# ☠ CorsicanCTF ☠

| |
|---|
| 🏠 Università di Corsica |
| 🗓 Vendredi 07-11-2025 |
| ⏱ 9:00-16:00 |
| 🧍 Master 2 Dev Full Stack & Master 2 Data Engineer |

## 1. Modalités

Vous les trouverez notamment sur le site officiel du CTF et sur le PDF des organisateurs. En voici un résumé :

- Orga : Mediterranean Institute of InfoSec (MIInS)
- Code d'enregistrement (en leet speak ^^) : C1C4MP3MU
- Chat Revolt
- Équipes de **6 joueurs maxi** (pensez à équilibrer !)
- Communication entre équipes interdite (jouez le jeu !)
- À priori pas de netcat, Nmap, ou SSH
- Les profs sont encouragés à participer

Le but d'un CTF est de récupérer des **flags de validation** (qui sont tout simplement des fichiers contenant un "flag", souvent un texte dans un fichier) pour prouver votre intrusion dans un système, et donc votre réussite. Un CTF simule une situation réelle (attaque ou red team, défense ou blue team).

Il y a généralement des challenges de plusieurs types, qui couvrent tous les domaines de l'informatique : cryptographie, stéganographie, exploitation de failles système, reverse engineering, cracking, injection de code, manipulation de fichiers, analyse réseau, information gathering & OSINT, attaques client, attaques serveur, développement sécurisé et détection de bugs, attaques sur le cloud et les conteneurs, énigmes, logique…

Les CTF constituent un mélange de tout cela. Ils impliquent souvent une **escalade de privilèges**.

**Le hacking est une discipline (voire un art !) extrêmement difficile, souvent ingrate et toujours exigeante qui nécessite de très bonnes connaissances à jour dans des domaines variés, un esprit logique/mathématique, du recul, de la rigueur.**

## 2. Logistique

Du fait du temps de préparation en classe très court, il va falloir que l'on s'entraîne principalement par nous-mêmes pour le CTF.

L'idée est de **constituer les équipes en amont**, et de tester nos compétences ensemble. Cela va nous permettre d'apprendre à travailler à plusieurs : il faut donc créer les équipes le plus tôt possible et s'entraîner et acquérir des connaissances.

Travailler ensemble sur un projet de plusieurs semaines/mois ou réviser ses examens est très différent de s'entraîner avec une contrainte temporelle courte. Lorsque vous vous entraînez sur un challenge, un site, fixez-vous une limite de temps réaliste. Même si vous ne finissez pas, cela permet de voir comment vous perdez du temps (souvent : manque de connaissances, mauvaise répartition des tâches). Plus vous essayerez plus vous vous améliorerez.

Si vous avez des compétences particulières, vous pouvez déjà réfléchir à une répartition des tâches qui en tient compte.

Les challenges seront vraisemblablement pensés pour offrir une difficulté progressive : tout le monde y trouvera son bonheur.

Ayez sous la main une liste d'URL d'outils en ligne pratiques comme CyberChef ou de type RequestBin (pour récupérer des requêtes). Gardez également à proximité des notes sur les challenges que vous aurez résolus durant vos entraînements, ça peut toujours servir.

Bien que la victoire soit l'objectif ultime, **s'amuser reste la première des priorités** !

# 3. Préparation matérielle

**Vous :** venez avec votre PC portable et votre smartphone chargés. Prenez tous vos chargeurs, des câbles supplémentaires, des casques/écouteurs.

Vos périphériques sans fil doivent être eux aussi chargés.

Vous pourriez être amenés à utiliser vos téléphones en point d'accès si le firewall de la fac vous bloque (gardez de la data disponible).

L'idée est de ne pas retourner à la maison y chercher quelque chose qu'on a oublié.

