Projet 1

Création d’une plateforme Web pour collecter les vulnérabilités des objets connecté

Denéchère Killian - Moulin Luc - Julien Chaupitre

I. Sommaire

II. Présentation du projet

Avec l'essor et la démocratisation de l'Internet des Objets (IoT), de plus en plus d'objets sont connectés. Cela pose alors la question de la sécurité. Les utilisateurs, souvent peu sensibilisés à ces risques, peuvent laisser leurs appareils vulnérables aux attaques, exposant ainsi leurs données personnelles et le réseau auquel ils sont connectés.

L'objectif de ce projet est de créer une plateforme web informative pour aider les utilisateurs à sécuriser leurs objets connectés. Cette plateforme informera les utilisateurs sur les vulnérabilités de sécurité des objets connectés en prenant en compte des éléments tels que l'objet concerné, sa version, la sévérité de la vulnérabilité, et bien d'autres aspects. De plus, le site sera constamment à jour en s'adaptant aux nouvelles menaces et aux nouveaux appareils, garantissant ainsi une protection optimale aux utilisateurs.

III. Choix technologiques

Pour ce projet, nous avons décidé d'opter pour l'utilisation du framework Python "Django". Celui-ci a déjà été utilisé par de nombreuses grandes entreprises telles qu'Instagram. Nous l'avons choisi en raison de sa rapidité de développement et de sa documentation très claire. De plus, notre équipe maîtrise déjà le langage Python, ce qui garantit un développement efficace, robuste et de qualité.

Pour la gestion des données, nous utiliserons la base de données SQLite, car elle est intégrée nativement à Django. Elle présente l'avantage d'être robuste et de disposer de fonctionnalités avancées qui seront utiles pour la suite du projet tout en restant très simple à gérer et rapide à mettre en place.

Django nous sert donc à la fois de serveur web organisé sous la forme de MVC et de serveur de base de données. Cette centralisation des usages nous permet de simplifier la création d'interfaces et le partage des données entre les divers composants du site.

IV. Partie base de données

Pour réaliser ce projet, nous avons besoin d'une quantité de données importante concernant les failles de sécurité de l'IoT déjà publiées. Ainsi, nous avons cherché à utiliser plusieurs API pour nous fournir ces données.

A. Nécessité du stockage

La plupart des API publiques ayant des accès gratuits restreint en termes de nombre de requêtes dans un laps de temps définie, nous avons pensé à mettre en place une base de données locale afin de stocker ces données. La base de données permet à la fois le stockage des informations de l'API et donc la non nécessité de joindre ce système à chaque chargement de page de notre site, mais elle permet aussi une meilleure rapidité de chargement du site. Avec la base de données, nous ne dépendons ni du service qui met à disposition son API, ni de la connexion internet de la personne ayant démarrée le serveur Django.

Pour remplir ce travail, nous avons choisi d'utiliser une base de données SQLite3 pour sa simplicité de gestion et d'utilisation.

B. Structure de la base de données

Comme expliqué précédemment, nous avons choisi d'utiliser Django en tant que serveur pour notre site web. Django inclut par défaut tout un système de gestion des données en base de données. Ce système se base sur un principe d'ORM (object-relational mapping) c'est-à-dire que nous créons des classes python avec des attributs particuliers qui reflètent les tables et colonnes de notre base de données. La manipulation d'instances de ces classes permet de refléter ces actions en base de données automatiquement.

Nos classes ont été créées en fonction des informations dont nous disposions dans l'API. Nous ne sauvegardons pas toutes les informations, mais seulement celles que nous souhaitons afficher par la suite sur notre site.

