IPSSI

13 rue Claude Tillier

75013, Paris

Veille technologique

Effectué le 9 octobre 2023

Spécialité : Technicien Systèmes et réseaux

Réalisé par :

Djimtone Ngarndo Moguidbe

Encadré par :

Antoine Millot

Table des matières

Introduction générale 2
PARTIE 1 : OUTILS UTILISES 2
PARTIE 2 : LES VULNERABILITES D'UN RESEAU INFORMATIQUE 3
Définition 3
Causes 4
Identifications et correction des vulnérabilités 4
Exploitation malveillante : 5
Exemple de vulnérabilités connues : 6
Articles intéressants à visiter : 8

Introduction générale

Dans cette documentation, nous présenterons notre veille technologique qui est les vulnérabilités d'un réseau informatique.

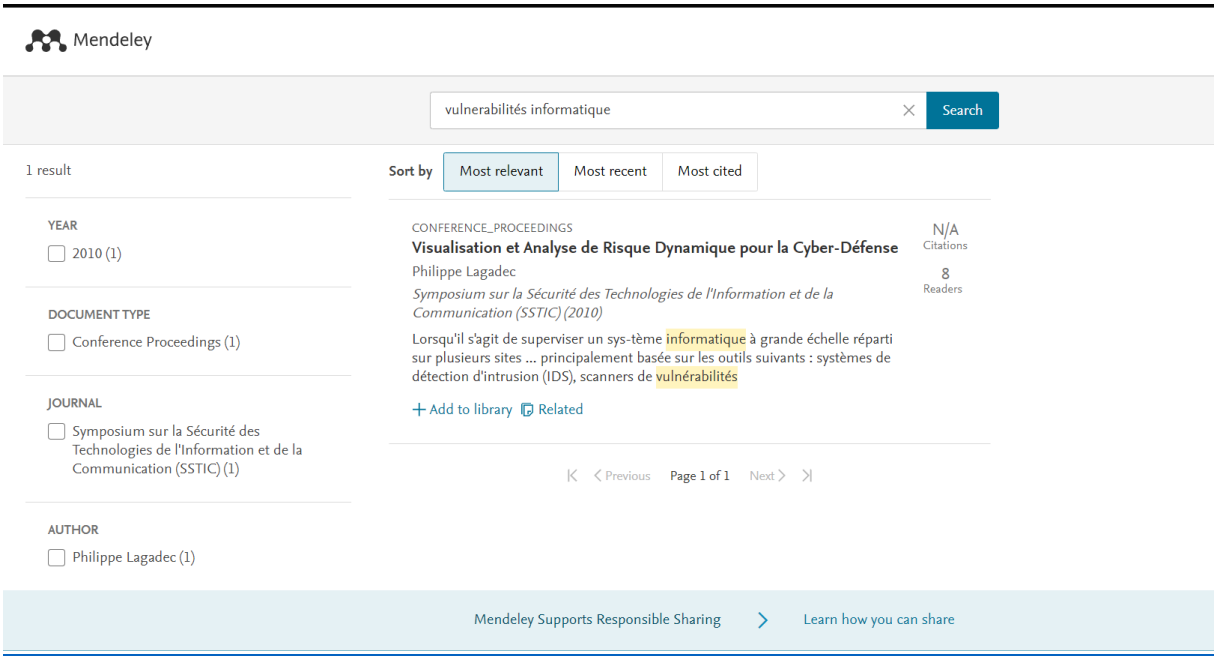
Nous présenterons les outils que nous avons utilisé pour la réalisation de cette documentation et quelques sites importants à visiter.