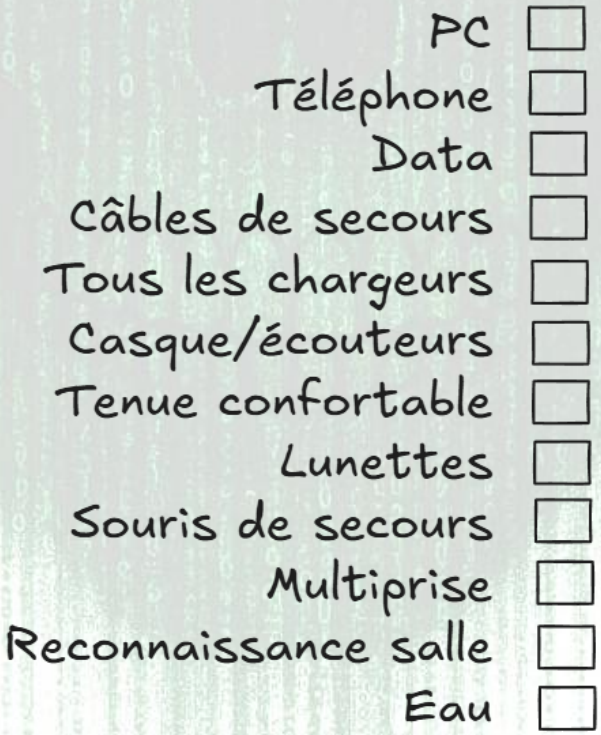
Dès que possible nous irons reconnaître notre environnement c'est-à-dire la ou les salles où se tiendra le CTF.

N'oubliez pas vos lunettes de vue si vous en avez.

Votre tenue doit être confortable car vous allez la porter 7h durant.

Prenez de l'eau en quantité suffisante (gourde isotherme).

**L'équipe :** n'oubliez pas une ou deux souris de secours, ainsi qu'une multiprise.



# 4. Préparation logicielle

**Un bon set-up est indispensable.** J'entends par là non pas une machine de guerre de gamer mais simplement un PC fonctionnel, avec un bon écran, un bureau clean, suffisamment de place sur vos disques durs, **et sans bugs**. Vous devez donc éliminer au préalable ces derniers pour ne pas passer la journée du CTF à faire des reboot ou des installations de drivers.

**Kali Linux est fortement recommandé pour les CTF.** Il faut donc l'installer dès maintenant si ce n'est déjà fait (il y a des milliers de tutoriels pour cela). Plusieurs solutions existent, la moins contraignante est de le faire tourner en tant que machine virtuelle (VirtualBox, VMWare…). Connaissez par ♥ vos mots de passe (admin pour les sudo, credentials sur les sites divers).

Le but n'est pas d'utiliser une distro quelconque "parce qu'on n'aime pas Kali" pour passer ensuite des heures à réinstaller ce que Kali possède déjà. C'est une perte de temps inutile.

Toutefois, si vous maîtrisez un autre OS orienté sécurité/pentesting (comme ParrotOS, BlackArch…) et que vous l'avez déjà installé, c'est ok, utilisez-le.

**Dans tous les cas votre système doit être à jour (Linux mais aussi Windows).** Il ne s'agit pas de passer la première heure du challenge à faire des upgrade système. Quelques jours avant le CTF, vous ouvrirez un terminal et ferez :

- sudo apt update
- sudo apt dist-upgrade

Vérifiez ensuite que les programmes usuels de Kali tournent.

Ayez sous la main tous les compilateurs et interpréteurs nécessaires.