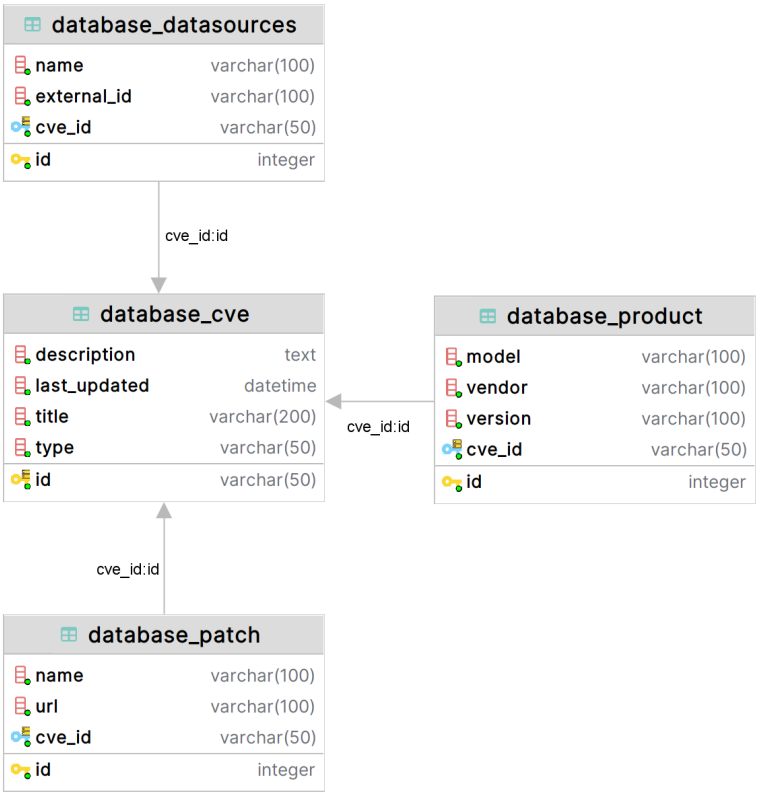


Schéma de la base de données

C. Changement de source de données

Nous avons tout d'abord utilisé une première API concernant toutes les CVE en tant que sources de données. Dans un second temps, nous avons cherché à filtrer les CVE concernant les appareils issus de l'internet des objets, mais nous nous sommes rendus compte que les résultats obtenus avec nos filtres n'étaient pas très concluants.

Face à ce constat, nous avons fait le choix assez tardif de changer de source de données pour une autre API dédiée, elle, aux objets de l'IoT. Ce changement nous a permis de ne plus avoir à différencier les failles de sécurité concernant l'IoT des autres, mais celà nous a contraint à revoir totalement la structure initiale de la base de données.

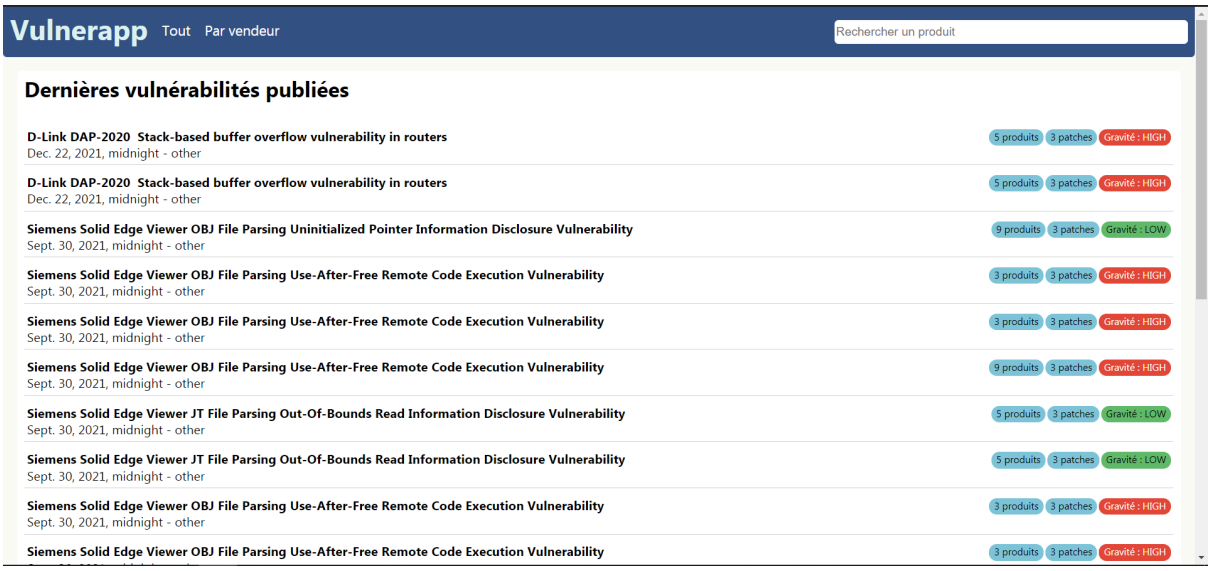
Grâce à cette nouvelle API, nous avons aussi obtenus de nouvelles informations autour des CVE comme les patchs disponibles ou bien les liens vers les différentes bases de données de failles officielles (gouvernements, constructeurs, ...).

