LiveCampus

Apprentissage connecté

Sommaire

LiveCampus

Apprentissage connecté

Introduction à la cybersécurité en France

LiveCampus

Apprentissage connecté

Security by Design

Approche proactive qui intègre des mécanismes de sécurité dès les premières étapes de la conception d'un système, produit ou logiciel.

Exemple : obligation d'authentification forte dès la phase de prototypage.

Avantage : évite les coûts liés à la correction tardive des failles.

LiveCampus

Apprentissage connecté

Privacy by Design

Approche complémentaire qui vise à intégrer la protection des données personnelles dès la conception d'un traitement ou d'un système.

Imposée par le RGPD (article 25).

Exemples : minimisation des données, chiffrement, pseudonymisation.

LiveCampus

Apprentissage connecté

Principes fondamentaux de sécurité

Modèle C.I.A. :

Confidentialité	Intégrité	Disponibilité
• seules les personnes autorisées peuvent accéder aux données.	• les données ne sont pas altérées de manière non autorisée.	• le service ou les données doivent rester accessibles.

LiveCampus

Apprentissage connecté

Concepts de sécurité essentiels

Défense en profondeur (Defense in Depth)

La **défense en profondeur** est une stratégie de sécurité qui repose sur l'idée que **plusieurs mécanismes de sécurité redondants et complémentaires doivent être mis en place à différents niveaux** d'un système d'information pour assurer sa protection.
Cette approche considère que **toute couche peut être contournée**, et qu'il faut donc prévoir **des barrières successives** pour ralentir, détecter, contenir ou bloquer l'attaque.

LiveCampus

Apprentissage connecté

Quelques analogies :

Un fossé : parefeu / segmentation
Un mur d'enceinte (contrôles d'accès physique ou logiques)
Des soldats de garde (IDS / IPS)
Un pont levis (politique automatiques de réponse)
Des rondes de surveillance (SIEM)

LiveCampus

Apprentissage connecté

Les objectifs :

- **Retarder la progression d'un attaquant**
- **Multiplier les points de détection**
- **Réduire les dégâts si une couche est compromise**
- **Gagner du temps afinn de pouvoir intervenir efficacement.**

LiveCampus

Apprentissage connecté

Niveau	Exemples de protection
Physique	Badge d'accès, vidéosurveillance, verrouillage des salles serveurs
Réseau	Firewall, VLAN, segmentation, VPN, détection d'intrusion réseau (NIDS)
Système (OS)	Mise à jour automatique, antivirus, restrictions sur les ports et services inutiles
Application	Filtrage des entrées, authentification forte, tests de sécurité (ex : fuzzing)
Identité & accès	MFA, politiques de mot de passe, contrôle des rôles
Données	Chiffrement, contrôle des accès aux fichiers, DLP (Data Loss Prevention)
Surveillance & réponse	SIEM, SOC, alertes automatisées, plan de réponse à incident

Couches typiques de sécurité :

LiveCampus

Apprentissage connecté

Les bonnes pratiques

Bonnes pratiques :

- Ne jamais dépendre d’une seule barrière (ex : pare-feu seul ≠ sécurité).
- Intégrer les logs et alertes à chaque niveau.
- Tester régulièrement la solidité de chaque couche (ex : tests d'intrusion).Assurer une cohérence entre les couches pour éviter les "zones grises".

LiveCampus

Apprentissage connecté

Principe du moindre privilège (Least Privilege Principle)

Définition :Le principe du moindre privilège impose que tout acteur (utilisateur, processus, machine, service) ne dispose que des autorisations minimales nécessaires à l'accomplissement de ses fonctions.

Cela limite les risques qu'un acteur puisse, volontairement ou non, compromettre un système.

Réduire la surface d'attaque.
Minimiser l'impact d'un compte compromis.
Éviter les erreurs humaines catastrophiques.
Faciliter les audits et la traçabilité.

LiveCampus

Apprentissage connecté

Les sanctions liées au RGPD

Scénario	Mauvaise pratique	Bonne pratique (principe du moindre privilège)
Un développeur teste une base de données	Accès root illimité	Compte avec droits SELECT et INSERT uniquement sur un schéma de test
Un service automatise une sauvegarde	Compte avec droits d'administration système	Compte de service avec droit d'écriture dans un dossier dédié
Un stagiaire accède à des documents internes	Accès aux répertoires stratégiques de l'entreprise	Accès uniquement aux ressources liées à son stage

LiveCampus

Apprentissage connecté

Mise en œuvre :

- **Utiliser des rôles (RBAC)** plutôt que des autorisations individuelles.
- **Segmenter les comptes** : un compte pour l'utilisateur classique, un autre pour les tâches d'admin.
- **Configurer les comptes de service** avec des droits minimums.
- **Limiter la durée de validité des droits temporaires**.
- **Auditer régulièrement les droits accordés.**

LiveCampus

Apprentissage connecté

À retenir :

Ces deux concepts sont complémentaires :
On met plusieurs barrières (défense en profondeur) et on restreint ce que chaque entité peut faire (moindre privilège).

