

Cybersécurité BUT3

Maxime MINGUELLA
2024-2025



Au menu de ce mois



02/10

MITRE ATT&CK Framework

Découvrir la CTI et la sécurité offensive de base

09/10

Défense en profondeur

Un aperçu plus approfondi des principes de sécurité et de l'architecture sécurisée

16/10

SIEM

Comment fonctionne un SOC ?

23/10

ANALYSE DES RISQUES + RUMPS

Introduction à la méthode EBIOS RM et évaluation RUMP

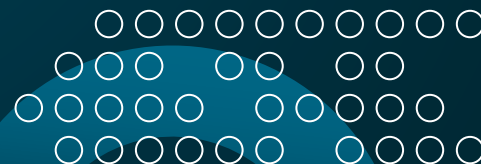




Who am I?

Maxime MINGUELLA

- Ingénieur cybersécurité
- Product Security Specialist OT @ Airbus



Évaluation

- 1/3 CM (interactif)
- 2/3 TD/TP (<https://tryhackme.com>)
- Des labs à réaliser sur Tryhackme avant l'évaluation finale
- Un RUMP à réaliser lors de l'évaluation finale (oral de 5 minutes sur un sujet choisi par vos soins → validation par email à me faire maxime.minguella@univ-amu.fr avant le 09/10/2024)
- Un examen écrit sur le contenu du cours (QCM + questions)

Disclaimer 😊

Article 323-1

Version en vigueur depuis le 26 janvier 2023

Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de trois ans d'emprisonnement et de 100 000 € d'amende.

Lorsqu'il en est résulté soit la suppression ou la modification de données contenues dans le système, soit une altération du fonctionnement de ce système, la peine est de cinq ans d'emprisonnement et de 150 000 € d'amende.

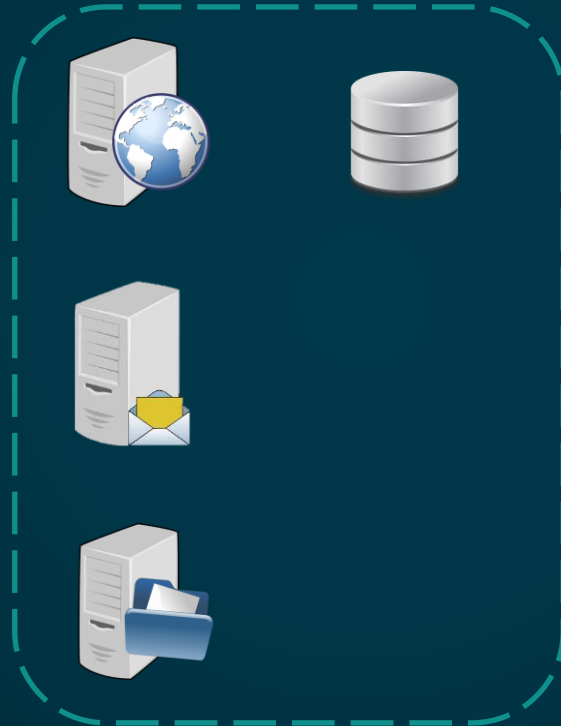
Lorsque les infractions prévues aux deux premiers alinéas ont été commises à l'encontre d'un système de traitement automatisé de données à caractère personnel mis en œuvre par l'Etat, la peine est portée à sept ans d'emprisonnement et à 300 000 € d'amende.

Mais aussi les article 323-1 à 8...

01

Mitre Att&ck Framework

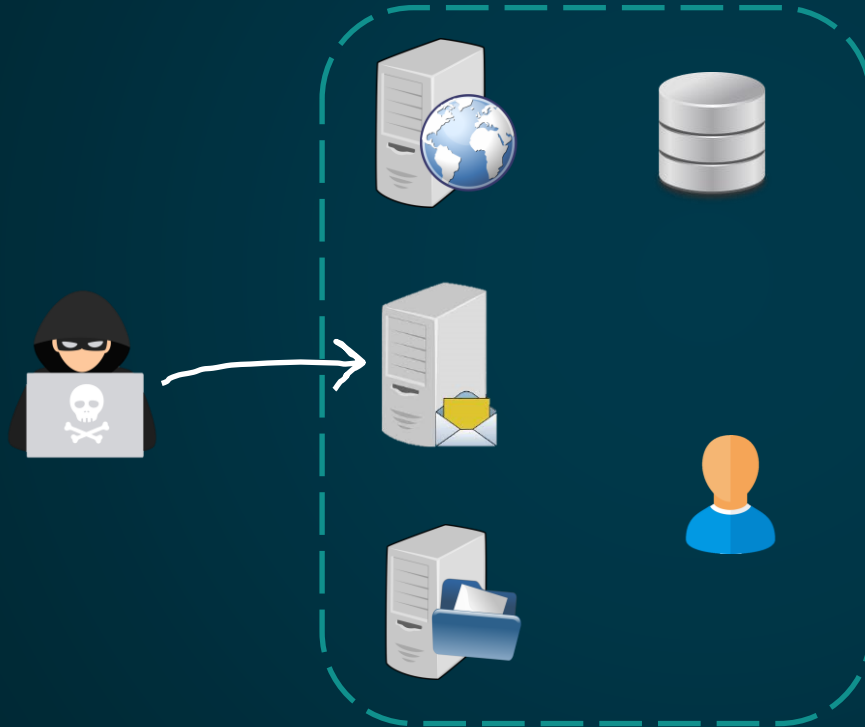
Exemple de scénario



Imaginons un scénario comme celui-ci avec :

- Un serveur de mail ;
- Un serveur de fichier;
- Un serveur web et sa base de données *high level*.

