



VEILLER : des risques identifiés aux vulnérabilités exploitables

Eric DUPUIS^{1,2*}

🕒 Résumé

Ce document présente la dynamique du travail sur les fragilités de l'entreprise, de la maîtrise technique à la surveillance des vulnérabilités apparaissant dans l'environnement

Il fait partie du cours introductif aux fondamentaux de la sécurité des systèmes d'information vue sous deux prismes quelques fois opposés dans la littérature : la gouvernance et la gestion opérationnelle de la sécurité. Le cours est constitué d'un ensemble de notes de synthèse indépendantes compilées en un document unique, mais édité par chapitre dans le cadre de ce cours.

Ce document ne constitue pas à lui seul le référentiel du cours CYBERDEF101. Il compile des notes de cours mises à disposition de l'auditeur comme support pédagogique partiel à ce cours introductif à la cyberdéfense d'entreprise.

🔑 Mots clefs

Veille, vulnérabilités, menaces, identification

¹ Enseignement sous la direction du Professeur Véronique Legrand, Conservatoire National des Arts et Métiers, Paris, France

² RSSI Orange Cyberdefense

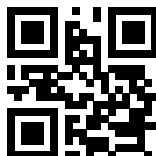
*email : eric.dupuis@lecnam.net – eric.dupuis@orange.com

DRAFT NOTES 2020-2021

Vérifiez la disponibilité d'une version plus récente de
SEC101-C3a-VulMan.doc.pdf sur GITHUB CYBERDEF ¹



2020 eduf@ction Publication en Creative Common BY-NC-ND



1. <https://github.com/edufaction/CYBERDEF/raw/master/Builder/SEC101-C3a-VulMan.doc.pdf>




1. Fragilités numériques

Nous avons vu dans l'équation d'évaluation simple du risque, que ce dernier dépendait directement des fragilités de l'entreprise. C'est par l'exploitation de ces failles que l'attaquant va pouvoir déployer toutes ses ambitions.

La notion de fragilité numérique ou digitale de l'entreprise est à prendre au sens large. Elle comprend les fragilités **humaines, organisationnelles et techniques** mais aussi la sensibilité à des scénarios d'attaques. C'est en effet la susceptibilité d'une organisation à subir des défaillances dans le temps que l'on nomme vulnérabilités.

1.1 Détecter les fragilités de l'entreprise

La première tâche de fond en cybersécurité pour une équipe dédiée est d'identifier les fragilités de l'ensemble de l'environnement numérique² de l'entreprise. Elle s'inscrit dans la dynamique de l'**anticipation** avec la recherche de fragilités ou de risques cyber dans l'entreprise et leur correction. Généralement les tâches associées à la couverture des vulnérabilités se déploient avant la **détection** d'évènement à risque, d'attaques, de déviance dans l'environnement mais aussi à l'extérieur du périmètre de l'entreprise. Elle se positionne néanmoins comme une activité qui peut déclencher des mécanismes de la **réaction** aux incidents, de la gestion de crise, par la nécessaire remédiation en cas de vulnérabilité critique. De multiples notions sont liées à la gestion des vulnérabilités, telles que le « scan » de vulnérabilités, l'évaluation de vulnérabilités (vulnerability assessment en anglais) ou l'application de correctifs (vulnerability patching ou patch management en anglais). On trouvera ces concepts bien décrits dans le livre blanc du SANS Institute Implementing a Vulnerability Management Process ³.

On peut distinguer deux grandes typologies d'actions pour identifier ces fragilités :

- ▶ l'audit de sécurité, qui permet de détecter des fragilités exploitables. Ce type d'audit peut se dérouler sous la forme de scénario exécuté par des équipes de « tests d'intrusion » soit sous la forme de campagne exécutée avec des scanners de vulnérabilités.
- ▶ la veille en vulnérabilités associée à la cartographie de l'environnement technique permettent de déclencher une alerte de sécurité si une vulnérabilité apparaissait sur un des produits, services ou logiciel surveillés.

La difficulté principale de ces activités est de bien définir les périmètres techniques et de responsabilité sur lesquelles elles portent.

Si l'audit de sécurité permet d'évaluer les fragilités des éléments (composants) de l'entreprise en se mettant dans la peau de l'attaquant, afin de découvrir les scénarios potentiellement actifs sur l'environnement digital de l'entreprise, il n'en demeure pas moins important de mettre en place des mécanismes complémentaires et continus pour la veille, la recherche, la détection, la correction de ces vulnérabilités.

