

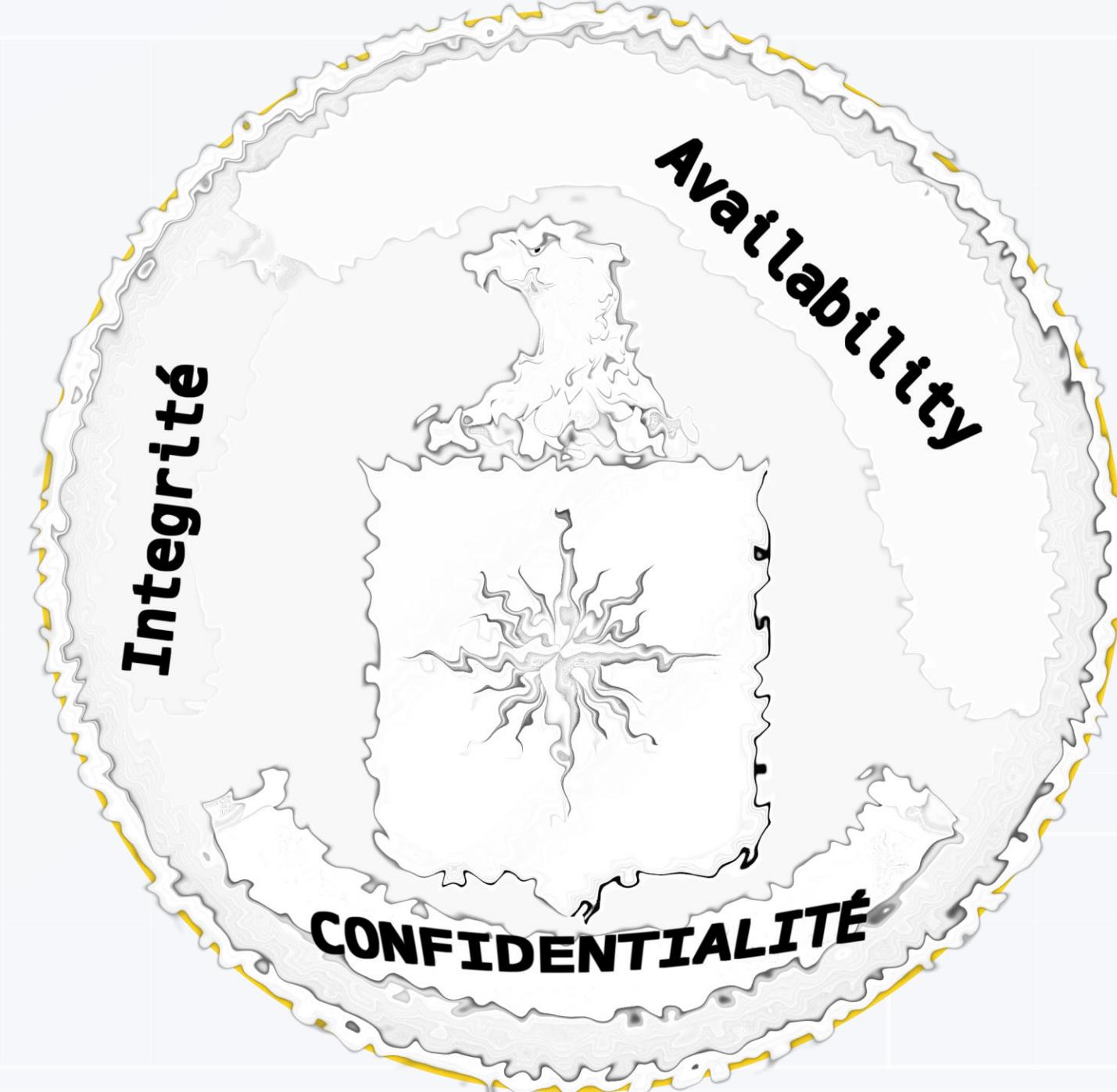
HD

MAJESTIC
MOVIES

10
MILLION+
VIEWS

CIA AGENT

ENGLISH MOVIE



TOP 10 des risques de sécurité

Les objectifs

- A la fin de la séquence, vous serez capables de :
- Citer les 10 risques de sécurité les plus répandus dans l'ordre de risque (top 10)