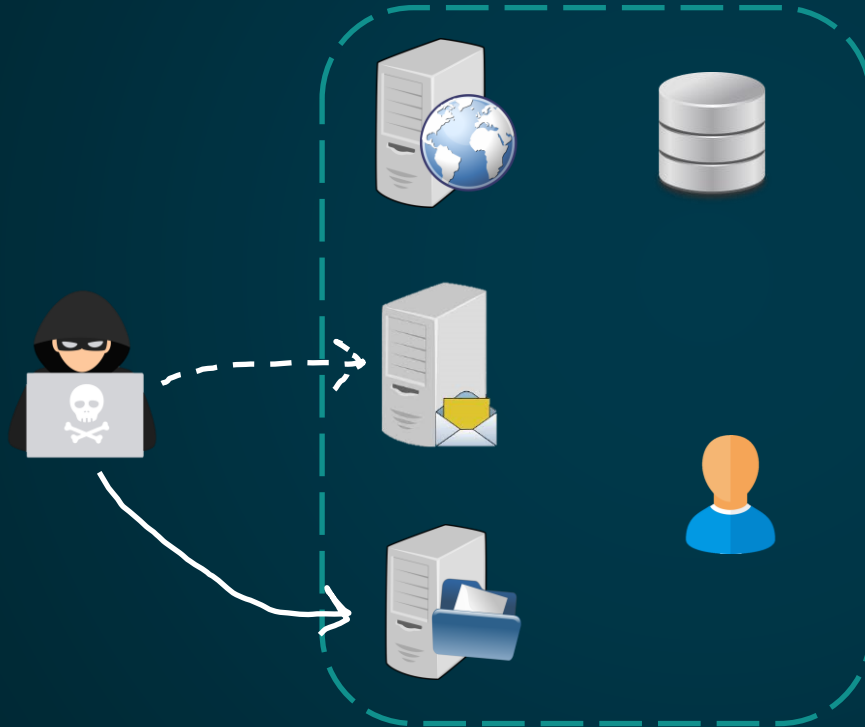
Exemple de scénario



Notre attaquant commence une phase de reconnaissance, avec un scan, et remarque qu'il y a un serveur de mail.

Il envoie donc un email de Phishing pour encourager le salarié naïf à cliquer sur un lien.

Exemple de scénario

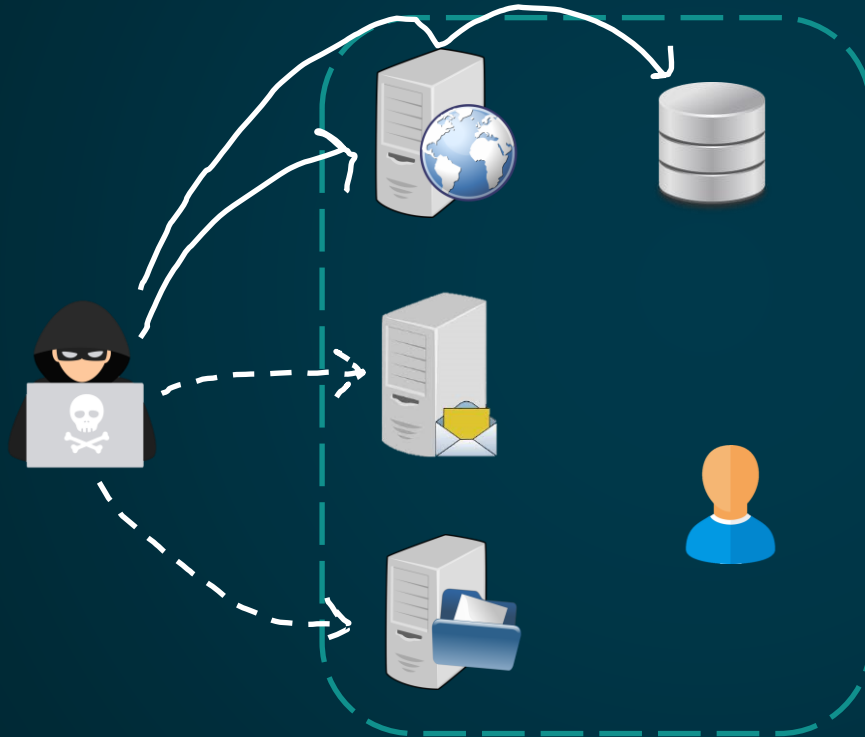


Maintenant qu'il a accès aux emails de notre victime, il se dit : « j'ai aussi découvert qu'il y avait un serveur de fichier, je vais voir si je peux m'y connecter ».

Bingo même identifiant, même mot de passe.

Sur ce serveur se trouve un fichier Excel non chiffré avec tous les mots de passe de la victime.

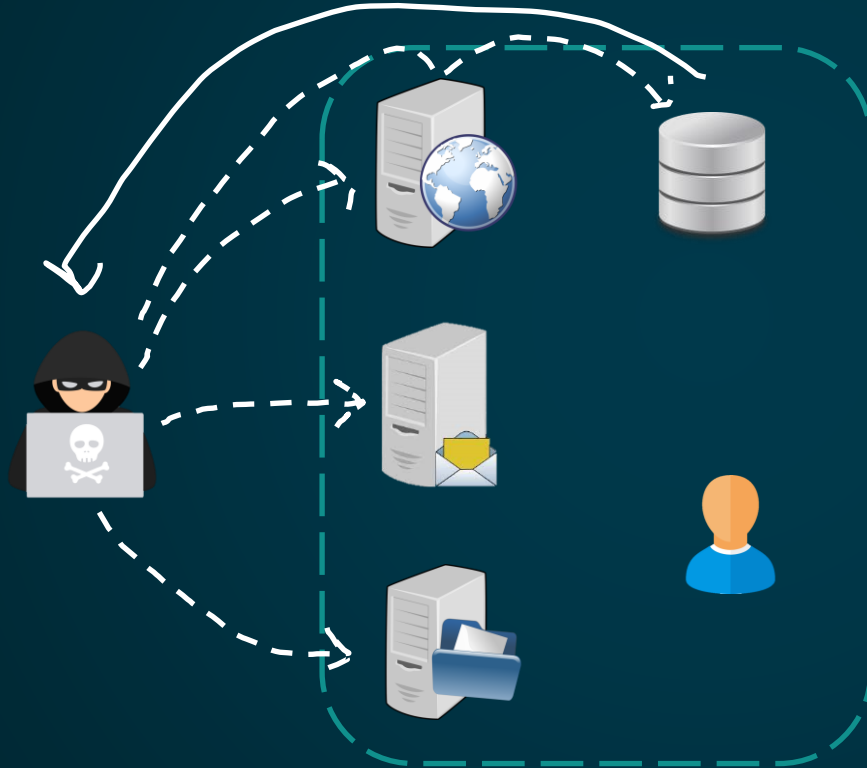
Exemple de scénario



Une fois ces nouveaux mot de passes connus, l'attaquant va essayer d'accéder au serveur web et donc à la base de données.

Bingo ! Pléthore d'information *high level* sur la société dans la base données.

Exemple de scénario



L'attaquant va ensuite exfiltrer les informations de la société vers son ordinateur.

Enfin, cerise sur le gâteau, il va détruire les données et laisser la société les mains vides.