V. Fonctionnalités du site

Le site est assez simple d'utilisation et de présentation. Plusieurs pages citées dans la suite de ce document sont mises à disposition de l'utilisateur.

1. Page d'accueil

La page d'accueil du site nous présente les 20 dernières failles recensées sur la plateforme. Pour chaque vulnérabilité, différentes informations sont affichées pour guider l'utilisateur dans sa navigation. C'est une présentation que nous retrouverons sur d'autres pages du site.

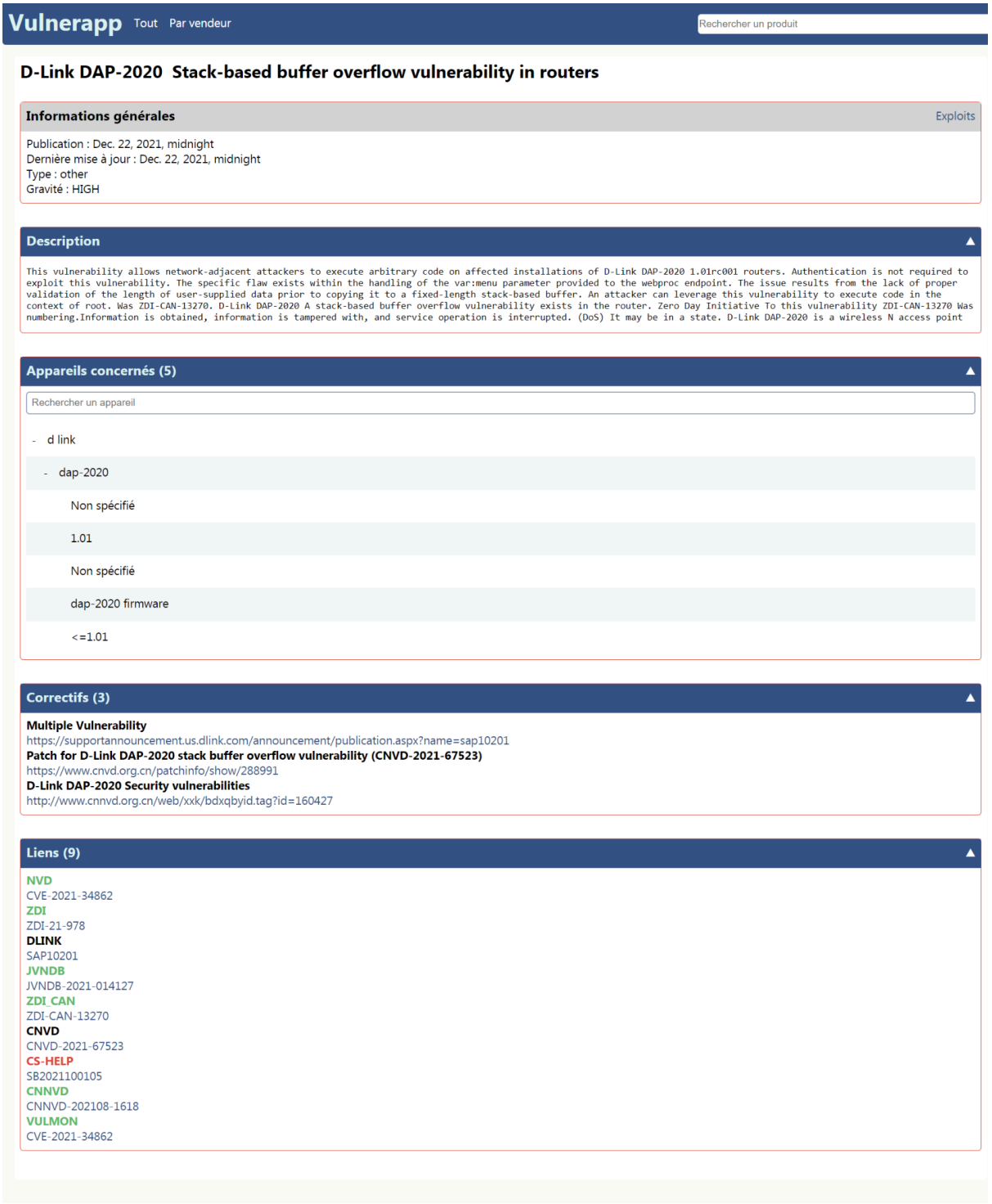


Page d'accueil du site

Un menu est toujours présent en haut de chaque page du site de sorte à retourner sur les pages principales du site à tout moment.

2. Page de détails d'une CVE

Chaque CVE possède une date de publication, une gravité, une description, des appareils concernés par cette CVE, qui peuvent être multiples. Des correctifs sont disponibles, ils peuvent être des liens vers le site du constructeur présentant le correctif. Enfin, des liens sont fournis pour savoir sur quelles bases de données nous pouvons retrouver cette CVE.



Page de détails d'une faille

Par défaut, les différentes catégories sont repliées sur elles-même car le contenu de celles-ci peut être très grand : cela varie en fonction des failles et de leur portée.