2. systèmes d'information de l'entreprise, services dans les cloud, réseaux sociaux...

3. <https://www.sans.org/reading-room/whitepapers/threats/implementing-vulnerability-management-process-34180>



1.2 Anticiper et surveiller les menaces

Comme nous l'avons vu, une grande partie des attaques sur l'entreprise est liée à l'exploitation de fragilités de celle-ci, ces fragilités étant dans la plupart des cas connues.

L'exploitation de ces fragilités, sont de deux grandes natures :

- ▶ attaques exploitant de manière **opportuniste** des fragilités non cataloguées avec ou sans ciblage particulier de l'attaqué ;
- ▶ attaques **ciblées** exploitant de manière spécifique des fragilités connues mais pas corrigées ou des fragilités non encore connues par les défenseurs.

On trouvera dans le chapitre 2, une description plus précise de ces notions de vulnérabilités connues et non connues. Les menaces sont généralement des scénarios, des codes malveillants, des mécanismes d'agression ... Le principe de gestion de la menace relève de la même dynamique de gestion que celle liée aux vulnérabilités.

1.3 Les basics sur les vulnérabilités HOT

Quand nous parlons de vulnérabilités, nous parlons globalement des fragilités dans l'environnement du numérique de l'entreprise. Nous pouvons distinguer trois grandes classes de fragilités :

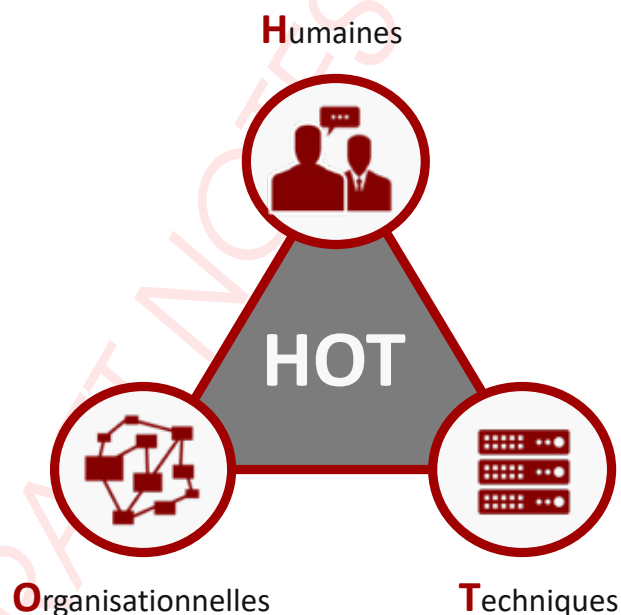


FIGURE 1. les types de vulnérabilités

- ▶ Fragilités techniques : généralement dénommées vulnérabilités au sens où ces fragilités rendent vulnérable tout ou partie d'un système. Pour rechercher ces vulnérabilités, on utilisera des techniques d'audit, de scan de fuzzing ... Ce sont ces vulnérabilités informatiques et réseaux que nous présenterons plus en détails ;



- ▶ **Fragilités humaines** : généralement des déviations comportementales, détournement d'usage légitime, sensibilité à l'ingénierie sociale, vulnérabilités sociales ou physiologiques que l'attaquant peut utiliser. Ces fragilités sont détectables avec des audits (exemple tests mail phishing). Elles sont réduites par des mécanismes de formations et de sensibilisation, ainsi que dans certains cas des processus d'habilitation ;
- ▶ **Fragilités organisationnelles** : un attaquant peut utiliser des déficiences organisationnelles pour obtenir des éléments pour conduire son attaque (exemple : pas de processus de vérification d'identité lors de demande sensible par téléphone).

Un attaquant utilisera bien entendu l'ensemble de ces fragilités pour conduire sa mission.

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

Ces vulnérabilités sont la conséquence de faiblesses dans la conception, le développement, le déploiement, la mise en œuvre ou l'utilisation d'un composant matériel ou logiciel du système.