Les objectifs

- A la fin de la séquence, vous serez capables de :
- Citer les 10 risques de sécurité les plus répandus dans l'ordre de risque (top 10)
- Expliquer avec ses propres mots le principe de chacun de ces risques

Les objectifs

- A la fin de la séquence, vous serez capables de :
- Citer les 10 risques de sécurité les plus répandus dans l'ordre de risque (top 10)
- Expliquez avec ses propres mots le principe de chacun de ces risques
- Expliquer avec ses propres mots les contre-mesures à ces risques

Les objectifs

- A la fin de la séquence, vous serez capables de :
- Citer les 10 risques de sécurité les plus répandus dans l'ordre de risque (top 10)
- Expliquer avec vos propres mots le principe de chacun de ces risques
- Expliquer ses vos propres mots les contre-mesures à ces risques

Un moyen d'éviter les radars



Activité

- Sans moyen d'aide extérieure
- Par groupe de : 3
- Réflexion :
 - Sur les différentes **causes** de risques qui pourraient nuire à l'application (effet direct)
 - Sur les différents risques qui pourraient impacter l'entreprise et ses revenus (effet indirect)
- Temps à disposition : 15 mn
- Mettez les post-it sur le tableau blanc

Importance de la sécurité

- Fuite de données personnelles et sensibles
- Réputation
- Conséquences judiciaires et pénales
- Statistiques :
 - 2365 attaques en 2023, impactant 343'338'964 personnes
 - Une fuite de donnée coûte 4.45 mio USD en moyenne
 - Phishing, Malware, Ddos, fuite de données
 - Coût total estimé en 2025 : 10.5 trillions USD

OWASP

OWASP

- **Open Web Application Security Project**
- Organisation internationale à but non lucratif
- Sécurité des applications Web
- Leur projet le plus connu est l'OWASP Top 10
- Vision : «*No more insecure software*»

Le top 10

1. Broken Access Control

Le top 10

1. Broken Access Control
2. Cryptographic failures

Le top 10

1. Broken Access Control
2. Cryptographic failures
3. **Injection**

Le top 10

1. Broken Access Control
2. Cryptographic failures
3. Injection
4. Insecure Design

Le top 10

1. Broken Access Control
2. Cryptographic failures
3. Injection
4. Insecure Design
5. Security Misconfiguration

Le top 10

1. Broken Access Control
2. Cryptographic failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components

Le top 10

1. Broken Access Control
2. Cryptographic failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures

Le top 10

1. Broken Access Control
2. Cryptographic failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures

Le top 10

1. Broken Access Control
2. Cryptographic failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. **Security Logging and Monitoring Failures**

Le top 10

1. Broken Access Control
2. Cryptographic failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery (SSRF)

Le top 10

1. Broken Access Control
2. Cryptographic failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery (SSRF)

Activité 1 : Définitions

Le top 10

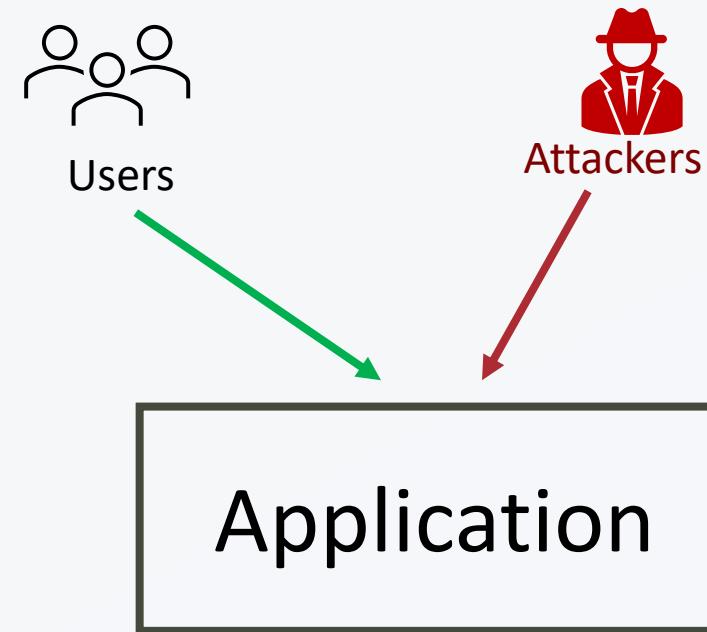
- Travail individuel
- Temps à disposition : 5 mn
- E-183-ALL-OWASP-Definition.docx (imprimé)

