



# Sécurité X Informatique

**Responsable de cours :**  
Ala Eddine KHARRAT  
[alaeddinekharrat@gmail.com](mailto:alaeddinekharrat@gmail.com)

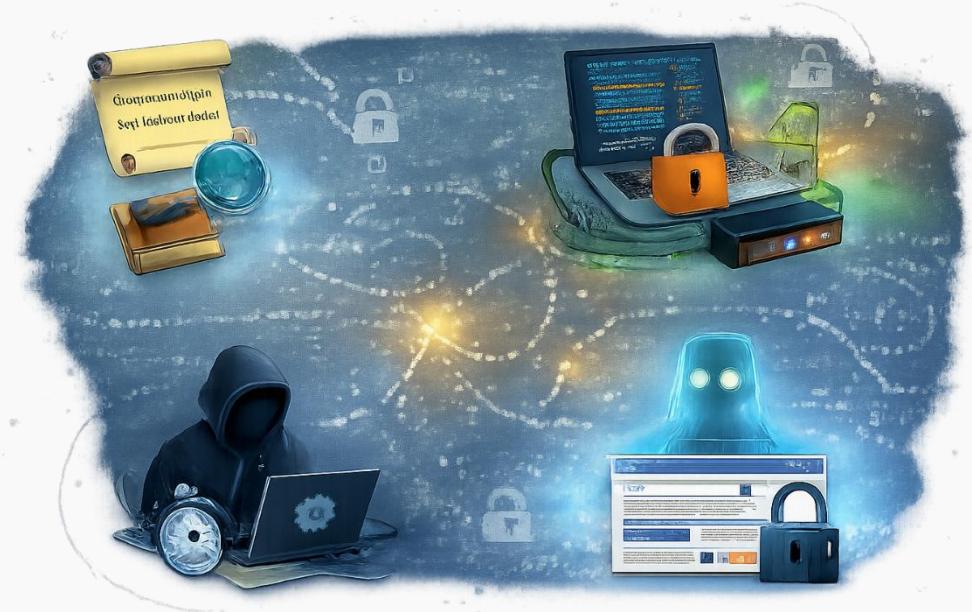
3LI - ESSTHS | 2025/2026

# Informations Générales sur les Projets (1)

## ➤ Types de Projets :

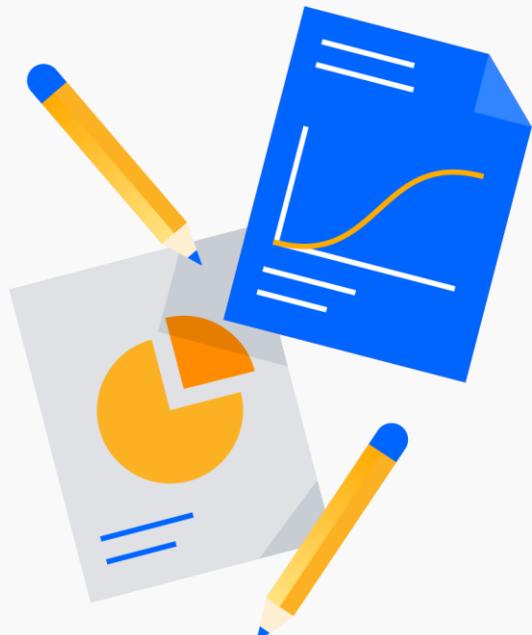
- Les projets sont divisés en quatre catégories. Chaque étudiant devra réaliser un seul projet.

- **Catégorie 1 : Breaking the Code**
- **Catégorie 2 : Inside the Wire**
- **Catégorie 3 : Silent Execution**
- **Catégorie 4 : Ghost in the Browser**



# Informations Générales sur les Projets (2)

- **Deadline : 03/01/2026 à 23h59**
- **Date stricte et non négociable** : Aucun projet soumis après cette date ne sera accepté, quelles que soient les raisons.
- **Travail à fournir** :
  1. **Code** : Bien structuré, avec des commentaires clairs.
  2. **Vidéo explicative** (entre 3 et 6 minutes maximum) :
    - Présentation du projet.
    - Démonstration du fonctionnement.
    - Explication détaillée du code.