Dans la partie des appareils concernés par la faille, ceux-ci sont présentés sous la forme d'un arbre dépliable où l'on commence par le vendeur du produit qui contient les différents modèles de produits qui eux-mêmes contiennent les versions de produits concernés.

Nous listons les corrections publiées pour cette vulnérabilité ainsi que les liens connexes avec différentes couleurs. Le vert représente les bases de données connues dont une page dédiée est disponible sur le web. Le rouge représente, à l'inverse, les base de données connus pour lesquelles aucune page dédiée n'est disponible sur le web. Le bleu correspond aux bases de données où la vérification de la présence de page ou non n'a pas été faite par notre système. Le but de ces couleurs est d'aider l'utilisateur dans le choix des informations complémentaires qu'il souhaite obtenir.

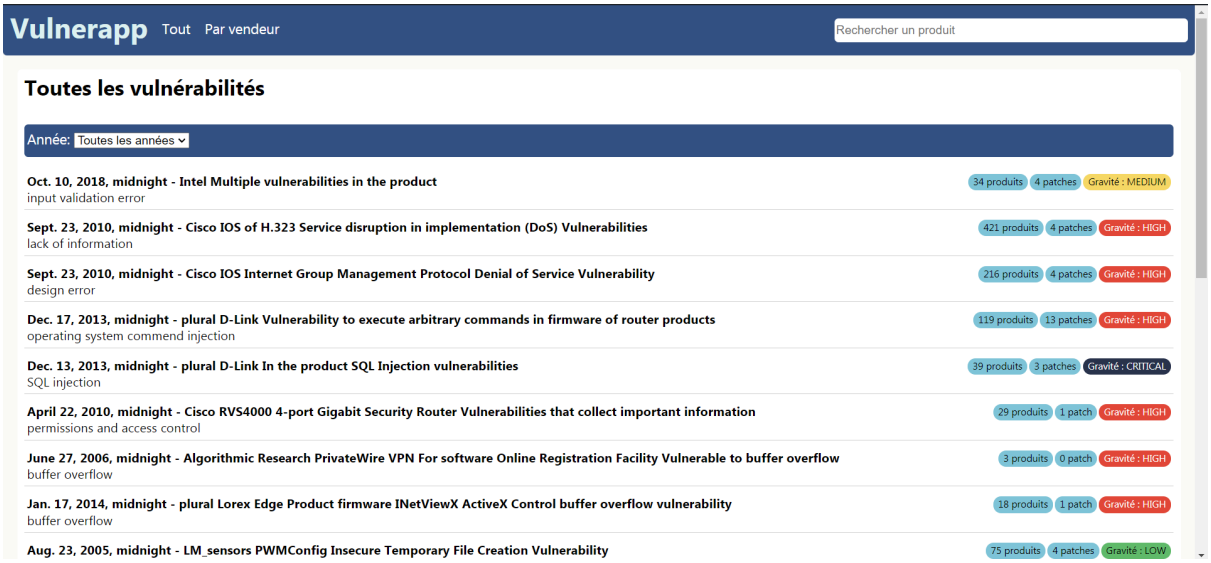
On notera aussi le lien "Exploits" disponible dans la première partie de la page "Informations générales" qui nous redirige vers la base de données https://www.exploit-db.com/ sur laquelle nous pouvons retrouver des POC mettant en œuvre certaines failles. Nous aurions aimé mieux intégrer ces informations en allant directement vérifier la présence ou non d'exploits sur cette base de données et d'autres, mais nous n'avons pas eu le temps de réaliser cet ajout.