1. Broken Access Control

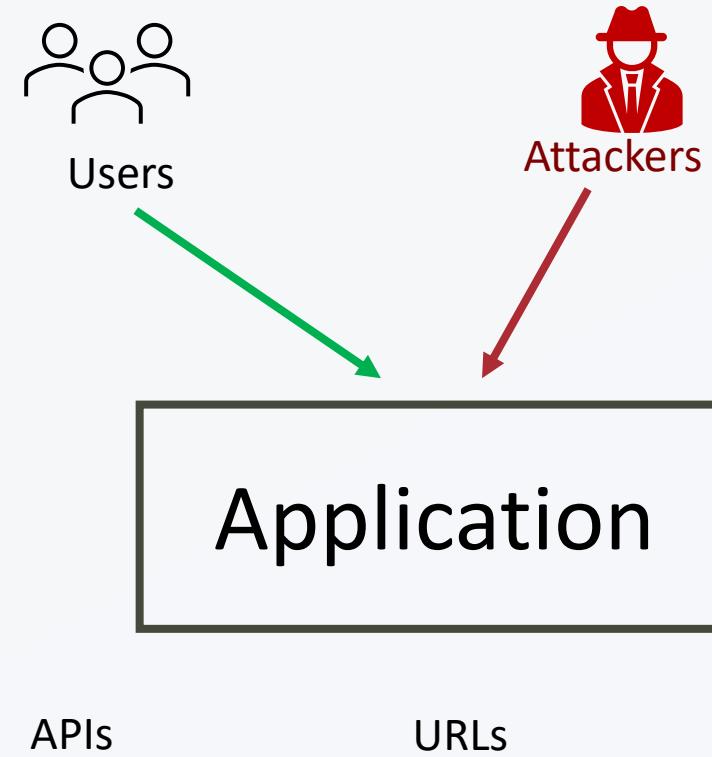
Contrôle d'accès défaillant

- Un contrôle d'accès est défaillant lorsque des utilisateurs accèdent à des informations qui ne leur sont pas destinées.
- Par exemple :
 - Un document est imprimé et est visible de tous
 - Un e-mail est envoyé au mauvais destinataire
 - Une page personnelle de votre système est accessible à un autre utilisateur