PARTIE 1 : OUTILS UTILISES

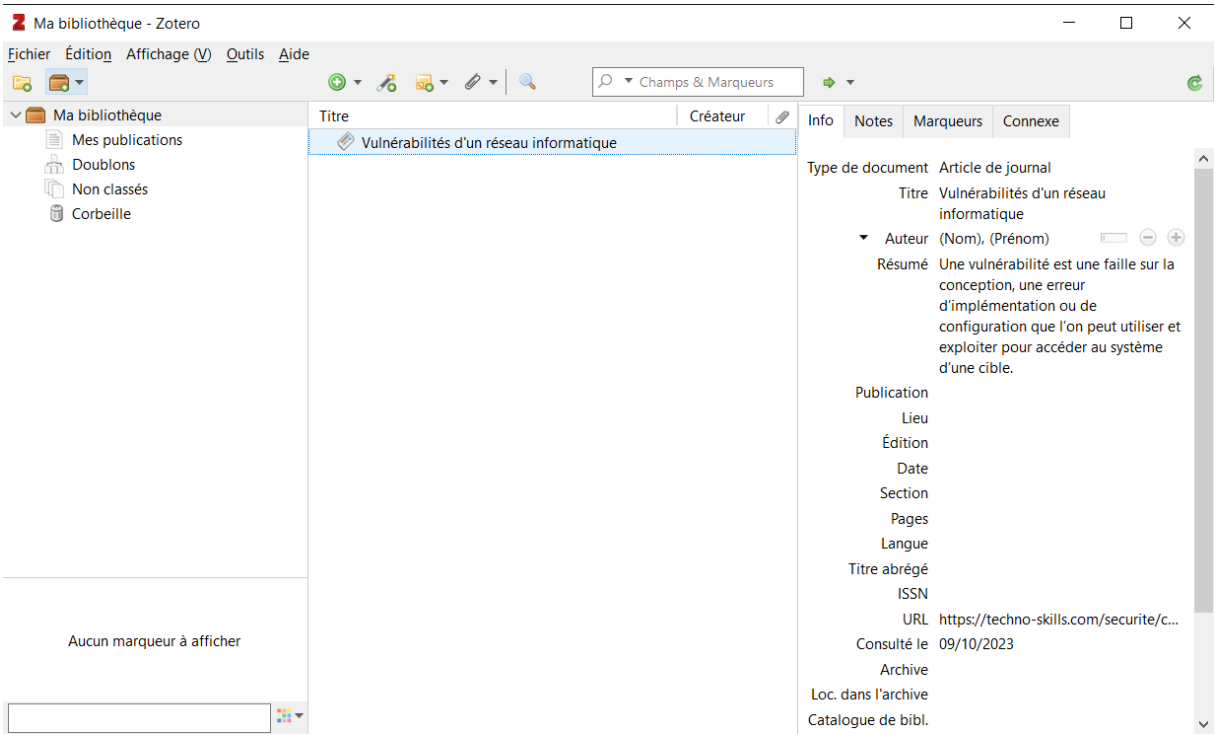
I- Outil pour Webographie :

Mendeley:

https://www.mendeley.com/search/?page=1&query=vulnerabilit%C3%A9s%20informatique&sortBy=relevance



Zotero:



PARTIE 2 : LES VULNERABILITES D'UN RESEAU INFORMATIQUE

Définition

Dans le domaine de la sécurité informatique, une **vulnérabilité** ou **faille** est une faiblesse dans un système informatique permettant à un attaquant de porter atteinte à l'intégrité de ce système, c'est-à-dire à son fonctionnement normal, à la confidentialité ou à l'intégrité des données qu'il contient.

Ces vulnérabilités sont la conséquence de faiblesses dans la conception, la mise en œuvre ou l'utilisation d'un composant matériel ou logiciel du système, mais il s'agit souvent d'anomalies logicielles liées à des erreurs de programmation ou à de mauvaises pratiques. Ces dysfonctionnements logiciels sont en général corrigés à mesure de leurs découvertes, mais l'utilisateur reste exposé à une éventuelle exploitation tant que le correctif (temporaire ou définitif) n'est pas publié et installé. C'est pourquoi il est important de maintenir les logiciels à jour avec les correctifs fournis par les éditeurs de logiciels. La procédure d'exploitation d'une vulnérabilité logicielle est appelée exploit.

Causes

Les vulnérabilités informatiques proviennent souvent de la négligence ou de l'inexpérience d'un programmeur. Il peut y avoir d'autres causes liées au contexte comme l'évolution des technologies, notamment en cryptographie. Une vulnérabilité permet généralement à l'attaquant de duper l'application, par exemple en outrepassant les vérifications de contrôle d'accès ou en exécutant des commandes sur le système hébergeant l'application.