« Il a les données de votre société, et vous pas »

MITRE ATT&CK Framework

MITRE ATT&CK®

Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
10 techniques	8 techniques	10 techniques	14 techniques	20 techniques	14 techniques	43 techniques	17 techniques	32 techniques	9 techniques	17 techniques	18 techniques	9 techniques	14 techniques
Active Scanning (3)	Acquire Access (3)	Content Injection (3)	Cloud Administration Command (3)	Account Manipulation (4)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Brute Force (4)	Account Discovery (4)	Exploitation of Remote Services (4)	Adversary-in-the-Middle (4)	Application Layer Protocol (4)	Automated Exfiltration (4)	Account Access Removal (4)
Gather Victim Host Information (4)	Acquire Infrastructure (4)	Drive-by Compromise (4)	Command and Scripting Interpreter (13)	BITS Jobs (4)	Access Token Manipulation (4)	Access Token Manipulation (4)	Credentials from Password Stores (4)	Browser Information Discovery (4)	Internal Spearphishing (4)	Archive Collected Data (4)	Communication Through Removable Media (4)	Data Transfer Size Limits (4)	Data Destruction (4)
Gather Victim Identity Information (3)	Compromise Accounts (2)	Exploit Public-Facing Application (4)	Container Administration Command (4)	Boot or Logon Autostart Execution (14)	Account Manipulation (4)	BITS Jobs (4)	Exploitation for Credential Access (4)	Cloud Infrastructure Discovery (4)	Remote Service Session Hijacking (2)	Automated Collection (4)	Content Injection (4)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact (4)
Gather Victim Network Information (4)	Compromise External Remote Services (4)	External Remote Services (4)	Deploy Container (4)	Boot or Logon Initialization Scripts (2)	Boot or Logon Autostart Execution (14)	Debugger Evasion (4)	Forced Authentication (4)	Cloud Service Dashboard (4)	Remote Services (4)	Browser Session Hijacking (4)	Data Encoding (2)	Exfiltration Over C2 Channel (4)	Data Manipulation (3)
Gather Victim Org Information (4)	Develop Capabilities (4)	Hardware Additions (4)	Exploitation for Client Execution (4)	Browser Extensions (4)	Boot or Logon Initialization Scripts (2)	Deobfuscate/Decode Files or Information (4)	Forge Web Credentials (2)	Cloud Service Discovery (4)	Replication Through Removable Media (4)	Cliptext Data (4)	Data Obfuscation (3)	Dynamic Resolution (2)	Defacement (2)
Phishing for Information (4)	Establish Accounts (3)	Phishing (4)	Inter-Process Communication (2)	Compromise Host Software Binary (4)	Boot or Logon Initialization Scripts (2)	Deploy Container (4)	Input Capture (4)	Cloud Storage Object Discovery (4)	Software Deployment Tools (4)	Data from Cloud Storage (4)	Encrypted Channel (2)	Exfiltration Over Other Network Channel (2)	Disk Wipe (2)
Search Closed Sources (2)	Obtain Capabilities (2)	Replication Through Removable Media (4)	Native API (4)	Create or Modify System Processes (4)	Create or Modify System Processes (4)	Domain or Tenant Policy Modification (2)	Modify Authentication Process (4)	Container and Resource Discovery (4)	Taint Shared Content (4)	Data from Configuration Repository (2)	Fallback Channels (4)	Exfiltration Over Physical Medium (1)	Endpoint Denial of Service (4)
Search Open Technical Databases (2)	Supply Chain Compromise (3)	Scheduled Task/Job (3)	Trusted Relationship (4)	Event Triggered Execution (13)	Event Triggered Execution (13)	Execution Guardrails (1)	Multi-Factor Authentication Interception (4)	Debugger Evasion (4)	Use Alternate Authentication Material (4)	Data from Local System (4)	Hide Infrastructure (4)	Exfiltration Over Web Service (4)	Financial Theft (4)
Search Open Websites/Domains (2)	Valid Accounts (4)	Valid Accounts (4)	Shared Modules (4)	External Remote Services (4)	Exploitation for Privilege Escalation (4)	File and Directory Permissions Modification (2)	Network Sniffing (4)	Domain Trust Discovery (4)	OS Credential Dumping (4)	Data from Network Shared Drive (4)	Ingress Tool Transfer (4)	Network Denial of Service (2)	Firmware Corruption (4)
Search Victim-Owned Websites (4)	User Execution (3)	User Execution (3)	Software Deployment Tools (4)	Hijack Execution Flow (13)	Hijack Execution Flow (13)	Hide Artifacts (22)	Steal Application Access Token (4)	Debugger Evasion (4)	Device Driver Discovery (4)	Non-Standard Port (4)	Multi-Stage Channels (4)	Scheduled Transfer (4)	Inhibit System Recovery (4)
	Windows Management Instrumentation (4)	Windows Management Instrumentation (4)	System Services (2)	Implant Internal Image (4)	Process Injection (12)	Hijack Execution Flow (13)	Steal or Forge Kerberos Tickets (4)	File and Directory Discovery (4)	Log Enumeration (4)	Protocol Tunneling (4)	Non-Application Layer Protocol (4)	Transfer Data to Cloud Account (4)	Resource Hijacking (4)
			Power Settings (4)	Modify Authentication Process (4)	Office Application Startup (4)	Scheduled Task/Job (3)	Impair Defenses (11)	Group Policy Discovery (4)	Network Service Discovery (4)	Email Collection (3)	Remote Access Software (4)	System Shutdown/Reboot (4)	Service Stop (4)
			Pre-OS Boot (3)	Valid Accounts (4)	Indicator Removal (4)	Indirect Command Execution (4)	Impersonation (4)	Steal or Forge Session Cookies (4)	Network Share Discovery (4)	Input Capture (4)	Traffic Signaling (2)		
			Scheduled Task/Job (3)		Masking (4)	Masking (4)	Modify Authentication Process (4)	Unsecured Credentials (4)	Password Policy Discovery (4)	Screen Capture (4)	Web Service (3)		
			Server Software Component (3)		Modify Cloud Infrastructure (2)	Modify Cloud Infrastructure (2)	Peripheral Device Discovery (4)			Video Capture (4)			

MITRE ATT&CK Framework

- Le Framework MITRE ATT&CK est une base de connaissances mondiale qui répertorie les tactiques et techniques utilisées par les cybercriminels et les groupes APT (Advanced Persistent Threats). Il a été conçu pour mieux comprendre comment se déroulent les cyberattaques.
- ATT&CK est l'abréviation de : Adversarial Tactics, Techniques and Common Knowledge.
- Ce Framework sert généralement de référence pour mettre en évidence les différentes phases du cycle de vie d'une attaque, des logiciels utilisés et des systèmes d'exploitations visés.
- Il s'agit également d'une ressource précieuse pour les « blue team », car il détaille les différentes TTP utilisées par des attaquants spécifiques et fournit aux entreprises des renseignements précieux sur les cybermenaces (CTI - Cyber threat intelligence) qui peuvent ensuite être utilisés pour mettre en œuvre des défenses et des contre mesures.