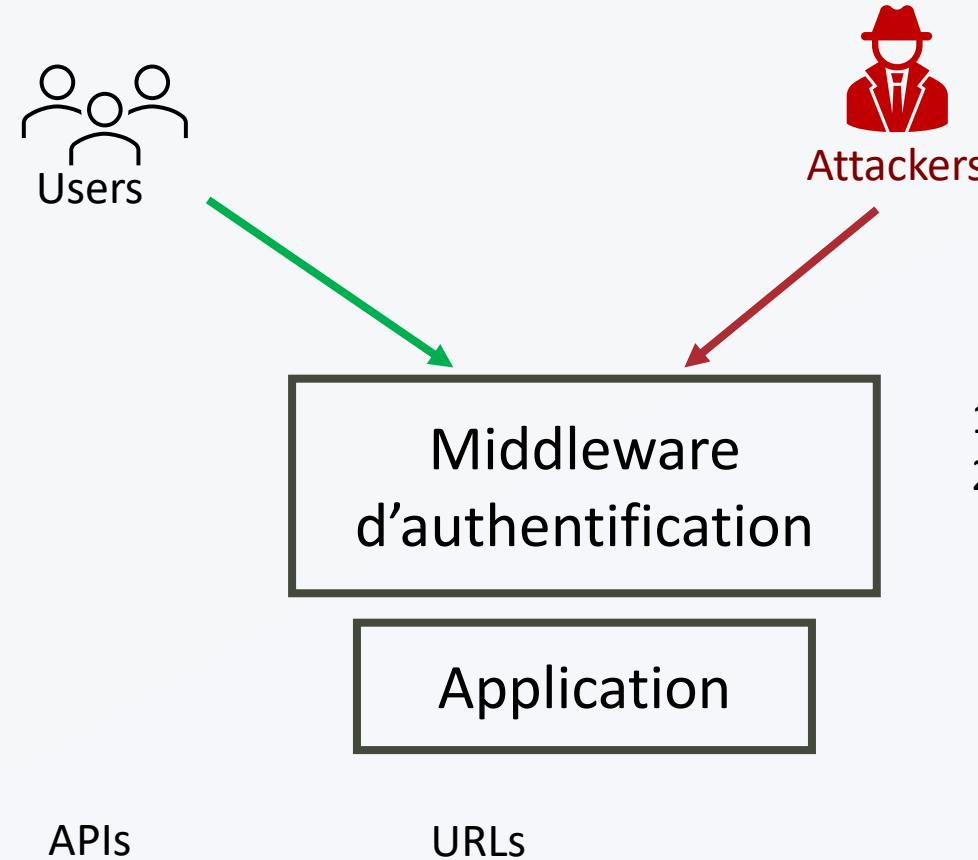
Principle



Principle



Principe



1. Vérifie que l'utilisateur soit authentifié
2. Vérifie que l'utilisateur utilise l'application dans la limite de ses droits

2. Cryptographic Failures

Défaillances cryptographiques

- Vulnérabilité liée à l'absence ou à un mauvais chiffrement des données.

Défaillances cryptographiques

- Est-ce que les données sensibles sont en clair dans le stockage (par exemple : mot de passe en clair dans la table users de la db) ?
- Est-ce que les fonctions de hachages sont sûres ? (MD5 ? SHA-1 ? SHA-256 ? SHA-512 ?)
 - Risques de collision ?
- Est-ce que l'aléatoire est vraiment aléatoire ?