Quelques vulnérabilités surviennent lorsque l'entrée d'un utilisateur n'est pas contrôlée, permettant l'exécution de commandes ou de requêtes SQL (connues sous le nom d'injection SQL). D'autres proviennent d'erreurs d'un programmeur lors de la vérification des buffers de données (qui peuvent alors être dépassés), causant ainsi une corruption de la pile mémoire (et ainsi permettre l'exécution de code fourni par l'attaquant).

Identifications et correction des vulnérabilités

Il existe de nombreux outils qui peuvent faciliter la découverte de vulnérabilités sur un système d'information, certains permettant leur suppression. Mais, bien que ces outils puissent fournir à un auditeur une bonne vision d'ensemble des vulnérabilités potentiellement présentes, ils ne peuvent pas remplacer le jugement humain. Se reposer uniquement sur des scanners automatiques de vulnérabilité rapportera de nombreux faux positifs et une vue limitée des problèmes présents dans le système.

Des vulnérabilités ont été trouvées dans tous les principaux systèmes d'exploitation, en premier lieu sur Windows, mais aussi sur Mac OS, différentes versions d'Unix et Linux, OpenVMS, et d'autres. La seule manière de réduire la probabilité qu'une vulnérabilité puisse être exploitée est de rester constamment vigilant en développant la maintenance système, de déployer une architecture sécurisée (par exemple en plaçant judicieusement des pare-feu), de contrôler les accès, et de mettre en place des audits de sécurité (à la fois pendant le développement et pendant le cycle de vie).

Les téléphones mobiles et smartphones sont des équipements informatiques. Les logiciels espions utilisent les failles de sécurité des systèmes

d'exploitation iOS ou Android des téléphones mobiles et évoluent en permanence. Les vulnérabilités zero-day étant très difficiles à trouver, elles font l'objet d'un véritable marché dans lequel des hackers vendent leurs trouvailles au plus offrant.

Exploitation malveillante :

Les failles de sécurité deviennent particulièrement intéressantes lorsqu'un programme contenant une de ces vulnérabilités est lancé avec des privilèges spéciaux, qu'il permet une authentification sur un système, ou bien encore lorsqu'il fournit un accès à des données sensibles.

Les Crackers et non « hackers », grâce à leur connaissance et à des outils appropriés, peuvent prendre le contrôle de machines vulnérables. Les failles de sécurité découvertes sont généralement colmatées au plus vite à l'aide d'un patch, afin d'empêcher des prises de contrôles intempestives ; cependant dans bien des cas, des machines restent vulnérables à des failles anciennes, les différents correctifs n'ayant pas été appliqués.

Certains logiciels malveillants utilisent des vulnérabilités pour infecter un système, se propager sur un réseau, etc.

L'exploitation d'une faille peut provoquer un déni de service du système, un accès à un système ou à des informations sensibles, voire une élévation des privilèges d'un utilisateur.

On parle de **faille distante** lorsque la vulnérabilité se situe dans un logiciel constituant un service réseau (par exemple un serveur Web) et qu'elle peut être exploitée par un attaquant distant, qui ne dispose pas d'un compte local. Les vulnérabilités locales peuvent être utilisées par un utilisateur malintentionné, qui possède un compte, pour effectuer une élévation des privilèges, ou bien par un attaquant distant pour augmenter ses privilèges, après l'exploitation d'une vulnérabilité distante.

On parle de **faille locale** lorsque la vulnérabilité n'est exploitable que par un utilisateur disposant d'un compte local. Les vulnérabilités distantes peuvent être utilisées par des attaquants pour obtenir un accès sur un système.

Exemple de vulnérabilités connues :

- dépassement de tampon ;

En informatique, un **dépassement de tampon** ou **débordement de tampon** (en anglais, ***buffer overflow*** ou ***BOF***) est un bug par lequel un processus, lors de l'écriture dans un tampon, écrit à l'extérieur de l'espace alloué au tampon, écrasant ainsi des informations nécessaires au processus.

Lorsque le bug se produit, le comportement de l'ordinateur devient imprévisible. Il en résulte souvent un blocage du programme, voire de tout le système.