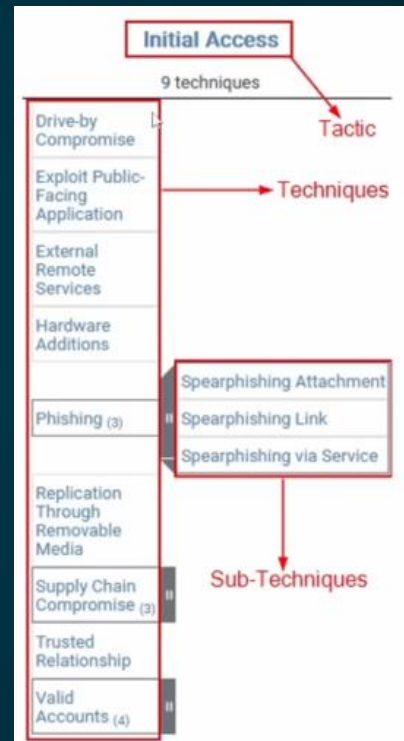
Tactics, techniques & Procedures (TTPs)

- « Tactics » caractérise chaque étape de la méthodologie d'une attaque.
 - Les tactiques représentent le but ou l'objectif d'un adversaire.
- « Techniques » est utilisé pour expliquer comment chaque tactique est orchestrée.
 - Les techniques décrivent les actions réalisées par un adversaire afin d'atteindre son objectif.
 - Les sous-technique expliquent en détail l'implémentation d'une technique spécifique.
- « Procedures » explique toutes les implémentations connues d'une technique ou sous-technique.

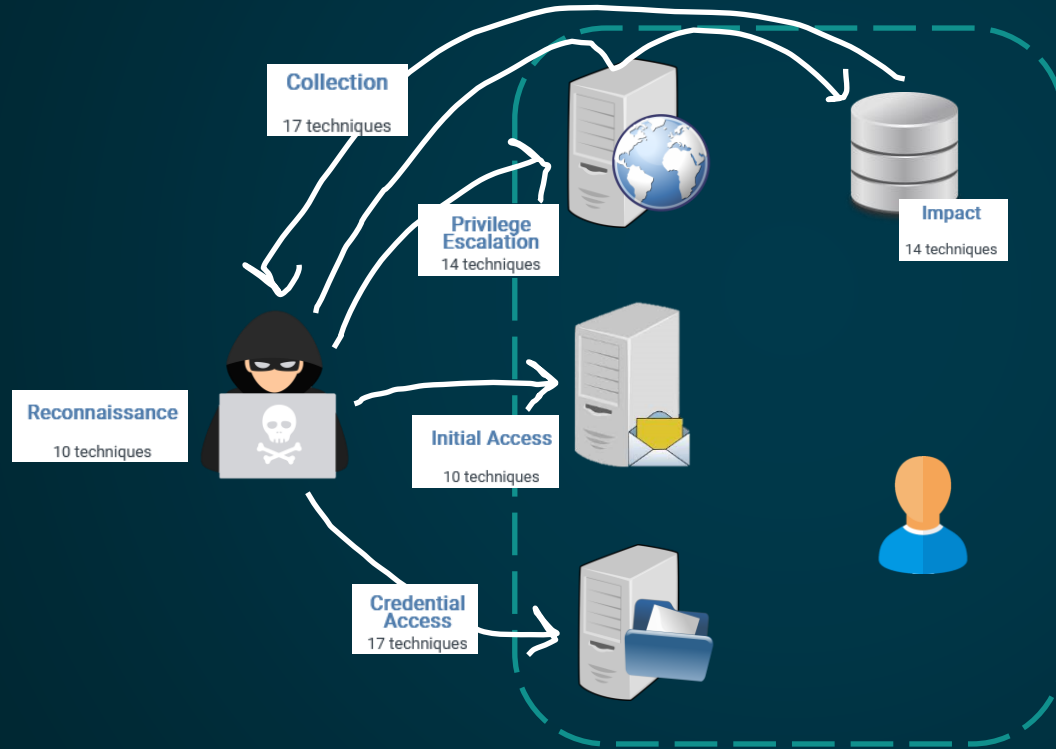
Tactics → Quoi ?

Techniques → Comment ?

Procedures → Détails



Exemple de scénario



On pourrait ainsi caractériser comme ceci le scénario de notre attaque en s'appuyant sur le framework Mitre Att&ck.

Et nous utiliserons les PPTs (People, Process and Technology) pour contrer les TTPs !

TP – Reconnaissance



Passive Reconnaissance


Learn about the essential tools for passive reconnaissance, such as whois, nslookup, and dig.

Easy ⌚ 60 min

Notions visées

- Utilisation de whois / nslookup / dig ;
- Capacité à s'informer sur sa cible.


Pour aller plus loin...



Active Reconnaissance

Learn how to use simple tools such as traceroute, ping, telnet, and a web browser to gather information.