**En résumé : il faut arriver avec un environnement de travail propre et fonctionnel !**

# 5. Prérequis généraux indispensables

- Notations décimales (base 10, acquises au primaire…), binaires (base 2), hexadécimales (base 16), octales (base 8) et savoir passer d'une base à l'autre sans hésiter
- Fonctionnement d'un réseau : modèle OSI vs TCP/IP, routage, commutation, paquets, les protocoles TCP/UDP, le protocole IP, les adressages (IPv4, IPv6, MAC…), Wireshark
- Windows et Windows Server : arborescence, journalisation, processus, gestion des users, des groupes, des droits …
- Requêtes HTTP : GET, POST… et savoir les forger (avec le navigateur, CURL, Burp Suite…), interactions client/serveur
- Éditer un fichier avec un éditeur hexadécimal
- Avoir une vision claire des mécanismes d'authentification sur le Web (stateful avec session, stateless avec JWT)
- Interactions fondamentales entre un microprocesseur (adressage, registres), la RAM (et les structures complexes associées : pile, tas), un programme (appels, fonctions, variables)
- Compilation, interprétation
- Identifier/produire des hash selon différents algorithmes
- Lire les métadonnées d'un fichier (EXIF…)
- Connaître le fonctionnement des chiffrements symétriques (DES, AES…) et asymétriques (RSA…) utilisés entre autres dans HTTPS
- Connaître les principaux protocoles/services et leurs ports TCP/UDP par défaut
- Maîtriser les bases de données et le langage SQL
- Bases de l'OSINT : Open Source INTelligence (maîtriser quelques outils associés)
- Maîtriser plusieurs langages de programmation : PHP (pour les challenges orientés web server), Python, C, JS (pour les challenges orientés web client), et même HTML !
- Savoir écrire/comprendre rapidement des algorithmes simples
- Savoir mener une veille technologique efficace
- Gérer son temps correctement
- Travailler en équipe et s'entendre sur la répartition des tâches
- Prendre des notes efficacement, afin de ne pas refaire les choses que vous avez déjà testées
- Last but not least : une (très) bonne maîtrise de l'Anglais !

**En résumé : il faut de larges connaissances informatiques, le plus à jour possible, et beaucoup, beaucoup de pratique…**

# 6. Prérequis Linux (Kali) indispensables

- Se situer et naviguer dans l'arborescence (cd, pwd…) et maîtriser les chemins absolus (qui partent de la racine) et relatifs (qui partent du répertoire courant)
- Créer/renommer/supprimer/déplacer un fichier ou un répertoire (touch, rm, rmdir, mv…)
- Lire le contenu d'un fichier (cat)
- Lister le contenu d'un répertoire (ls - options usuelles)
- Système de droits et notations (symbolique comme rwxrwxr--, leur équivalent en octal comme 774), modifier ces permissions (chmod)
- Connaître les permissions spéciales (suid, sticky bit…)
- Utiliser des commandes de recherche de fichiers/dossiers, par exemple find / -name "nom_fichier"
- Consulter l'aide d'une commande (man…)
- Utiliser les commandes de recherche de texte (grep…)
- Se connecter en SSH (avec mot de passe, avec clef privée…)
- Maîtriser les bases des redirections et des pipes
- Manipuler des archives (tar, gzip, zip, unzip)
- Utiliser les logs
- Produire un hash : echo -n "salut" | sha256sum
- Fichiers système "sensibles" : /etc/passwd, /etc/shadow, /etc/group…
- Bases de la gestion des processus : ps
- Savoir où trouver les logs (/var/log, journalctl…)

**En résumé : sachez parfaitement utiliser la ligne de commande pour interagir rapidement et efficacement avec votre OS de type POSIX.**

# 7. Prérequis de cybersécurité

Vous devez creuser dans toutes les directions suivantes :

**Web - client**

- XSS (stored, reflected, DOM-based)
- CSRF (Cross-Site Request Forgery)
- Clickjacking
- Open redirect

**Web - serveur**

- SQL Injections
- Command injection
- File inclusion (LFI/RFI)
- SSRF (Server-Side Request Forgery)
- Insecure deserialization
- Upload de fichiers non filtrés
- Directory traversal & fuzzing (Dirb, GoBuster…)

**API et logique applicative**

- IDOR (Insecure Direct Object Reference)
- Auth bypass (tokens JWT, OAuth)
- Race conditions