Il y a trois grandes classes de faiblesses ou vulnérabilités numériques :

- ▶ **Faillles de configuration** ou de défaut d'usage (utilisation d'un système en dehors de ses zones de fonctionnement stable et maîtrisé)
- ▶ **Faillles Logicielles** : failles de développement, de programmation qui conduisent généralement de l'exploitation de bugs logiciels. Il faut distinguer les logiciels développés de manière dédiée, et les logiciels dits sur étagère. Les dysfonctionnements des logiciels sur étagère (éditeurs logiciels) sont en général corrigés à mesure de leurs découvertes, mais il y a un délai entre le moment de la découverte et la correction,
- ▶ **Faillles de conception** : failles issues de défaut de conception. Ces failles sont souvent liées à des failles protocolaires issues de faille de conception d'un protocole de communication, ou de format de données.

Nous pouvons décomposer les failles dites logicielles, en deux groupes

- ▶ Les failles des logiciels ou **codes sur mesure**, développés dans l'entreprise ou par un tiers mais non édités en tant que logiciel indépendant. Nous pouvons y inclure tous les codes logiciels développés en interne.
- ▶ Les failles logicielles de produits ou codes connus, reconnus souvent dénommées **progiciels** (produits logiciels). On peut aussi y distinguer deux sous classes les logiciels où les sources sont accessibles, et les codes dits fermés où l'utilisateur ne dispose que du code binaire exécutable. Nous verrons que les démarches de recherche de failles dans ces deux types de code sont un peu différentes.

Quand on parle de fragilités, il n'y a pas que les failles de conception ou de développement. Les failles de configuration des systèmes d'information représentent encore une grande



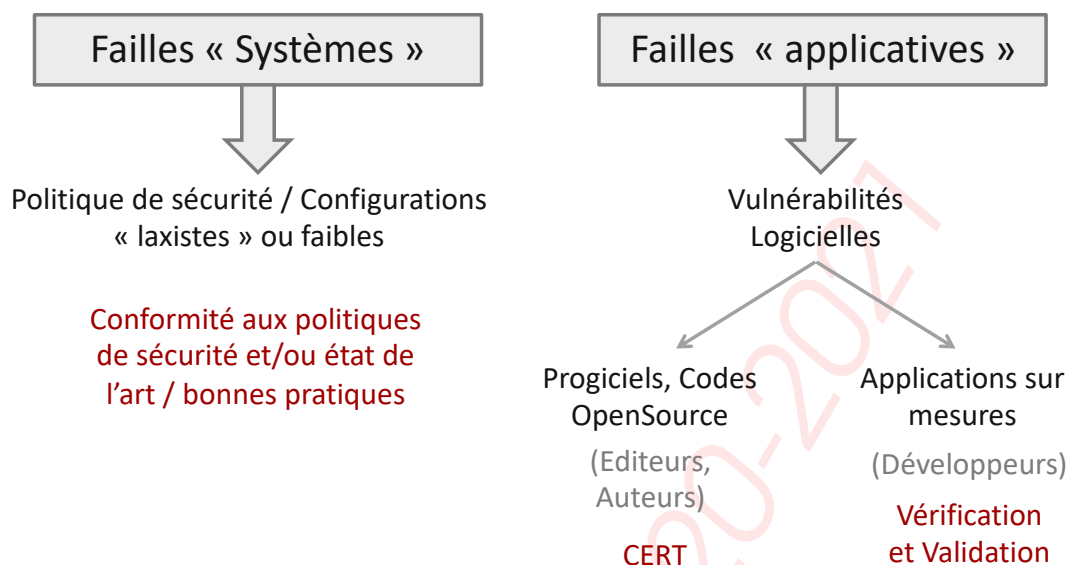


FIGURE 2. Les types de vulnérabilités

partie des fragilités utilisées par les attaques. On trouve encore des administrateurs système qui utilisent dans les outils de filtrage la règle :

AllowAll vs DenyAll

Tout est autorisé sauf ce qui est interdit (**Allow All**) plutôt que de respecter le concept de base de la sécurité tout est interdit (**Deny All**) sauf ce qui est autorisé.

La recherche et la découverte de ces vulnérabilités utilisent donc des outillages un peu différents. On distinguera donc :

- ▶ les failles systèmes et de configuration,
- ▶ les failles dans le développement, dites failles applicatives.

1.4 Exemples de vulnérabilités

A titre d'exemple et d'illustration je vous propose d'examiner rapidement des vulnérabilités techniques : deux failles de conception et une de programmation. Nous ne rentrerons pas dans les détails des vulnérabilités. Ce chapitre a pour objectif de présenter concrètement ce qu'est une vulnérabilité, sur la base d'exemples simples.