LiveCampus

Apprentissage connecté

Typologie des menaces
OWASP TOP 10. Le dernier publié est en 2023.

Liste actualisée des 10 risques les plus critiques pour les applications web :

1. Broken Access Control
2. Cryptographic Failures
3. Injection (ex : SQL, NoSQL)
4. Insecure Design
5. Security Misconfiguration

LiveCampus

Apprentissage connecté

6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. SSRF (Server-Side Request Forgery)

LiveCampus

Apprentissage connecté

Explication des différentes menaces

Broken Access Control

Les restrictions d'accès ne sont pas correctement appliquées.

Exemple :Un utilisateur non autorisé accède à des pages d'administration (/admin) via modification d'URL (IDOR).Suppression de données d'autres utilisateurs via API.

Risques :
Violation de données, usurpation d'identité, escalade de privilèges

LiveCampus

Apprentissage connecté

Recommandations :

- Implémenter un contrôle d’accès côté serveur, pas seulement dans l'interface.
- Désactiver les fonctions masquées plutôt que de les cacher.
- Journaliser et alerter les accès anormaux.

LiveCampus

Apprentissage connecté

Cryptographic Failures (anciennement "Sensitive Data Exposure")

Les données sensibles ne sont pas protégées correctement.

Exemple :Transmission de données en clair (HTTP au lieu de HTTPS).Mauvais algorithme de chiffrement (ex : MD5, SHA-1).Mauvaise gestion des clés (stockées en clair).

Risques :Vol d'identifiants, usurpation d'identité, non-conformité RGPD.

LiveCampus

Apprentissage connecté

Recommandations :

Utiliser TLS 1.2+ pour les communications.
Chiffrer les données sensibles au repos et en transit.
Gérer les clés et secrets avec un coffre-fort (ex : HashiCorp Vault).

LiveCampus

Apprentissage connecté

Injection

Des données non fiables sont traitées comme du code ou des requêtes.
Exemple :SQL Injection : OR '1'='1 dans un champ de login.

Command Injection : ; rm -rf / via une API mal protégée.LDAP, NoSQL, OS Command, XML injections.

Risques :Lecture, modification ou suppression de données.
Prise de contrôle du serveur.

LiveCampus

Apprentissage connecté

Recommandations :

- Utiliser des requêtes préparées (paramétrées).
- Valider et filtrer les entrées.
- Éviter l'interprétation dynamique de code ou commandes.

LiveCampus

Apprentissage connecté

Exemple d'une injection

Une application web propose un formulaire de connexion simple avec deux champs :Nom d'utilisateurMot de passe
La requête SQL côté serveur ressemble à ceci (en pseudo-code PHP/MySQL) :

```
$query = "SELECT * FROM utilisateurs WHERE username = '$username' AND password = '$password'";
```

Aucune **validation** ni **sanitisation** (filtrer, nettoyer les entrées utilisateur)des entrées n'est effectuée.

LiveCampus

Apprentissage connecté

Injection SQL — Exemple d’exploitation

Objectif de l'attaquant :Contourner l'authentification sans connaître les identifiants.

Entrée utilisateur malveillante :
Nom d'utilisateur : ' OR '1'='1
Mot de passe : peu importeLa requête devient :

LiveCampus

Apprentissage connecté

```
SELECT * FROM utilisateurs WHERE username = '' OR '1'='1' AND password = 'peuimporte';
```

Or, '1'='1' est toujours vrai, donc la requête retourne tous les utilisateurs, souvent le premier dans la base (admin).

LiveCampus

Apprentissage connecté

Insecure Design

Absence de réflexion sécurité dès la phase de conception.

Exemple :Application qui ne prévoit aucun mécanisme de verrouillage de compte après plusieurs tentatives d’échec.

Aucune séparation des rôles utilisateurs.

Risques :Application intrinsèquement vulnérable, difficile à corriger.

LiveCampus

Apprentissage connecté

Recommandations :

- Utiliser des modèles de menaces (threat modeling).
- Pratiquer la revue d'architecture.
- Appliquer les principes de Security by Design.

LiveCampus

Apprentissage connecté

Dora dans le détail

Security Misconfiguration

Mauvaise configuration des composants logiciels ou du serveur.