**Authentification & session**

- Bruteforce / dictionnaire (Hydra pour SSH, etc)
- Session hijacking
- JWT : algorithmes faibles
- Réinitialisation de mot de passe non sécurisée
- MFA contournable

**Réseau**

- Analyse de captures de trames & paquets (Wireshark)
- Attaque MITM (ARP poisoning, DNS spoofing)
- Fingerprinting (HTTP, FTP, SMB…)

**Cryptographie**

- Faiblesses de hash (MD5 & collisions, cracking avec Hashcat)
- RSA mal implémenté (PKCS#1 v1.5)
- SSL/TLS : configurations obsolètes

**Exploit binaire & reverse engineering**

- Buffer overflow (stack/heap)
- Format string vulnerabilities
- Analyse statique / dynamique (éditeur hexa, désassembleur, GDB)

**OSINT & recherche inversée**

- Scraping de données (réseaux sociaux, shodan.io)
- Recherche d'images : Google images, TinEye
- Recon DNS
- Exploitation de CVE connus (exploit-db, GitHub)
- Analyse de fichiers (Exif avec exiftool, métadonnées)

**Outils & méthodologie**

- Scripting (Python, Bash, PHP)
- Automatisation (Burp Suite, OWASP ZAP)
- Frameworks de pentest (Metasploit indispensable)
- Prise de note (Obsidian, Notion…)

La phase d'Information Gathering consiste à collecter un maximum d'informations sur votre cible. Ces informations vous permettront d'identifier les services qu'elle propose (CMS, serveur Web, mail, FTP … etc). Si vous connaissez la version du service utilisée, le port, vous pouvez mener des recherches sur les sites qui répertorient les failles de sécurité (CVE). Constituez-vous une liste de ces sites pour le jour J.

GitHub propose aussi pas mal d'exploits (un exploit est un code qui permet de tirer profit d'une vulnérabilité connue).

Pour l'entraînement proprement dit, **commencer par des CTF ne va pas vous servir à grand-chose** … à part vous dégoûter.

Mieux vaut débuter avec des challenges simples et ciblés (vous avez comme ça une idée de quel type de vulnérabilité exploiter).

Pour cela, vous pouvez faire les challenges faciles de :

- RootMe (ceux qui rapportent peu de points) dans les sections suivantes : Web Client, Web Server, Cryptographie, Réseau, App Script, Stegano…
- Il y a aussi ceux d'OverTheWire (le wargame "Natas" est un bon point de départ)
- ceux proposés par PortSwigger vous aideront à utiliser Burp tout en vous guidant dans l'exploitation de vulnérabilités

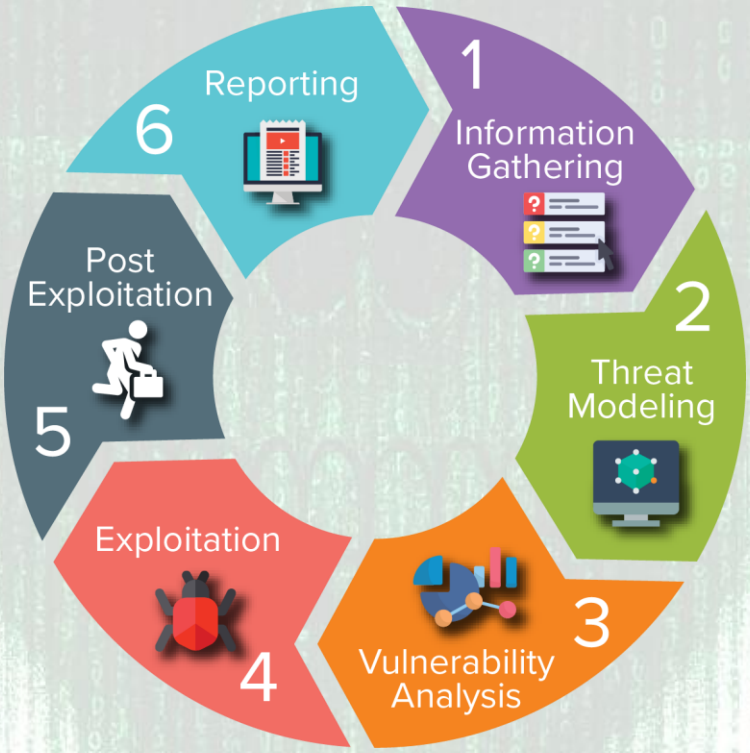
Ensuite vous enchaînerez avec des challenges plus complexes faisant appel à différentes techniques d'exploitation. Certains de TryHackMe sont assez faciles et sympas (classez-les par niveau de difficulté/gratuité). Attention, TryHackMe nécessite d'utiliser OpenVPN pour pouvoir vous connecter à leurs machines.