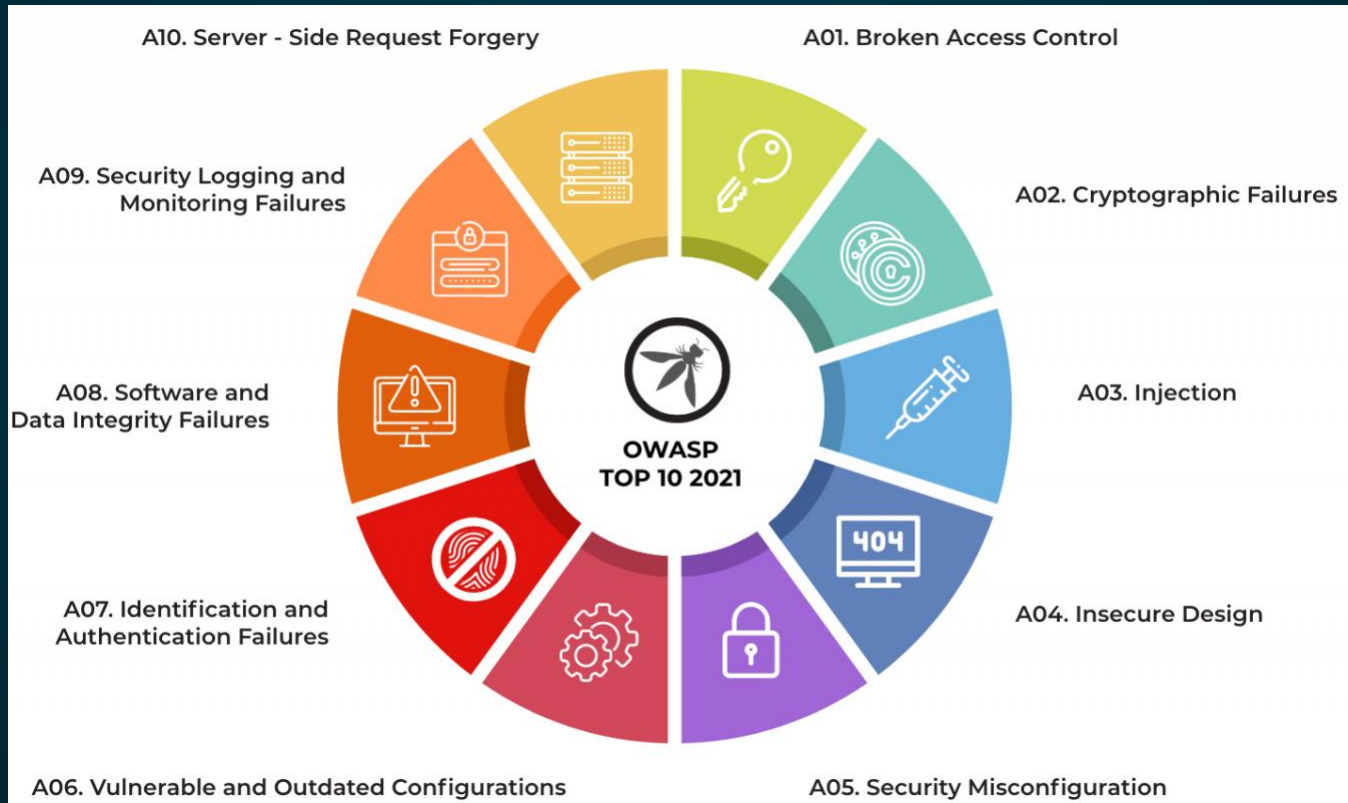
Easy ⌚ 60 min



Basic Pentesting

This is a machine that allows you to practise web app hacking and privilege escalation

Easy ⌚ 0 min



Le Phishing : Comprendre et Prévenir

Définition : Le phishing est une technique de cyberattaque où un attaquant se fait passer pour une entité de confiance afin de tromper les victimes et obtenir des informations sensibles telles que des identifiants, des mots de passe ou des informations bancaires.

Types de Phishing :

- **Phishing par e-mail** : Envoi de courriels frauduleux imitant des entreprises légitimes ;
- **Phishing par SMS (Smishing)** : Envoi de messages texte trompeurs ;
- **Phishing par téléphone (Vishing)** : Appels téléphoniques frauduleux ;
- **Phishing par réseaux sociaux** : Messages directs sur des plateformes sociales.

Méthodes Utilisées :

- **Usurpation d'identité** : Utilisation de logos et de noms d'entreprises pour paraître légitime ;
- **Création de faux sites web** : Imitation de sites web officiels pour voler des informations ;
- **Messages d'urgence** : Incitation à agir rapidement pour éviter des conséquences négatives.

Prévention :

- **Vérification des sources** : Toujours vérifier l'authenticité des messages ;
- **Éducation et sensibilisation** : Former les utilisateurs à reconnaître les signes de phishing ;
- **Utilisation de logiciels de sécurité** : Installer des filtres anti-phishing et des logiciels de sécurité.

Defang d'URL

Définition : Le defang d'URL est une technique utilisée pour rendre les liens web non cliquables et donc inoffensifs, tout en restant lisibles pour les humains.

Objectif : Empêcher les utilisateurs de cliquer accidentellement sur des liens potentiellement malveillants.

Pourquoi utiliser le Defang d'URL ?

- **Sécurité** : Protège contre les attaques de phishing et autres menaces en rendant les liens inactifs ;
- **Analyse** : Permet aux analystes de sécurité de partager des liens suspects sans risquer une exécution accidentelle.

Avantages du Defang d'URL :

- **Prévention** : Réduit les risques d'interaction accidentelle avec des sites malveillants ;
- **Collaboration** : Facilite le partage sécurisé d'indicateurs de compromission (IOC) entre équipes de sécurité.

Defang d'URL - Exemple

Caractères modifiés :

1. Protocole :
 - `http://` devient `hxxp://`
 - `https://` devient `hxxps://`
2. Points dans les noms de domaine :
 - `.` devient `[.]`
3. Séparateurs de protocole :
 - `://` devient `[:]//`

Exemples concrets :

- URL originale : `http://example.com`
 - Defanged : `hxxp://example[.]com`
- URL originale : `https://malicious-site.com/path`
 - Defanged : `hxxps[:]//malicious-site[.]com/path`
- URL originale : `https://ent.univ-amu.fr`
 - Defanged : `hxxps[:]//ent[.]univ-amu[.]fr`

Pourquoi ces changements ?

- **Protocole** : En remplaçant `http` par `hxxp`, on empêche les navigateurs de reconnaître le lien comme une URL valide ;
- **Points** : En remplaçant les points par `[.]`, on évite que les noms de domaine soient interprétés comme des liens cliquables ;
- **Séparateurs de protocole** : En modifiant `://` en `[:]//`, on désactive le lien tout en conservant sa lisibilité pour les humains.

TP – Initial Access - Phishing



Phishing Analysis Fundamentals

Learn all the components that make up an email.

Easy 30 min

Notions visées

- Analyser un email ;
- Comprendre la démarche d'un attaquant ;
- Connaître les principales techniques d'entrée dans un système ;
- Comprendre le but du defang d'URL.

Pour aller plus loin...



Phishing Emails in Action

Learn the different indicators of phishing attempts by examining actual phishing emails.

Easy 30 min

Injectons XSS

Définition : **XSS** est une attaque par injection de code où un attaquant insère du code malveillant dans un site web légitime. Ce code est ensuite exécuté dans le navigateur de l'utilisateur.

Types de XSS :

1. **XSS Reflet (Non-Persistant) :**
 - Le script malveillant est renvoyé par le serveur web et exécuté immédiatement ;
2. **XSS Stocké (Persistant) :**
 - Le script est stocké sur le serveur (ex. base de données) et exécuté lorsque l'utilisateur accède à la page ;
3. **XSS Basé sur le DOM :**
 - Le script modifie le Document Object Model (DOM) de la page web pour exécuter du code malveillant.

Risques :

- Vol de cookies de session ;
- Usurpation d'identité ;
- Accès non autorisé à des informations sensibles.