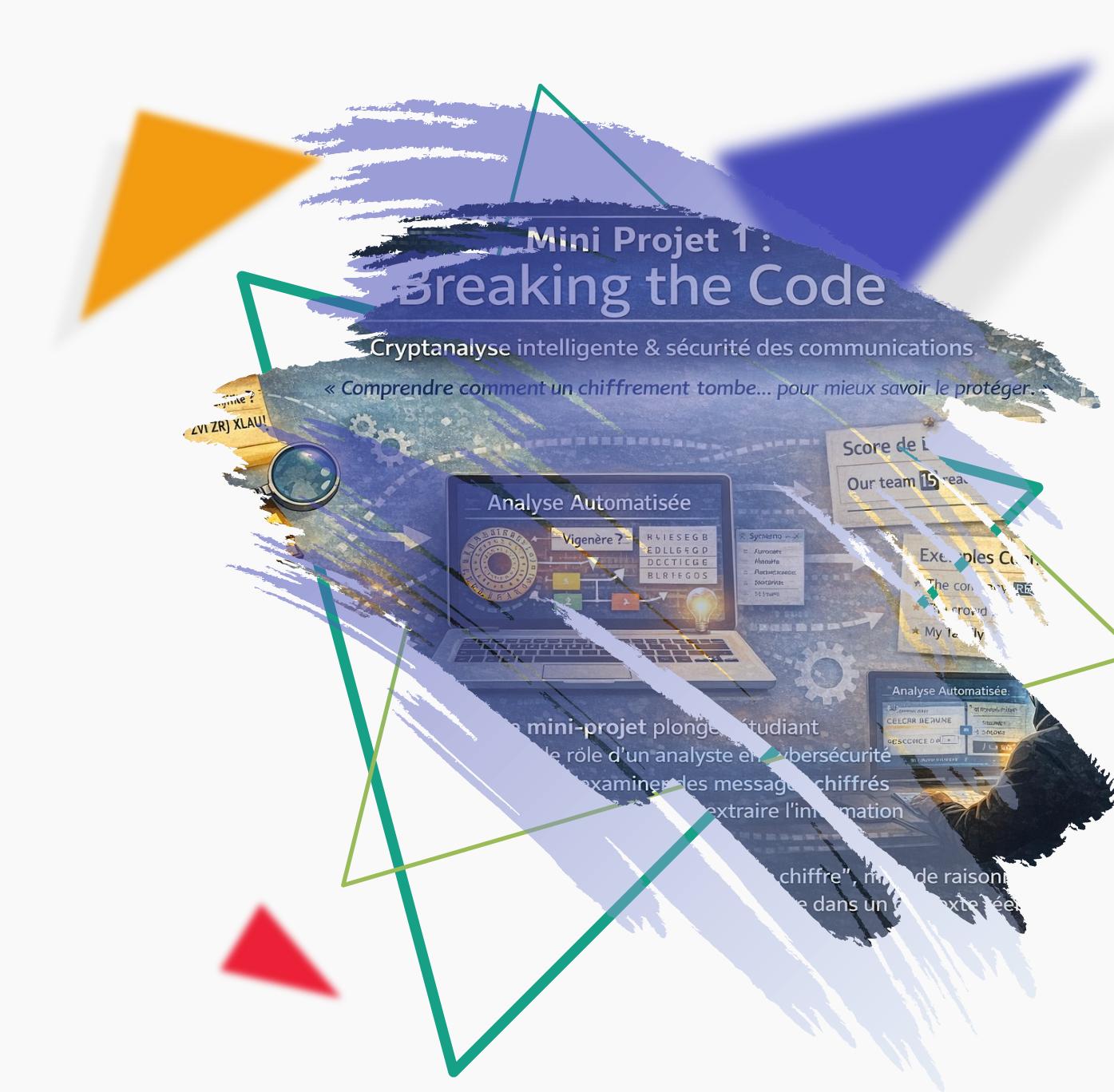


# Mini Projet 1 :

---

## ***Breaking the Code : Cryptanalyse intelligente & sécurité des communications***

- **P1-C1** : Cryptanalyse intelligente automatique
- **P1-C2** : Cryptanalyse à partir de communications interceptées



# Mini Projet 1 : *Breaking the Code*

## Concept général :

### Cryptanalyse intelligente & sécurité des communications

« Comprendre comment un chiffrement tombe... pour mieux savoir le protéger. »

- Ce mini-projet plonge l'étudiant dans le rôle d'un **analyste en cybersécurité** chargé d'**examiner des messages chiffrés** ou interceptés et d'en **extraire l'information** sans connaître la clé.
- L'objectif n'est pas seulement de "**casser un chiffre**", mais de **raisonner, tester, évaluer et automatiser** l'analyse, comme dans un contexte réel d'audit ou d'investigation numérique.