Fiche Techno-Blog : Exemples et cas typiques

Les exemples de vulnérabilités tant en développement qu'en configuration et les moyens de les couvrir sont de bons sujets pour une fiche TECHNO dans le cadre des travaux demandés.



La majorité des failles informatiques du domaine du Web et des applications sur mesure est due à une utilisation non prévue de l'applicatif. Un utilisateur peut envoyer une information plus longue que prévue (buffer overflow), ou une valeur non gérée (négative, quand le logiciel attend une valeur positive), ou quand il ajoute des symboles non attendus (des guillemets, caractères spéciaux alors qu'il était prévu seulement des lettres), si les vérifications des données ne sont pas faites correctement, alors le logiciel, le programme ou l'application peut se mettre dans un état qui, dans certains cas, peut être détourné.

1.4.1 Faille type XSS

Si nous prenons par exemple, un code qui affiche une image avec un titre, que ce titre d'image soit saisi par un utilisateur et qu'aucun contrôle ne soit fait. Dans l'application, l'affichage se fait par un code PHP du style :

```
<?php ...
    $image = readimage().".png";
    $title = readtitle();
...
    print '';
...?>
```

et permet de générer le code HTML suivant :

```
<html>...
    
...</html>
```

Un utilisateur malveillant pourrait avoir saisi autre chose qu'un simple titre, et faire en sorte que la variable `$title` puisse contenir une chaîne de caractère un peu particulière. Le pirate aura entré, par exemple, comme titre de sa photo sur ce site un peu faible, une chaîne comme : « un titre de mon image/"><script>...script malveillant...;</script> »

```
<html>...
    <
        script>...scriptmalveillant...;</script>"
...
</html>
```

L'exécution du script javascript malveillant se fera à la lecture de cette page générée. Si cette donnée est stockée sur un serveur, l'action sera effective pour toute les personnes qui consulteront l'image avec son titre piégé par un script malveillant.



1.4.2 Faille type SQL Injection

Nous allons rapidement explorer un grand classique des vulnérabilités sur les applications de sites Web sur Internet : l'injection SQL. Le principe est d'injecter dans une requête SQL (langage d'interrogation de base de données), utilisée dans une application PHP par exemple. Supposons que dans l'application, la requête suivante soit utilisée :

```
SELECT fieldlist
FROM table
WHERE field = ' $EMAIL ';
```

Supposons que la saisie de l'utilisateur, saisisse un email avec une chaîne un peu modifiée (ajout d'un simple « ' » en plus) :

```
SELECT fieldlist
FROM table
WHERE field = ' contact@test.com' ';
```

L'exécution de cette requête va générer une erreur, et en fonction de la gestion des erreurs du code PHP, l'utilisateur pourra apercevoir que cette requête a provoqué une erreur d'exécution. Ceci permet à l'utilisateur de rapidement déterminer que le code est sensible à une attaque par injection SQL. Il peut alors à loisir trouver la meilleure manière de l'exploiter, en entrant un email forgé avec une chaîne plus malicieuse.

La chaîne « OU 'x'='x' » étant toujours VRAI, on pourrait obtenir des informations complètes de certaines tables. Bien entendu, l'usage de vulnérabilité SQL injection n'est généralement pas trivial, mais avec un peu d'habitude, il est possible de construire des attaques sophistiquées sur des codes vulnérables.

```
SELECT fieldlist
FROM table
WHERE field = ' somebody' OU 'x' = 'x' ';
```

1.4.3 Vulnérabilités WEB

Vous trouverez sur le site Open Web Application Security Project [↗](https://www.owasp.org)⁴, le top TEN des vulnérabilités découvertes sur les sites WEB et pour ceux qui souhaitent creuser un peu plus, il existe de nombreux sites présentant en détail des vulnérabilités et des manières de les exploiter (à des fins pédagogiques!). Le site de Pixis (Hackndo) [↗](https://beta.hackndo.com)⁵, par exemple, vous donne quelques partages particuliers d'un Ethical hacker.

4. <https://www.owasp.org>

5. <https://beta.hackndo.com>



1.5 Failles de programmation

1.5.1 Exemple : SMB etherblue

On peut trouver des vulnérabilités dans des produits et services très connus, et déployés depuis très longtemps. Parmi ces vulnérabilités les plus célèbres (voir figure 3), la classe de vulnérabilités du protocole SMB en version 1, est celle qui continue encore à faire des victimes. Le protocole SMB (Server Message Block) est un protocole permettant le partage de ressources (fichiers et imprimantes) sur des réseaux locaux avec des PC sous Windows. Sa version 1 du protocole SMB, vulnérable à la faille EternalBlue.