Exemple :Console d'administration accessible sans mot de passe.
Messages d'erreur détaillés affichés en production.

Ports ou services non utilisés laissés ouverts.

Risques :Exposition à des attaques automatisées ou ciblées.

LiveCampus

Apprentissage connecté

Recommandations :

- Définir une configuration standard sécurisée (baseline).
- Automatiser le déploiement sécurisé (IaC).
- Désactiver les services inutiles.

LiveCampus

Apprentissage connecté

Vulnerable and Outdated Components

Utilisation de bibliothèques, frameworks ou serveurs vulnérables ou obsolètes.

Exemple :Version vulnérable de jQuery, Log4j ou Apache Struts utilisée.

Pas de politique de mise à jour.

Risques :Exploitation de failles connues sans interaction de l'utilisateur.

LiveCampus

Apprentissage connecté

Recommandations :

- Maintenir un inventaire des composants.
- Utiliser des outils d’analyse de dépendances (ex : npm audit, Snyk, Trivy).
- Mettre à jour régulièrement les librairies.

LiveCampus

Apprentissage connecté

Identification and Authentication Failures

Failles dans les mécanismes d’authentification ou d’identification.

Exemple :Mots de passe faibles ou non hachés.Absence de MFA (authentification multifactorielle).

Sessions non invalidées après logout.

Risques :Compromission de comptes utilisateurs, escalade de privilèges.

LiveCampus

Apprentissage connecté

Recommandations :

- Implémenter le MFA.Stocker les mots de passe avec un algorithme robuste (ex : bcrypt).
- Gérer correctement les sessions (tokens, expiration, révocation).

LiveCampus

Apprentissage connecté

Software and Data Integrity Failures

Absence de vérification de l'intégrité des logiciels ou données critiques.

Exemple :
•Déploiement d'un script distant non vérifié.
•Pipeline CI/CD compromis.

Risques :
•Installation de malwares, supply chain attack.

LiveCampus

Apprentissage connecté

Recommandations :

- Signer les composants logiciels.
- Utiliser des hashes de vérification.
- Sécuriser les chaînes DevOps.

LiveCampus

Apprentissage connecté

Security Logging and Monitoring Failures

Absence de journaux, d’alertes ou de surveillance adéquats.

Exemple :Aucune trace des connexions ou tentatives d'accès.

Logs non centralisés, non analysés.

Risques :Attaques non détectées, difficulté d’analyse post-incident.

LiveCampus

Apprentissage connecté

Recommandations :

- Activer la journalisation des événements critiques.
- Centraliser et corréler les logs (ex : SIEM).
- Mettre en place des alertes et un plan de réponse à incident.

LiveCampus

Apprentissage connecté

SSRF – Server-Side Request Forgery

L'application permet à un attaquant de faire des requêtes HTTP depuis le serveur vers des ressources internes.

Exemple :L'application interroge une URL donnée par l'utilisateur (GET /fetch?url=...), ce qui permet de cibler localhost:8080 ou les métadonnées cloud (169.254.169.254).

Risques :Accès à des services internes non exposés.
Exfiltration de données sensibles ou attaques réseau internes.

LiveCampus

Apprentissage connecté

Recommandations :

- Filtrer les URL autorisées (whitelisting).
- Interdire les connexions vers les IP privées ou localhost.
- Utiliser des proxies sécurisés ou sandbox.

LiveCampus

Apprentissage connecté

Étude de cas : la faille Equifax (2017)

Contexte :

- Fuite de **147 millions de données personnelles**.
- Exploitation d'une vulnérabilité Apache Struts (CVE-2017-5638).
- Patch disponible **plusieurs mois avant** l'attaque.

LiveCampus

Apprentissage connecté

Étude de cas : la faille Equifax (2017)

Analyse :Non-application des correctifs de sécurité (patch management déficient).

Manque de supervision des systèmes vulnérables.

Absence de stratégie "Security by Design".

LiveCampus

Apprentissage connecté

Étude de cas : la faille Equifax (2017)

Leçons :

- Mettre en place une **politique de mise à jour systématique**.
- Documenter et **gérer les composants open source utilisés**.
- Automatiser la **détection de vulnérabilités connues** (ex : via Trivy ou Snyk).
- Lier le processus de développement à la veille sécurité.

LiveCampus

Apprentissage connecté

Étude de cas : la faille Equifax (2017)

Activité possible :

Mini débat en classe : "Est-ce qu'Equifax aurait pu éviter l'attaque si elle avait appliqué le principe de Security by Design ? Pourquoi ?"

LiveCampus

Apprentissage connecté

Petit outil intéressant

https://bonjourlafuite.eu.org/

LiveCampus

Apprentissage connecté

Cas pratique