Prévention :

- **Validation des entrées :** Vérifier et nettoyer toutes les données entrantes (formulaire de contact par exemple) ;
- **Encodage des sorties :** Convertir les données utilisateur en une forme sécurisée avant de les afficher ;
- **Utilisation de Content Security Policy (CSP) :** Limiter les sources de scripts exécutables.

Injectons XSS - Exemple

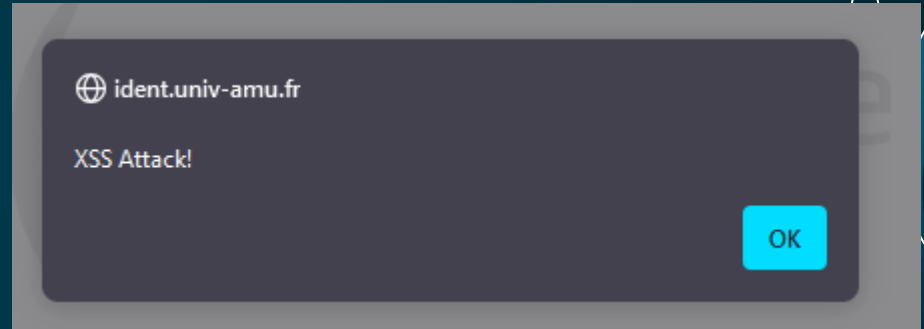
Scénario : Un site web possède un formulaire de recherche qui affiche les résultats de la recherche directement sur la page.

Étape 1 : Injection du Script Malveillant L'attaquant entre le code suivant dans le champ de recherche :

```
<script>alert('XSS Attack!');</script>
```

Étape 2 : Exécution du Script Le site web renvoie la recherche et affiche le contenu sans le filtrer :

Résultat : Le navigateur de l'utilisateur exécute le script et affiche une alerte avec le message "XSS Attack!".




Impact :

- L'attaquant peut exécuter des scripts malveillants dans le navigateur de l'utilisateur ;
- Peut conduire au vol de cookies, à la redirection vers des sites malveillants, ou à l'exécution de commandes non autorisées.

Prévention :

- **Validation des entrées** : Filtrer et échapper les caractères spéciaux ;
- **Encodage des sorties** : Utiliser des fonctions d'encodage pour afficher les données utilisateur de manière sécurisée.

TP – Initial Access - XSS



XSS

Explore in-depth the different types of XSS and their root causes.

Easy ⌚ 120 min

Notions visées

- Comprendre les risques de l'injection de code
- Comprendre les contre-mesures

Pour aller plus loin...

<https://tryhackme.com/r/room/dombasedattacks>

Injectons SQL

Définition : **SQL Injection (SQLi)** est une vulnérabilité de sécurité web qui permet à un attaquant d'interférer avec les requêtes qu'une application fait à sa base de données.

Types d'Injections SQL :

1. **Injectons SQL Basées sur les Erreurs :**
 - L'attaquant obtient des informations sur la base de données en provoquant des erreurs.
2. **Injectons SQL Basées sur l'Union :**
 - L'attaquant récupère des données supplémentaires en combinant des requêtes.
3. **Injectons SQL Aveugles :**
 - L'attaquant envoie des requêtes qui ne renvoient pas directement de données mais permettent de déduire des informations.

Risques :

- Accès non autorisé à des données sensibles ;
- Modification ou suppression de données ;
- Compromission du serveur ou de l'infrastructure backend.

Prévention :

- **Validation des entrées :** Vérifier et nettoyer toutes les données entrantes ;
- **Utilisation de requêtes préparées :** Utiliser des requêtes avec des paramètres pour éviter l'injection de code ;
- **Utilisation d'ORM (Object-Relational Mapping) :** Utiliser des outils qui gèrent les requêtes SQL de manière sécurisée (ex: PDO en PHP).

Injection SQL - Exemple

Scénario : Un site web possède un formulaire de connexion qui vérifie les identifiants des utilisateurs.

Étape 1 : Injection du Code Malveillant L'attaquant entre le code suivant dans le champ "Nom d'utilisateur" :

```
' OR '1'='1'
```

Étape 2 : Exécution de la Requête Le site web exécute la requête SQL suivante :

```
SELECT * FROM utilisateurs WHERE nom_utilisateur = '' OR '1'='1' AND mot_de_passe = '';
```

Résultat : La condition `'1'='1'` est toujours vraie, donc la requête renvoie tous les utilisateurs de la base de données, permettant à l'attaquant de se connecter sans connaître le mot de passe.


Impact :

- Accès non autorisé à des comptes utilisateurs ;
- Possibilité de voler des informations sensibles ou de modifier des données.

Prévention :

- **Validation des entrées** : Filtrer et échapper les caractères spéciaux ;
- **Utilisation de requêtes préparées** : Utiliser des requêtes avec des paramètres pour éviter l'injection de code.

TP – Initial Access - SQLi



SQL Injection


Learn how to detect and exploit SQL Injection vulnerabilities

📶 Medium ⌚ 30 min

Notions visées

- Comprendre le principe d'une injection SQL ;
- Comprendre le mécanisme d'injection ;
- Connaître les contre-mesures.


Pour aller plus loin...



Advanced SQL Injection

Learn advanced injection techniques to exploit a web app.

📶 Medium ⌚ 60 min



SQL Injection Lab

Understand how SQL injection attacks work and how to exploit this vulnerability.

📶 Easy ⌚ 0 min

Gestion des Identités et des Accès (IAM)

La Identity Access Management (IAM) est un cadre qui permet à l'équipe informatique de contrôler l'accès aux systèmes, aux réseaux et aux ressources en fonction de l'identité de chaque utilisateur.

Elle est constituée de deux composants principaux :

1. **Gestion des identités** : vérifie l'identité de l'utilisateur sur la base des informations stockées dans une base de données de gestion des identités.
2. **Gestion des accès** : utilise l'identité du demandeur pour confirmer ses droits d'accès à différents systèmes, applications, données, terminaux et autres ressources.

Les principales fonctions d'une solution IAM sont les suivantes :

- **Identification** : *Attribuer une identité numérique unique à chaque utilisateur ;*
- **Authentification** : *Authentifier l'utilisateur (vérifier son identité) ;*
- **Authorisation** : *Autoriser un accès approprié aux ressources pertinentes ;*
- **Accountability** : *Surveiller et gérer les identités afin de les aligner sur les changements survenant dans l'entreprise.*

Gestion des accès - Authentification multifacteur (MFA)

L'authentification multifacteur (MFA) est une fonction de sécurité qui n'accorde l'accès à un utilisateur qu'après avoir vérifié son identité au moyen d'un ou plusieurs identifiants, en plus de son nom d'utilisateur et de son mot de passe. Il peut s'agir d'un code de sécurité envoyé par SMS ou par email, d'un jeton de sécurité fourni par une application d'authentification, ou encore d'un identifiant biométrique.