3. Toutes les failles

Avec le menu disponible en haut de page, l'utilisateur peut se rendre sur la page listant toutes les failles incluses dans notre système.

Il est possible de trier ces failles par années et de changer de pages en pages grâce au système de pagination intégré.

Nous n'avons malheureusement pas eu le temps d'intégrer plus d'outils de filtrage et de tri. Nous aurions souhaité par exemple ajouter la possibilité de filtrer par gravité de faille ou bien de trier par nombre de produits concernés.



Page listant toutes les vulnérabilités du site

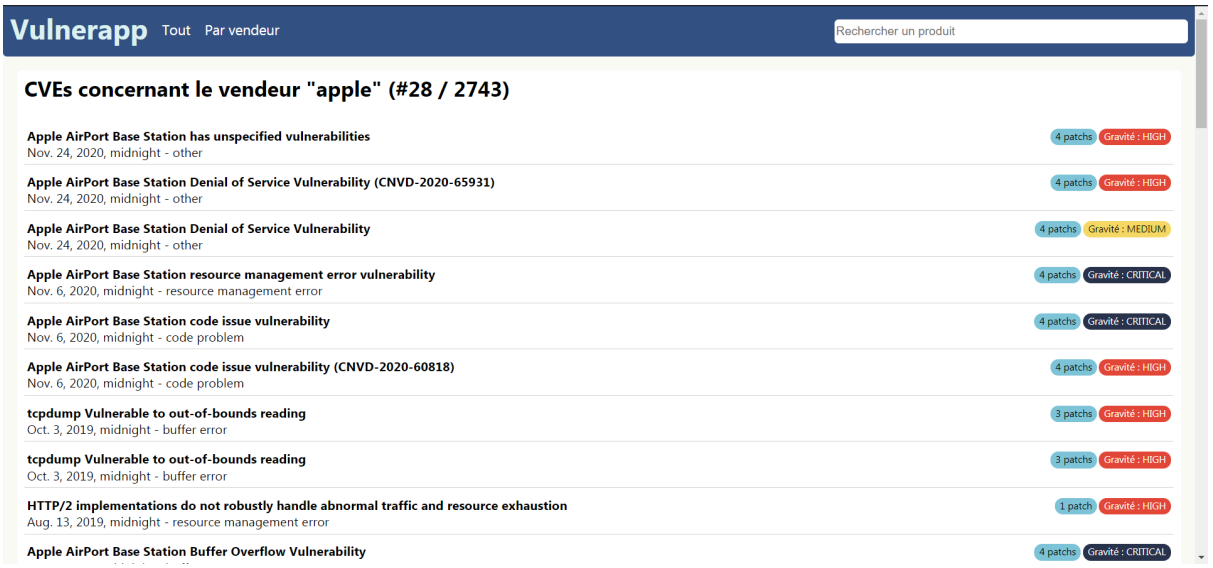
4. Liste des vendeurs

Sur le même principe que la page listant toutes les failles de notre application, l'utilisateur peut se rendre sur la page listant les différents vendeurs de produits concernés par des failles de sécurité.

Là encore, un système de pagination est intégré tout comme un système de recherche dédié aux vendeurs (plus rapide que la recherche globale).

Page listant les vendeurs

En cliquant sur l'un des vendeurs listés sur l'interface ci-dessus, l'utilisateur arrive sur la page listant les failles attribuées à ce vendeur.