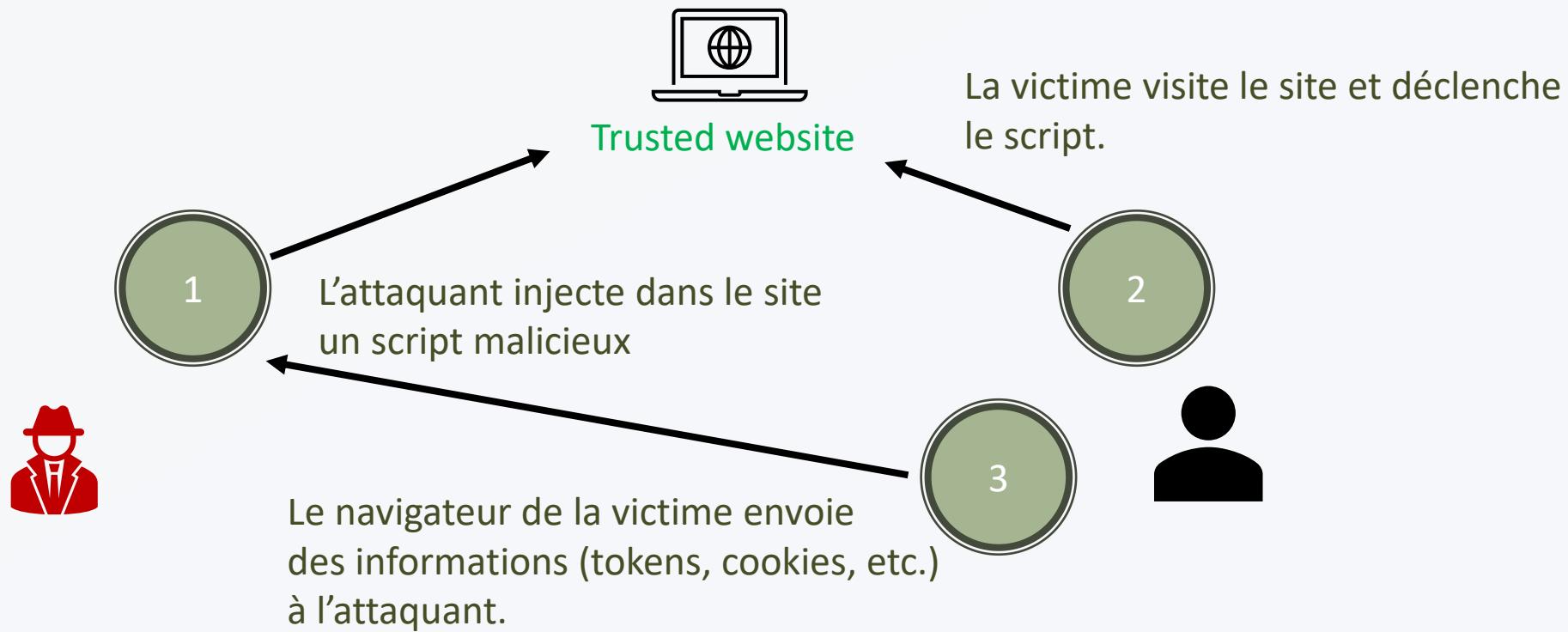
3. Injection

Injection

- Une vulnérabilité qui permet à un attaquant de relayer du code malveillant via une application vers un autre système.
- Cela peut inclure la compromission des systèmes back-end ainsi que d'autres clients connectés à l'application vulnérable.
- Effets :
 - Permettre à un attaquant d'exécuter des appels du système d'exploitation sur une machine cible
 - Permettre à un attaquant de compromettre les bases de données back-end
 - Permettre à un attaquant de compromettre ou de détourner les sessions d'autres utilisateurs
 - Permettre à un attaquant de forcer des actions au nom d'autres utilisateurs ou services

Injection

- Cross-site scripting (XSS) :



Injection SQL

- Injection SQL :
 - Supposons le code suivant :
 - `SELECT ALL FROM t_users WHERE idUser = «Valeur du formulaire»`
 - Que pourrait-il mal se passer ?

Injection

- «Valeur du formulaire» pourrait être :
- “ OR 1 = 1
 - Le code suivant serait exécuté :
 - SELECT ALL FROM t_users WHERE idUser = “ OR 1 = 1
 - Que retournerait la commande ?

Injection

- «Valeur du formulaire» pourrait être :
- “”; DROP DATABASE ...; --
 - Le code suivant serait exécuté :
 - SELECT ALL FROM t_users WHERE idUser = “”; DROP DATABASE ...; --
 - Que retournerait la commande ?

Injection : se prémunir

- Valider les entrées de formulaire
- Appliquer le principe de moindres privilèges
- Eviter de passer par des services externes
- Tester les retours de fonctions
- Vérifier les exceptions

4. Insecure Design

Conception non-sécurisée

- Vulnérabilité liée à la non-prise en considération des principes de sécurité dès les premières étapes du processus de conception de l'application.
- Analogie avec la vie de tous les jours :
 - Un constructeur automobile n'a pas pris en compte le fait que la ceinture de sécurité devait résister à un choc de plus de 1000N
 - En cas de choc, même si l'automobiliste a parfaitement bouclé sa ceinture, la sécurité est compromise.
 - On distingue ici l'implémentation de la conception.

Conception non-sécurisée

- Catégorie assez vaste
- Englobe les défauts architecturaux de votre application
- Historiquement :
 - On planifie → On développe → On teste → On livre → **On corrige la sécurité**
- Shift left :
 - On planifie → **On prévoit la sécurité** → On développe → On teste → On livre

Conception non-sécurisée

- Quel est le risque de sécurité avec les «Questions secrètes» qui permettent de récupérer un compte ?