Ceux de HackTheBox sont à mon sens plus difficiles (même ceux de la catégorie "facile").

Vous pouvez, si vous séchez, regarder les walkthrough proposés (mais l'idée est quand même de chercher !).

Quand il s'agit de trouver un mot de passe (cracker un hash, bruteforcer une session SSH…) le mot de passe se trouve très souvent dans la liste pédagogique issue d'un célèbre leak : rockyou.txt (trouvable sur GitHub, et un peu partout).

Pour bruteforcer les noms de répertoires, Kali dispose de pas mal de dictionnaires qui font le job (pour les challenges).




Source de l'image : https://www.malwarebytes.com/blog/news/2019/09/hacking-with-aws-incorporating-leaky-buckets-osint-workflow

# 8. Conclusion

**Toutes ces listes sont non exhaustives** et ne sont pas triées selon le niveau de "difficulté" : certaines notions sont moins faciles à maîtriser que d'autres. Mais cela constitue la base de la base, **vous ne devez pas faire d'impasse**. En conséquence, si l'un de ces points n'est pas parfaitement clair, **entraînez-vous** !

Il existe pas mal de bouquins sur le sujet, méfiez-vous de ceux aux titres accrocheurs. Personnellement je trouve que les ouvrages traitant du bug bounty sont généralement pas trop mal faits.

**Dans tous les cas il y a d'innombrables ressources gratuites et de qualité, y compris et surtout sur le clearweb !**

Ajoutons qu'une bonne utilisation de l'**IA en amont du concours** serait de lui demander de vous créer des exercices complets couvrant un plusieurs de ces thèmes, et de faire des allers-retours entre elle et votre OS pour vous corriger et vous améliorer, pour lui demander des alternatives…

**Pour conclure, rappelez-vous que le Hacking avec un grand H, a une histoire, un état d'esprit, et une éthique !**

# 9. Liens utiles

| | |
|---|---|
| Orga du CTF | ctf.miins.org |
| Chat Revolt du CTF | rvlt.gg/gpBJHt8m |
| RootMe | root-me.org |
| TryHackMe | tryhackme.com |
| PortSwigger Academy | portswigger.net/web-security |
| Hack The Box | hackthebox.com/hacker/hacking-labs |
| OverTheWire | overthewire.org/wargames |
| Pico CTF | picoctf.org |
| Site de l'OWASP | owasp.org |
| Hack Tricks | book.hacktricks.wiki |
| Des payloads à foison | github.com/swisskyrepo/PayloadsAllTheThings |
| Cyber Chef | gchq.github.io/CyberChef |
| FuzzDB | github.com/fuzzdb-project/fuzzdb |
| Exploit DB | exploit-db.com |
| Common Vuln Exposure | cve.org |
| Exploits Linux | github.com/The-Z-Labs/linux-exploit-suggester |
| NMAP | nmap.org |
| Wordlists utiles | github.com/danielmiessler/SecLists |
| Ressources de qualité | github.com/Hack-with-Github/Awesome-Hacking |
| VulnHub | vulnhub.com |
| Metasploit | metasploit.com |
| Tin Eye | tineye.com |

# Have Fun !