

Protection des données

Solutions

- ✓ créer des comptes SQL avec le moins de privilèges possibles
- ✓ utiliser le chiffrement des chmaps côté client ou par l'application si pertinent

Logiciel dans les nuages

Hébergement

- data centers en propre (*on premise*)
- data centers payants :
  - serveurs *bare metal*
  - serveurs mutualisés (machines virtuelles)
- dans les nuages (*cloud*)

Software as a Service

Principaux risques liés à une offre logicielle hébergée dans un cloud :

Software as a Service

Principaux risques liés à une offre logicielle hébergée dans un cloud :

- cloisonnement défaillant entre clients
- compromission de l'infrastructure
- perte des données des clients
- interruption globale du service

Le cloud est quelque part...

Principales mesures de protection

- WAF en amont pour détecter/empêcher les attaques
- répartition de charge et absorbeur de trafic anormal
- isolation applicative des données des clients (contrôle d'accès)
- authentification forte pour les accès administrateurs
- supervision des accès d'authentification et des clients
- sauvegardes régulières des données des clients sur un site distant

Bonnes pratiques

Déploiement

Solutions

- ✓ effectuer la revue de sécurité finale
- ✓ rédiger un guide d'installation et de configuration
- ✓ compiler sans information de débogage
- ✓ archiver les sources, l'environnement de compilation et toute la documentation interne hors ligne
- ✓ signer les binaires nécessaires sur un ordinateur hors ligne ou avec un HSM
- ✓ mettre à disposition sur un serveur sécurisé le programme, avec empreinte cryptographique et vérifier régulièrement l'intégrité du serveur
- ✓ gérer les correctifs de sécurité et versions futures

Intégration

Solutions

- ✓ changer tous les mots de passe
- ✓ appliquer le principe de moindre privilège (droits SQL, droits sur le système de fichiers, etc.)
- ✓ supprimer les fuites d'information (configuration des messages d'erreur, bannières, etc.) et désactiver le mode de débogage
- ✓ bien lire les documentations et réaliser le durcissement indiqué

Intégration

Solutions

- ✓ ne laisser accessibles aux utilisateurs que les flux strictement nécessaires : pare-feu
- ✓ bloquer l'accès aux interfaces d'administration depuis les réseaux autres que ceux des administrateurs : pare-feu, relai-inverse filtrant (*reverse proxy*) ou configuration du service Web
- ✓ mettre à jour les composants dès que nécessaire

Table des matières

Dernières formalités

- questions ?
- le mot de la fin
- quiz final QUALIOPI
- questionnaire de satisfaction
- tour de table sur votre avis et choses à améliorer