# P1-C1 : Cryptanalyse intelligente automatique

## Objectif général :

- Développer un outil de **cryptanalyse** intelligent capable d'**analyser** un message chiffré sans information préalable sur la clé, et de **proposer** automatiquement le meilleur texte clair possible.

## Description :

- Dans cette option, l'étudiant doit concevoir un programme qui imite le raisonnement d'un analyste sécurité face à un message chiffré inconnu.
- Le système devra :
  - ✓ Tester plusieurs hypothèses de déchiffrement
  - ✓ Évaluer la qualité linguistique des résultats obtenus
  - ✓ Sélectionner automatiquement la solution la plus crédible



# P1-C1 : Cryptanalyse intelligente automatique

- Le projet s'appuie sur les chiffrements étudiés en TP (mono-alphabétique et/ou poly-alphabétique), mais laisse une grande liberté sur :
  - ✓ la méthode d'analyse,
  - ✓ les critères de validation,
  - ✓ la façon de classer les résultats.
- L'accent est mis sur :
  - ✓ l'automatisation de l'analyse,
  - ✓ la logique de décision,
  - ✓ la lisibilité du résultat final pour un humain.
- L'étudiant n'est pas jugé sur la “force brute” seule, mais sur **sa capacité à faire parler les données**.
- **Toutes les expérimentations doivent rester dans un cadre local et pédagogique.**

# P1-C2 : Cryptanalyse à partir de communications interceptées

## Objectif général :

- **Analyser** et **casser** un message chiffré issu d'une communication réseau, comme dans un scénario réel d'**écoute ou d'investigation**.

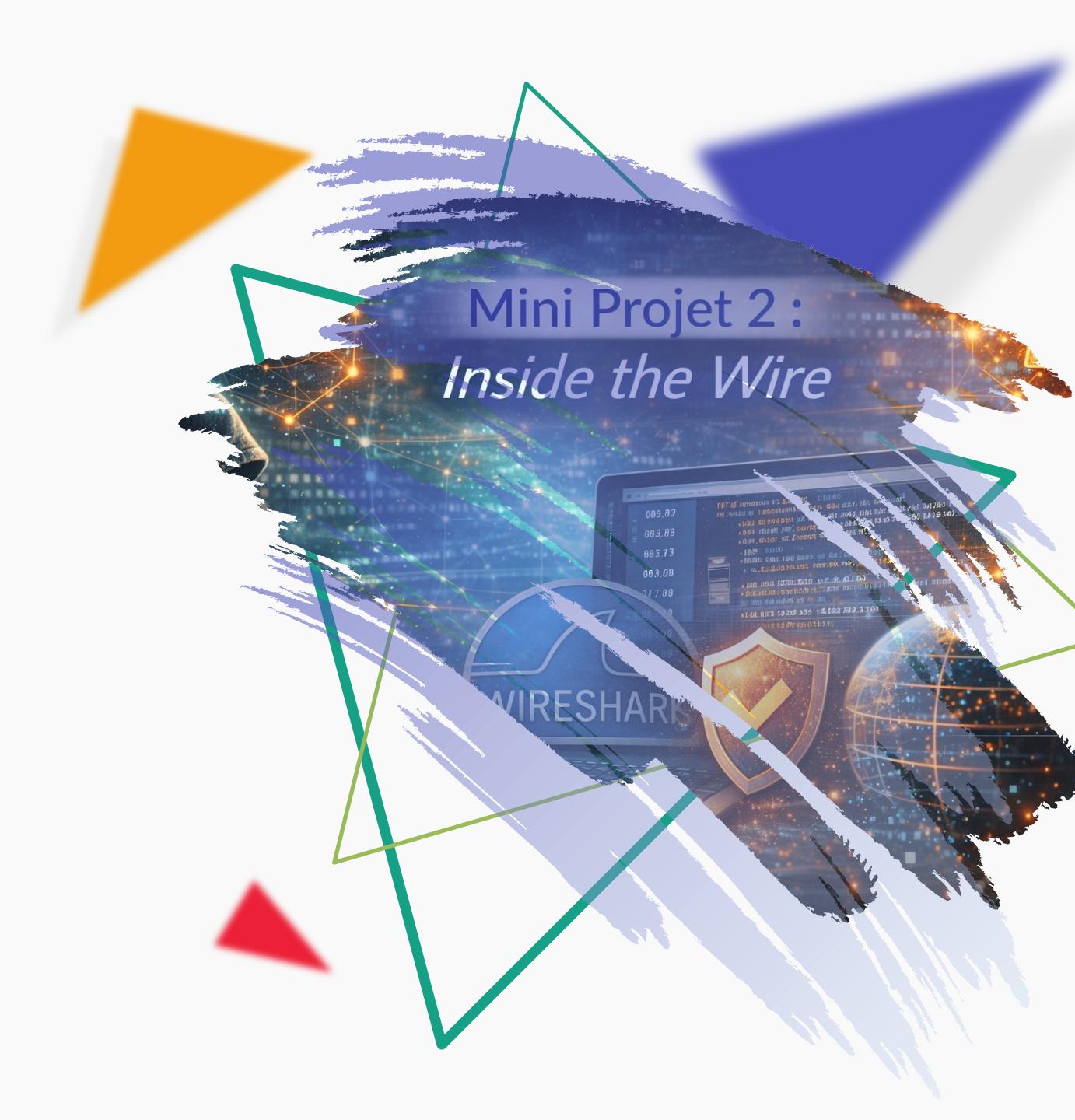
## Description :