Conception non-sécurisée

- Questions secrètes :
- Les questions et les réponses ne peuvent pas être considérées comme une preuve d'identité, car plus d'une personne peut connaître les réponses.
- Un tel code doit être supprimé et remplacé par une conception plus sécurisée.

Conception non-sécurisée

- Que se passerait-il si vous faîtes une application web pour une chaîne de restaurants sans prendre en compte la vraisemblance des réservations ?

Conception non-sécurisée

- Les attaquants pourraient modéliser ce cas d'utilisation (use case) et tester s'ils peuvent réserver six cents places et tous les restaurants à la fois en quelques demandes, provoquant une perte massive de revenus et/ou un chaos dans les réservations.
- Un dégât d'image aurait aussi lieu.

Conception non-sécurisée : se prémunir

- Implémenter la sécurité dès le début du développement (shift left)
- Modéliser les menaces
- Intégrer les contrôles de sécurité dans les User Stories
- Restreindre les ressources par service ou utilisateur
- Intégrer des contrôles de vraisemblance

5. Security Misconfiguration

Mauvaise configuration de sécurité

- Vulnérabilité liée :
 - à l'absence de hardening, fonctionnalités inutiles ou inactives
 - Comptes et mots de passe par défaut
 - Erreurs de stack toujours affichées en production

6. Vulnerable and Outdated Components

Composants vulnérables et obsolètes

- Selon OWASP
 - Votre application est vulnérable si vous ne connaissez pas toutes les versions de vos composants et si ils sont vulnérables ou dépassés
 - Webapp composée de nombreuses modules et librairies open-source ou d'une tierce partie
 - Comme ces modules sont exécutés dans votre application, ils ont les mêmes accès que votre code. Donc un impact potentiellement très grand.

Composants vulnérables et obsolètes

- Mettre à jour des composants coûteux du temps
- Dans les grandes entreprises, la mise à jour d'un composant doit être acceptée par plusieurs personnes / équipes / départements
- Compromis de productivité
 - Le temps pris pour améliorer la sécurité est du temps qu'on ne pourra pas utiliser pour développer de nouveaux produits
 - Chaque entreprise doit faire un choix en gérant ses risques

Composants vulnérables et obsolètes

- Solution
 - Connaître l'état de vos applications, modules externes, librairies
 - Connaître leur versions, vulnérable ou pas
 - Tester pro activement votre application
 - Mettre à jour continuellement vos applications et implémenter les patchs si nécessaire
- C'est un problème d'organisation/personnes/process plus qu'un problème technique!

7. Identification and Authentication Failures

Echec d'identification ou d'authentification

- Selon OWASP
 - Confirmer l'identité, ses autorisations et la gestion de sa session est un processus critique pour se protéger des attaques liées à ce domaine.
- Une webapp doit confirmer qui vous êtes et qui vous prétendez être
- Problèmes
 - Une webapp demande une identification mais ne vérifie pas que l'identité est correcte

Echec d'identification ou d'authentification

- Problème 1
 - Une webapp demande une identification mais ne vérifie pas que l'identité est correcte
 - Analogie de la personne qui prétend être pilote
 - Exemple du code SMS renvoyé mais qui n'est pas contrôlé par l'application
- Problème 2
 - Une webapp communique avec un serveur et demande un certificat, mais ne vérifie pas que le certificat vient bien du serveur
 - On peut vous envoyer vers un autre site web frauduleux (XSS)
- Problème 3
 - Une webapp ouvre une nouvelle session mais ne ferme pas correctement la précédente
 - Analogie de la pompe à essence
 - Log in sur un ordinateur public et vous oubliez de vous déconnecter

Echec d'identification ou d'authentification

- Solutions
 - Une webapp doit identifier et authentifier proprement les utilisateurs
 - Semble simple mais peut être très compliqué
- Exemples complexes
 - Gestion des SSO
 - Compte Office / Eduvaud ?
 - Compte Gmail