Le bug peut aussi être provoqué intentionnellement et être exploité pour compromettre la politique de sécurité d'un système. Cette technique est couramment utilisée par les pirates. La stratégie de l'attaquant est alors de détourner le programme bugué en lui faisant exécuter des instructions qu'il a introduites dans le processus.

- injection SQL ;

La faille SQLi, abréviation de *SQL Injection*, soit **injection SQL** en français, est un groupe de méthodes d'exploitation de faille de sécurité d'une application interagissant avec une base de données. Elle permet d'injecter dans la requête SQL en cours un morceau de requête non prévu par le système et pouvant en compromettre la sécurité.

Il existe plusieurs types d'injection SQL :

- la méthode *blind based* (associée à sa cousine la *time based*), qui permet de détourner la requête SQL en cours sur le système et d'injecter des morceaux qui vont retourner caractère par caractère ce que l'attaquant cherche à extraire de la base de données. La méthode *blind based*, ainsi que la *time based*, se basent sur la réponse du serveur : si la requête d'origine renvoie bien le même résultat qu'à l'origine (et indique donc que le caractère est valide) ou ne renvoie pas le même résultat (et indique donc que le caractère testé n'est pas le bon). La *time based* a pour seule différence qu'elle se base sur le temps de réponse du serveur plutôt que sur la réponse en elle-même ;

- la méthode *error based*, qui permet de détourner la requête SQL en cours sur le système et d'injecter des morceaux qui vont retourner champ par champ ce que l'on cherche à extraire de la base de données. Cette méthode profite d'une faiblesse des systèmes de base de données permettant de détourner un message d'erreur généré par le système de base de données et préalablement volontairement provoquée par l'injection SQL pour lui faire retourner une valeur précise récupérée en base de données ;
- la méthode *union based*, qui permet de détourner la requête SQL en cours sur le système et d'injecter des morceaux qui vont retourner un ensemble de données directement extraites de la base de données. Cette méthode profite de certaines méthodes afin de détourner entièrement le retour de la requête SQL d'origine afin de lui faire retourner en une seule requête un important volume de données, directement récupéré en base de données. Dans ses exemples les plus violents, il est possible de récupérer des tables entières de base de données en une ou deux requêtes, même si en général cette méthode retourne entre 10 et 100 lignes de la base de données par requête SQL détournée ;
- la méthode *Stacked queries*, la plus dangereuse de toutes. Profitant d'une erreur de configuration du serveur de base de données, cette méthode permet d'exécuter n'importe quelle requête SQL sur le système ciblé, ce qui ne se limite pas seulement à récupérer des données comme les 3 précédentes. En effet, quand ce type de requête n'est pas désactivé, il suffit d'injecter une autre requête SQL, et elle sera exécutée sans problème, qu'elle aille chercher des données, ou en modifier directement dans la base de données.
- cross site scripting.

Le **cross-site scripting** (abrégé **XSS**) est un type de faille de sécurité des sites web permettant d'injecter du contenu dans une page, provoquant ainsi des actions sur les navigateurs web visitant la page. Les possibilités des XSS sont très larges puisque l'attaquant peut utiliser tous les langages pris en charge par le navigateur (JavaScript, Java...) et de nouvelles possibilités sont régulièrement découvertes notamment avec l'arrivée de nouvelles technologies comme HTML5. Il est par exemple possible de rediriger vers un autre site pour de l'hameçonnage ou encore de voler la session en récupérant les cookies.

Le cross-site scripting est abrégé XSS pour ne pas être confondu avec le CSS (feuilles de style)[1], X se lisant « *cross* » (croix, à travers) en anglais.

Articles intéressants à visiter :

- https://geekflare.com/fr/top-network-vulnerabilities/
- https://oneoperateur.fr/lutilisateur-la-premiere-vulnerabilite-dun-reseau-informatique/
- https://nexxo.tech/blogue/6-vulnerabilites-de-votre-infrastructure-informatique-et-comment-les-rectifier/
- https://www.next-decision.fr/wiki/failles-securite-informatique#:~:text=Une%20faille%20de%20s%C3%A9curit%C3%A9%20ou,%C3%A0%20des%20donn%C3%A9es%20non%20autoris%C3%A9es.
- https://fr.wikipedia.org/wiki/Vuln%C3%A9rabilit%C3%A9_(informatique)
-