- Dans cette option, le message chiffré n'est plus donné directement sous forme de fichier, mais transite via une communication réseau (client/serveur, socket, échange simulé, etc.).
- **L'étudiant devra :**
  - ✓ Récupérer ou observer un message chiffré circulant sur le réseau
  - ✓ Identifier le type de chiffrement utilisé
  - ✓ Appliquer une méthode de cryptanalyse pour retrouver le contenu original



# P1-C2 : Cryptanalyse à partir de communications interceptées

- **Le projet met en évidence le fait que :**
  - ✓ un chiffrement faible reste vulnérable même sur un réseau fonctionnel,
  - ✓ la sécurité ne se limite pas au transport, mais aussi à l'algorithme utilisé.
- **Cette option encourage une vision globale :**
  - ✓ réseau + chiffrement + analyse
  - ✓ attaque passive, sans modification du système cible
- **⚠️ Toutes les expérimentations doivent rester dans un cadre local et pédagogique.**



# Mini Projet 2 :

---

## *Inside the Wire :*

### *Attaques et défense des communications TCP*

- **P2-C1 :** Détection d'une attaque Man-in-the-Middle
- **P2-C2 :** Élaboration d'une suite de tests d'attaques pour sockets TCP

# Mini Projet 2 : *Inside the Wire*

## Concept général :

### Attaques et défense des communications TCP

« Ce qui transite sur le réseau peut être observé, modifié... ou protégé. »

- Ce mini-projet place l'étudiant au cœur des **communications réseau TCP**, en lui faisant explorer les attaques classiques, les failles de conception, et les mécanismes de défense utilisés dans les systèmes réels.
- L'objectif est de comprendre concrètement **comment une communication peut être compromise**, puis **comment elle peut être durcie face à des attaquants actifs ou passifs**.

# P2-C1 : Détection d'une attaque Man-in-the-Middle

## Objectif général :

- Développer un système de communication TCP capable de **déetecter automatiquement** la présence d'une attaque **Man-in-the-Middle**, à partir de l'**analyse des messages échangés** entre **un client et un serveur**.

## Description :

- Dans cette option, l'étudiant met en place une communication TCP normale (client/serveur), puis conçoit des mécanismes de détection permettant d'identifier qu'un tiers malveillant est susceptible d'intercepter, modifier ou rejouer les messages échangés.
- **L'étudiant se place du côté du défenseur :**
  - ✓ il ne réalise pas l'**attaque MITM**, mais détecte ses effets à travers des incohérences ou anomalies dans les échanges.



# P2-C1 : Détection d'une attaque Man-in-the-Middle

- **Le système doit permettre :**
  - ✓ d'identifier des messages modifiés ou altérés,
  - ✓ de détecter des messages rejoués ou dupliqués,
  - ✓ de repérer des incohérences dans l'ordre ou le contenu des messages,
  - ✓ de déclencher une alerte de sécurité en cas de comportement suspect.
- **Le projet met en évidence :**
  - ✓ qu'une communication TCP peut fonctionner tout en étant vulnérable,
  - ✓ qu'une attaque MITM peut exister sans être directement visible,
  - ✓ que la sécurité repose aussi sur la logique de détection et l'analyse du comportement des échanges.
- **L'objectif n'est pas de bloquer toutes les attaques, mais de savoir reconnaître qu'une communication n'est plus fiable.**
- **Toutes les expérimentations doivent rester dans un cadre local et pédagogique.**

# P2-C2 : *Élaboration d'une suite de tests d'attaques pour sockets TCP*

## Objectif général :

- Concevoir une suite complète de tests d'attaques réseau permettant **d'évaluer** la vulnérabilité et la robustesse d'un **service TCP** basé sur des sockets.