Liste des failles pour le vendeur "apple"

Nous retrouvons l'affichage déjà présenté des failles ainsi qu'une information supplémentaire concernant le classement du vendeur concerné par la recherche. Nous pouvons voir qu'Apple est le 28ème vendeur de produits ayant le plus de failles lui étant liées sur les 2743 vendeurs recensés dans notre base de données (Cisco étant le premier de notre liste).

VI. Axes d'amélioration

La version actuelle du projet n'est bien sûr pas assez aboutie pour une mise en production. Cela est principalement dû au contexte de cet exercice dont l'objectif n'est pas la mise en place d'une plateforme complète. Aussi, nous avons identifié des axes d'améliorations que nous pourrions explorer si nous étions amenés à continuer ce projet :

1. Métriques & score de vendeurs

Nous pourrions ajouter une métrique permettant de mesurer la rapidité de correction des produits. Par exemple en suivant le temps écoulé entre la découverte d'une vulnérabilité et la publication d'un correctif ou encore en mesurant le nombre de corrections publiées face au nombre de produits concernés ou la gravité de faille.

Nous pourrions ainsi établir un score de fiabilité des différents constructeurs / vendeurs.

2. Interprétation et recherche avancée

En intégrant une couche d'intelligence artificielle basée sur nos données, nous pourrions améliorer la recherche face aux demandes de l'utilisateur. Nous pourrions aussi faire en sorte qu'elle puisse détecter et corriger les fautes de frappe et erreurs qui se sont glissées dans la base de données.

Nous aimerions aussi calculer différentes informations autour des différents constructeurs ou produits afin de guider l'utilisateur dans le choix de ses appareils avant achat ainsi que dans la façon de mettre à jour leurs produits pour ne plus que ceux-ci soient vulnérables.

Un bon exemple de ce que nous souhaiterions atteindre est le site https://www.cvedetails.com/.

3. Plus de données

Nous recensons environ 5000 CVE actuellement et l'API que nous utilisons nous permettrais d'en récupérer jusqu'à 42.000 environ. En ajoutant plus de données dans notre base de données, nous améliorons les métriques précédemment présentées et permettons aux utilisateurs de retrouver plus de résultats pour leurs appareils.

Nous pourrions aussi intégrer des données issues d'autre sources de données comme le CWE (Common Weakness Enumeration), qui permet de catégoriser les types de faiblesses et de vulnérabilités dans les logiciels. Cette métrique peut fournir une perspective différente sur les risques potentiels associés à un produit donné.

4. Intégrer les exploits

Au même titre que les bases de données de failles, il existe des bases de données répertoriant les exploitations de failles. Souvent, ces base de données présentent du code malveillant, mais aussi des informations bien plus techniques autour de ces failles avec des Proofs Of Concept ou des démonstrations. Ces informations seraient très utiles pour un public plus averti et dont l'objectif ne serait plus une simple exploration des failles.

VII. Conclusion

Nous avons apprécié réaliser ce projet car ce fut l'occasion pour nous l'occasion de découvrir et d'approfondir notre connaissance autour des CVE. Nous avons pu voir qu'énormément de failles sont signalées tous les jours à travers le monde et que les systèmes numériques actuels ne sont pas à l'abri d'attaques exploitant des vulnérabilités.

Ces bases de données sont donc très importantes à la fois pour le public qui cherche à se tenir au courant de la fiabilité/sécurisation des produits qu'il achète et à la fois pour les professionnels du domaine qui effectuent de la veille technologique.

Grâce à ce projet, nous avons acquis des connaissances et techniques qui nous seront très utiles dans le milieu de la cybersécurité autant à titre personnel que professionnel. Il est essentiel de se tenir à jour sur les failles de sécurité et c'est ce projet qui nous a permis d'en prendre conscience.

Nous avons aussi découvert l'existence de bases de données d'exploits liés à des vulnérabilités. Nous avons même pu découvrir certaines failles qui étaient encore disponibles sur nos machines personnelles.

A force de manipuler les CVE, nous avons acquis une compréhension plus fine de leur construction et de leur structure. Nous sommes convaincus que cette étude sera précieuse dans nos futures missions professionnelles.