Le principe est de fournir quelque chose que l'on connaît (ex: un mot de passe), quelque chose que l'on possède (ex: un téléphone/carte à puce), quelque chose que l'on est (ex: empreinte digitale).

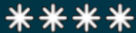



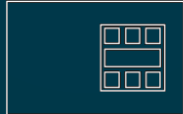

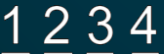
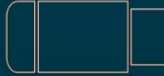

De nos jours, les MFA par le biais d'un email ou d'un SMS sont à éviter. La fiabilité d'une sécurité par SMS peut être remise en question si l'attaquant peut se fournir un double de la carte SIM de la victime.

Les tokens OTP (One Time Password / Mot de passe à usage unique) sont à privilégier.



Notions à approfondir à la maison : *Zero Trust, Principe du moindre privilège (Least Privilege), Gestion des accès privilégiés (PAM).*

Gestion des accès – MFA

Knowledge Factor (something you know)	Possession Factor (something you have)	Inherence Factor (something you are)
 Password	 Smartphone	 Fingerprint
 Security Question	 Smart Card	 Retina Pattern
 PIN	 Hardware Token	 Face Recognition

Gestion des accès - OTP



Le mot de passe à usage unique (ou OTP en abrégé) est un code unique que vous ne pouvez utiliser qu'une seule fois. L'OTP est généralement un code à 6 chiffres que l'utilisateur doit saisir pour se connecter à son application lors de l'authentification à deux facteurs (2FA) ou de l'authentification à plusieurs facteurs (MFA).

L'OTP peut être basé sur un compteur d'événements (HOTP) ou un compteur de temps (TOTP).

Les **soft tokens** sont des applications que vous pouvez installer sur votre ordinateur ou votre téléphone pour générer des codes OTP.

→ Exemple : Aegis Authenticator, Google Authenticator, Duo Mobile, etc.

Les **hard tokens** sont des porte-clés physiques dotés d'un écran minuscule qui génèrent des jetons OTP ou des équipements physique comme une clef USB.

→ Exemple : Yubikey, Onlykey, Nitrokey, Carte à puce, etc.

Gestion des accès - HOTP vs TOTP : lequel est le plus sûr ?

HMAC-Based One-Time Password (HOTP)	Time-Based One-Time Password (TOTP)
Compteur d'évènements	Compteur de temps
Le compteur s'incrmente après une authentification effectuée ou un appuie sur le bouton	Le compteur s'incrmente toutes les 30 secondes
Le code OTP est valide pendant une durée indéterminée (tant qu'un nouveau code n'est pas généré)	Le code OTP est valide pendant 30 secondes uniquement
Nécessite une fenêtre de validation (doit enregistrer des codes avant et après celui attendu dans le doute où il y aurait une désynchronisation)	Ne nécessite pas une fenêtre de validation

Gestion des accès - HOTP vs TOTP : lequel est le plus sûr ?

→ Lequel choisir ?

Un seul code TOTP est valable à la fois, ce qui rend le TOTP moins piratable que le HOTP.
Les codes TOTP changent toutes les 30 secondes, ce qui rend le TOTP plus sûr que le HOTP.

En définitive, la question HOTP vs TOTP a une réponse claire : **le TOTP est beaucoup plus sûr que le HOTP** parce qu'il utilise l'algorithme HOTP sous-jacent tout en introduisant des modifications qui améliorent la sécurité.

Il n'y a aucune raison d'utiliser HOTP au lieu de TOTP. La seule exception concerne les anciens systèmes qui ne prennent pas en charge l'heure Unix.

Gestion des accès – SSO (Sigle Sign-On)

Définition : L'authentification unique (SSO) est une méthode d'authentification qui permet à un utilisateur d'accéder à plusieurs applications avec un seul jeu de identifiants..

Objectif : Simplifier la gestion des identifiants et améliorer la sécurité en réduisant le nombre de mots de passe à mémoriser.

Principe :

- L'authentification SSO repose sur une relation de confiance entre un fournisseur d'identité et une application ;
- La relation de confiance s'appuie souvent sur l'échange d'un certificat qui va permettre de signer les informations envoyées afin que les deux parties sachent qu'elles proviennent d'une source approuvée ;
- Ces informations se présentent sous la forme de jetons contenant les informations d'identification de l'utilisateur, comme un e-mail ou un nom d'utilisateur.