## Description :

- Dans cette option, l'étudiant se place dans le rôle **d'un auditeur sécurité** chargé **d'analyser un service TCP sans en connaître les mécanismes internes.**
- Le travail consiste à **identifier** les points faibles potentiels d'une communication par sockets TCP, puis à **concevoir** et **organiser** une série de tests d'attaques permettant de révéler ces faiblesses.
- L'étudiant ne reçoit aucune liste prédéfinie d'attaques : il doit raisonner, s'appuyer sur les concepts vus en cours et en TP, et proposer ses propres scénarios de tests.



## P2-C2 : *Élaboration d'une suite de tests d'attaques pour sockets TCP*

- **Le système (ou framework) doit permettre**
  - ✓ de définir et structurer des tests d'attaques réseau,
  - ✓ d'exécuter ces tests sur un service TCP cible,
  - ✓ d'observer et analyser les réactions du serveur,
  - ✓ de documenter les comportements normaux et anormaux.
- **Le projet met en évidence :**
  - ✓ les limites d'un service TCP face à des comportements hostiles,
  - ✓ l'importance des tests de sécurité dans le cycle de développement,
  - ✓ la différence entre un service fonctionnel et un service audité.
- **La qualité de la réflexion, la pertinence des scénarios proposés et la clarté de l'analyse sont plus importantes que le nombre de tests réalisés.**
- **Toutes les expérimentations doivent rester dans un cadre local et pédagogique.**

# Mini Projet 3 :

---

## *Silent Execution : Simulation d'un malware éducatif*

- **P3-C1** : Simulation et analyse des logiciels malveillants



# Mini Projet 3 - *Silent Execution*

## Concept général :

### Ingénierie, propagation et défense des logiciels malveillants (simulation)

« Un malware ne commence jamais par attaquer.  
Il commence par se faire passer pour quelque chose d'inoffensif. »

## Objectif général :

- Concevoir une application légitime simulée (ex. calculatrice, outil système, utilitaire simple) qui reproduit le comportement d'un malware moderne, dans un environnement contrôlé, afin d'analyser :
  - ✓ la persistance,
  - ✓ la propagation,
  - ✓ et l'impact sur le système.

# P3-C1 : Simulation et analyse des logiciels malveillants

## Description :

- Dans ce mini-projet, l'étudiant **développe** une application qui semble inoffensive à l'exécution, mais qui simule des comportements typiques d'un malware, tels qu'observés dans le TP.
- L'objectif n'est pas de nuire, mais de comprendre comment un malware fonctionne de l'intérieur, et pourquoi il est difficile à détecter.
- Le projet doit montrer une séparation claire entre :
  - ✓ **l'apparence légitime,**
  - ✓ **et le comportement caché simulé.**



# P3-C1 : Simulation et analyse des logiciels malveillants

- **Le système doit permettre**

- ✓ de simuler une persistance au démarrage (copie, auto-lancement simulé),
- ✓ de simuler une duplication du programme dans des emplacements stratégiques,
- ✓ de simuler une analyse des fichiers utilisateur (ex. bureau, documents),
- ✓ de simuler un comportement de type ransomware :
  - verrouillage logique,
  - renommage fictif,
  - ou journalisation des fichiers ciblés,
- ✓ de produire des logs détaillés de toutes les actions.



# P3-C1 : *Simulation et analyse des logiciels malveillants*

- **Le projet doit illustrer le principe de propagation, par exemple :**
  - ✓ copie simulée vers d'autres emplacements,
  - ✓ propagation logique (scénarios, états, graphes),
  - ✓ démonstration théorique ou locale du mécanisme.
- **Le projet met en évidence :**
  - ✓ comment un malware se dissimule derrière une application légitime,
  - ✓ les mécanismes classiques de persistance,
  - ✓ les impacts potentiels sur un poste utilisateur,
  - ✓ la difficulté de détection sans analyse comportementale.