8. Software and Data Integrity Failures

Défaillance des logiciels et de l'intégrité de leurs données

- Selon OWASP,
 - CI/CD non sécurisé : automatisation des processus sans processus de vérification suffisant
 - Mise à jour/téléchargement de composant automatique
- Lié au 6^e point! Mais différencié pour souligner et mettre en avant les risques liés aux cycles rapides de développement

Défaillance des logiciels et de l'intégrité de leurs données

- Analogie de la fabrication de chocolat
 - Si les ingrédients sont bons, le chocolat est bon
 - Mais si les ingrédients sont mauvais?
- Dans une webapp, quels sont les ingrédients?
- Autre problème
 - Si une mise à jour d'un logiciel externe est buggée?

Défaillance des logiciels et de l'intégrité de leurs données

- Solutions

- Être conscient des risques liés aux processus automatisés dans le cycle de développement des applications
- S'assurer que ces processus sont vérifiés et contrôlés régulièrement

9. Security Logging and Monitoring Failures

Echec des logs de sécurité et de surveillance

- Selon OWASP,
 - Sans journalisation/logging et surveillance/monitoring, les failles ne peuvent pas être détectées
- Les problèmes de sécurité sont inévitables
- Mais :
 - Hacker un système prend du temps, ce n'est pas Matrix ou Mr Robot!
 - Il faut un certain nombre d'étapes

Echec des logs de sécurité et de surveillance

- Analogie du cambrioleur
- Qu'est-ce qui aurait pu empêcher le cambriolage de continuer?

Echec des logs de sécurité et de surveillance

- Journaliser et surveiller nous permet de stopper l'attaque
- UK National Cyber Security Center
 - <https://www.ncsc.gov.uk/guidance/introduction-logging-security-purposes>
- Exemples
 - Qui a consulté ou téléchargé un fichier spécifique ?
 - Y a-t-il eu des tentatives d'authentification incorrectes ?
 - Qui s'est connecté récemment ?
 - Des événements d'authentification se sont-ils produits à des moments inattendus ou à partir d'emplacements inattendus ?

Echec des logs de sécurité et de surveillance

- Solutions
 - Log, surveillance et alertes
 - Permettent aux équipes chargées de la surveillance de détecter une attaque ou une brèche
 - Permettent aux équipes de stopper l'attaque avant de faire plus de dommage

10. Server-Side Request Forgery (SSRF)

Falsification des requêtes serveur

- SSRF permet à l'attaquant de contraindre l'application d'envoyer une requête à une destination inattendue.
- Cela permet à l'attaquant de se faire passer pour le serveur victime et atteindre des API internes, des bases de données.
- SSRF se produit lorsqu'un serveur veut atteindre des ressources externes, une image par exemple.

Falsification des requêtes serveur

- Solutions
- Only-allowed ou not-Allowed liste doit être appliquée
- Webapps sont vulnérables aux attaques SSRF

Activité 2 : reconnaisances et contre-mesures

Reconnaissance et contre-mesures

- Travail en groupe de deux sur un risque donné du top 10.
- A partir du site <https://owasp.org/Top10/fr/>
- Prenez une CWE (common weakness enumeration) liée à votre risque et compléter la présentation.
- A produire :
 - Un document synthétique basé sur le modèle «E-183-ALL-RisqueN.pptx»
 - Vous renomerez le document (RisqueN sera remplacé par le **nom du risque traité**)
 - Présentation à laisser dans Teams
- Temps à disposition : 30 mn
- Rendu : présentation à la classe (5-7 mn max / risque)

Activité 3 : Attaques informatiques récentes

Attaques informatiques récentes

- Travail individuel
- Trouvez une attaque informatique récente et notez-la sur un post-it
- Situez cette attaque dans le risque OWASP
- Donnez la cause et quelques conséquences

Conclusion

Mot de la fin

- Reprenons vos post-it (activité 1)
- Placez-les dans les catégories du top 10 OWASP
- Qu'avez-vous appris ?
- Quel sentiment avez-vous maintenant ?