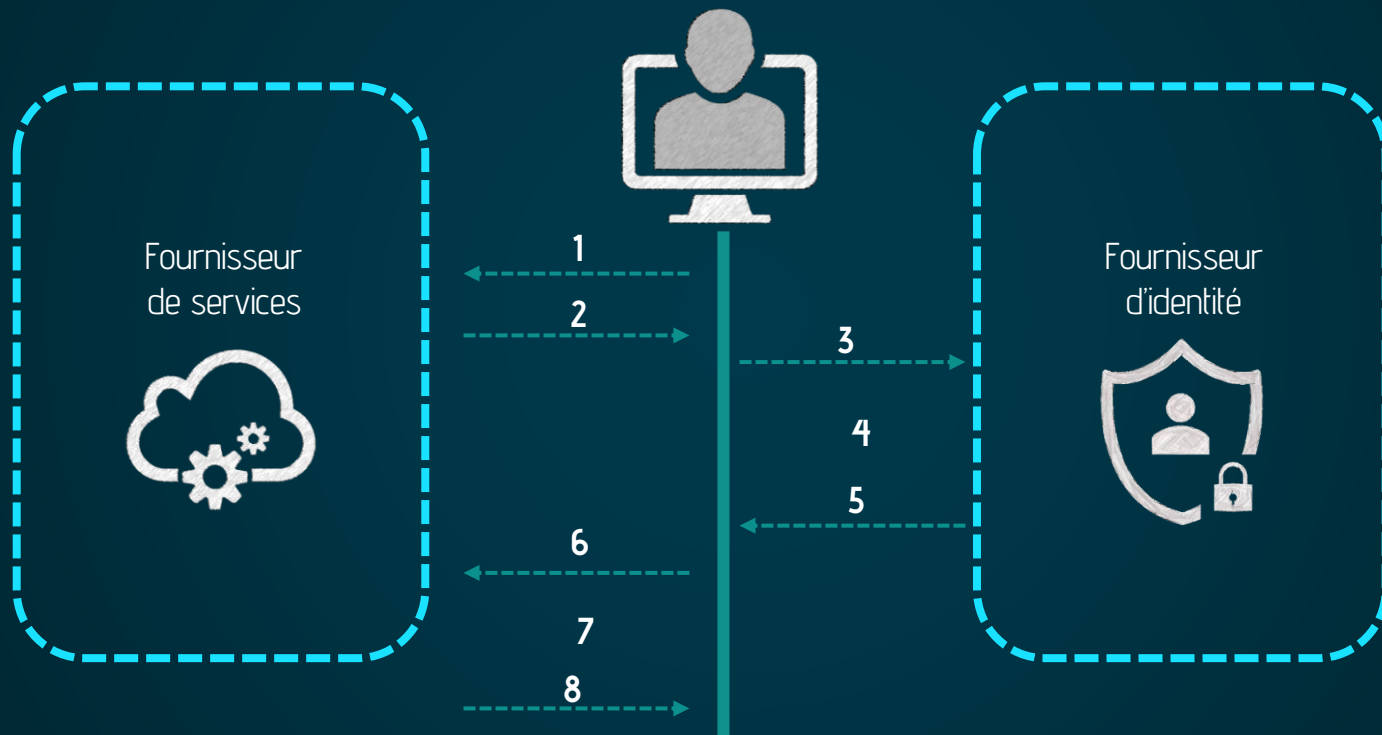


Gestion des accès – SSO (Sigle Sign-On)

Le flux de connexion se déroule généralement comme suit :

1. L'utilisateur se rend sur l'application ou le site Web auquel il souhaite accéder : le fournisseur de services.
2. Le fournisseur de services envoie au système SSO (fournisseur d'identité) un jeton contenant des informations sur l'utilisateur, comme son e-mail, dans le cadre de la requête d'authentification de cet utilisateur.
3. Le fournisseur d'identité vérifie d'abord si l'utilisateur est déjà authentifié, auquel cas il lui accorde l'accès à l'application du fournisseur de services, puis il passe à l'étape 5.
4. Si l'utilisateur n'est pas déjà connecté, il est invité à le faire en fournissant les informations d'identification requises par le fournisseur d'identité. Il peut simplement s'agir d'un nom d'utilisateur et d'un mot de passe, ou cela peut inclure une autre forme d'authentification comme un OTP (ou autre MFA).
5. Une fois que le fournisseur d'identité valide les informations d'identification fournies, il renvoie un jeton au fournisseur de services pour confirmer l'authentification.
6. Le fournisseur de services reçoit le jeton par l'intermédiaire du navigateur de l'utilisateur.
7. Le jeton reçu par le fournisseur de services est validé d'après la relation de confiance établie entre le fournisseur de services et le fournisseur d'identité au moment de la configuration initiale.
8. L'accès au fournisseur de services est accordé à l'utilisateur.

Gestion des accès – SSO (Sigle Sign-On)



Gestion des accès – SSO (Sigle Sign-On)

Avantages du SSO :

- **Sécurité améliorée** : Moins de mots de passe à gérer, réduisant le risque de mots de passe faibles ou réutilisés ;
- **Expérience utilisateur** : Connexion simplifiée et plus rapide aux applications ;
- **Gestion centralisée** : Facilite la gestion des accès et des permissions.

Défis et Considérations :

- **Point de défaillance unique** : Si le SSO est compromis, toutes les applications connectées le sont aussi ;
- **Complexité de mise en œuvre** : Nécessite une intégration et une configuration précises.

Cas d'utilisation :

- **Entreprises** : Accès aux applications internes (email, CRM, ERP) ;
- **Éducation** : Accès aux plateformes d'apprentissage et aux ressources en ligne (ex. ident.univ-amu.fr).

Protocoles courants :

- SAML (Security Assertion Markup Language) ;
- OAuth ;
- OpenID Connect ;
- CAS (Central Authentication Service).

Techniques de Contrôle d'Accès : DAC, RBAC et MAC

Discretionary Access Control (DAC)

Définition : Le contrôle d'accès discrétionnaire permet aux propriétaires de ressources de décider qui peut accéder à leurs ressources et quelles actions ils peuvent effectuer.

Caractéristiques :

- **Flexibilité :** Les propriétaires peuvent accorder ou révoquer des permissions à leur discrétion ;
- **Utilisation courante :** Systèmes d'exploitation comme Windows et Unix, ou un Cloud (ex. partage de photo).

Avantages :

- Facile à mettre en œuvre ;
- Grande flexibilité pour les utilisateurs.

Inconvénients :

- Moins sécurisé, car les utilisateurs peuvent partager des accès sans contrôle centralisé.

Techniques de Contrôle d'Accès : DAC, RBAC et MAC

Role-Based Access Control (RBAC)

Définition : Le contrôle d'accès basé sur les rôles attribue des permissions à des rôles spécifiques plutôt qu'à des utilisateurs individuels.

Caractéristiques :

- **Gestion centralisée :** Les administrateurs définissent des rôles et attribuent des utilisateurs à ces rôles ;
- **Scalabilité :** Idéal pour les grandes organisations avec de nombreux utilisateurs.

Avantages :

- Simplifie la gestion des permissions ;
- Réduit les erreurs humaines.

Inconvénients :

- Peut être complexe à configurer initialement.

Techniques de Contrôle d'Accès : DAC, RBAC et MAC

Mandatory Access Control (MAC)

Définition : Le contrôle d'accès obligatoire utilise des politiques de sécurité centralisées pour contrôler l'accès aux ressources.

Caractéristiques :

- **Hierarchique** : Les accès sont déterminés par des niveaux de classification et des autorisations ;
- **Utilisation courante** : Militaire, gouvernement et autres environnements hautement sécurisés.

Avantages :

- Très sécurisé, car les utilisateurs ne peuvent pas modifier les permissions ;
- Contrôle strict des accès.

Inconvénients :

- Moins flexible.

TP – Initial Access - IAM



Identity and Access Management

Learn about identification, authentication, authorisation, accounting, and identity management.

Easy ⌚ 120 min

Notions visées

- Différence entre Authentification et Identification ;
- Single Sign-On principe ;
- MFA ;
- Access Control Models.

Pour aller plus loin...



Introduction to CryptOps

Key management strategies for DevSecOps.

Easy ⌚ 60 min



Introduction to Cryptography

Learn about encryption algorithms such as AES, Diffie-Hellman key exchange, hashing, PKI, and TLS.

Medium ⌚ 240 min