**Toutes les expérimentations doivent rester dans un cadre local et pédagogique.**



## Mini Projet 4 :

### *Ghost in the Browser*

# Mini Projet 4 :

---

## *Ghost in the Browser :*

### *Automatisation, usurpation et défense des interactions Web*

- **P4-C1** : Automatisation après authentification
- **P4-C2** : Détection et blocage d'un scraper

# Mini Projet 4 : *Ghost in the Browser*

## Concept général :

### Automatisation, usurpation et défense des interactions Web

« Ce qui s'exécute dans le navigateur peut agir à votre place... ou contre vous. »

- Ce mini-projet plonge l'étudiant au cœur du navigateur Web, devenu aujourd'hui un point central des interactions numériques : **authentification, messagerie, réseaux sociaux, services en ligne, applications Web dynamiques**, etc.
- L'objectif est d'**explorer le rôle du navigateur comme intermédiaire**, capable :
  - ✓ d'agir légitimement au nom de l'utilisateur,
  - ✓ mais aussi d'être instrumentalisé par des scripts automatisés (scrapers, bots, outils d'usurpation).

# M4-C1 : Automatisation après authentification

## Objectif général :

- Développer un outil de scraping et d'automatisation capable de se **connecter** à un site Web via une authentification, puis **d'effectuer** une action précise à la place de l'utilisateur, comme l'envoi d'un message ou la publication d'un contenu.

## Description :

- Dans cette option, l'étudiant doit **concevoir** un programme capable de piloter un navigateur Web afin **d'imiter** le comportement d'un utilisateur réel après authentification, en accédant à une fonctionnalité interne du site et en réalisant automatiquement une action précise (envoi de message, publication, soumission de formulaire).



# M4-C1 : Automatisation après authentification

- Le système devra :
  - ✓ **piloter** un navigateur Web (headless ou avec interface graphique) pour simuler une navigation humaine,
  - ✓ **se connecter** à un site Web nécessitant une authentification (login, mot de passe, session, cookies),
  - ✓ **maintenir** et gérer correctement l'état connecté de l'utilisateur,
  - ✓ **accéder** à une fonctionnalité interne disponible uniquement après authentification(messagerie, publication, formulaire, interface utilisateur),
  - ✓ **déclencher** automatiquement une action unique ciblée, telle que :
    - l'envoi d'un message,
    - la publication d'un contenu,
    - ou la soumission d'un formulaire,
  - ✓ **reproduire** une séquence d'actions cohérente, fidèle au comportement attendu d'un utilisateur légitime.

# M4-C1 : Automatisation après authentification

- Le projet met en évidence :

- ✓ le rôle du navigateur comme relais d'identité et de droits de l'utilisateur,
- ✓ le fait qu'une fois authentifié, un programme peut agir exactement dans le périmètre fonctionnel de l'utilisateur,
- ✓ la notion de “**Ghost in the Browser**”, où aucune faille n'est exploitée, mais où le navigateur devient un intermédiaire d'actions automatisées,
- ✓ les risques associés à ces techniques dans des contextes réels : automatisation abusive, spam, manipulation de services Web.

# M4-C2 : Détection et blocage d'un scraper

## Objectif général :

- Concevoir une application Web dynamique intégrant des mécanismes de **détection** et de **protection** contre le scraping automatisé, tout en restant pleinement fonctionnelle pour un utilisateur humain légitime.

## Description :

- Dans cette option, l'étudiant doit **concevoir** une application Web dynamique complète, puis se placer du côté du défenseur afin d'**identifier**, **détecter** et **bloquer** un scraper automatisé tentant d'accéder ou d'extraire des informations depuis le site.



# M4-C2 : Détection et blocage d'un scraper

- Le système devra :
  - ✓ proposer une application Web fonctionnelle comprenant :
    - un menu de navigation,
    - Au moins 3 pages dynamiques,
    - et une notion de session ou de login utilisateur,
  - ✓ exposer des informations accessibles à un utilisateur humain légitime,
  - ✓ détecter des comportements caractéristiques d'un scraper automatisé
  - ✓ différencier un utilisateur humain d'un outil d'automatisation,
  - ✓ appliquer des mécanismes de protection permettant de :
    - limiter l'accès aux données,
    - refuser certaines requêtes,
    - ou bloquer complètement le scraper,
  - ✓ protéger les informations sensibles contre l'extraction automatisée.

# M4-C1 : Automatisation après authentification

- Le projet met en évidence :

- ✓ la différence fondamentale entre navigation humaine et automatisée,
- ✓ les techniques utilisées par les applications Web pour se défendre contre le scraping,
- ✓ les limites des protections côté serveur et côté client,
- ✓ l'importance de concevoir des applications Web résilientes face à l'automatisation abusive,
- ✓ la dimension défensive du concept “**Ghost in the Browser**”, où le navigateur devient un point de contrôle plutôt qu'un vecteur d'attaque.