Windows : une faille 0-day révélée dans SMB, le correctif ...

<https://www.nextinpact.com/news/103173-windows-faille-0-day-revele...>

6 févr. 2017 - Une faille 0-day existe dans Windows, plus spécialement dans la manière dont le système gère le trafic SMB. Un prototype d'exploitation est déjà en circulation, ... Analyses de la rédaction. ...

Faible critique dans le protocole SMB de Windows ...

cybersecurite.over-blog.com/article-faille-critique-dans-le-protocole-smb...

17 févr. 2017 - La société de sécurité Vupen émet une alerte jugée comme "critique" sur une vulnérabilité concernant le système d'exploitation Windows pour ...

MS17-010 - Security Update for Microsoft Windows SMB ...

<https://www.sophos.com/threat-analyses/vulnerabilities/VET-001035>

14 mars 2017 - MS17-010 - Security Update for Microsoft Windows SMB Server. Pour plus ... de test des SophosLabs. Faibles connues, Aucune faille connue.

Analyse des attaques des ransomware Wannacry et Jaff ...

<https://www.vadesecure.com/analyse-attaques-ransomware-wannacry-jaff>

15 mai 2017 - Ce ransomware se propage au travers d'une faille du protocole de partage SMB v1 (Server Message Block) non patchée au moment de ...

FIGURE 3. Tempo faille SMB - google

Dans la base de données du système CVE, on retrouve l'identifiant de cette vulnérabilité : **CVE-2017-0144**.

1.5.2 Exemple : programmation erronée

Le célèbre débordement de pile, ou BUFFER OVERFLOW, fait partie des exemples d'erreur de programmation, en particulier avec des langages permissifs, pouvant conduire à des situations exploitables par des codes malveillants.

Si nous prenons l'exemple simple d'une fonction de copie de mémoire par indexation de tableau en C suivante :

```
void *memcpy(char *dest; char *src, size_t *n)
{ /* copie de bloc de memoire */
// size_t : taille du tableau
// *src : pointeur source
// *dest : pointeur destination
for (size_t i=0; i<n; i++)
```




```
dst[i] = src[i];  
}
```

Un nombre négatif (et donc incorrect en principe pour copier une zone de mémoire dans une autre) peut être passé en valeur dans 'n', et par la suite est interprété comme un grand nombre à cause du complément à 2 et de la façon dont les nombres négatifs sont codés. Ceci qui provoque un débordement de mémoire et un plantage si ce débordement touche une zone de mémoire virtuelle non liée à une zone de mémoire physique.

1.6 Vulnérabilités et configuration

Un des plus grands classiques de vulnérabilité système concerne les défauts de configuration, en particulier les défauts de configuration des équipements et systèmes dans un environnement réseau.

L'outil le plus classique et accessible est NMAP. Cet outil est conçu pour détecter les ports ouverts, identifier les services hébergés et obtenir des informations sur le système de l'ordinateur distant. La technique de scan de port est utilisée par les administrateurs des systèmes informatiques pour contrôler la sécurité des serveurs de leurs réseaux. La même technique est aussi utilisée par les acteurs malveillants pour tenter de trouver des failles dans des systèmes informatiques. De nos jours un balayage de ports (port scan ou portscan en anglais) effectué sur un système tiers est généralement considéré comme une agression, car il préface une intrusion. Il est donc recommandé de l'utiliser de manière responsable lorsqu'on utilise ce type d'outils dans son entreprise. En effet –, scan de ports est une des activités considérées comme suspectes par un système de détection d'intrusion.

1.7 Vulnérabilités et exploits

Il arrive que la procédure d'exploitation d'une faille d'un logiciel soit documentée et utilisable soit sous la forme d'un code logiciel et/ou de procédure descriptive détaillée appelée « exploit ». Ces exploits ne sont pas systématiquement publiés.

1.8 Vulnérabilités et divulgation

La divulgation publique des vulnérabilités est soumise à un modèle de divulgation de vulnérabilité dans lequel une vulnérabilité ou un problème est révélé uniquement après une période permettant à la vulnérabilité ou au problème d'être corrigé. Cette période distingue le modèle de la divulgation complète.

Tout fournisseur de logiciels de sécurité, de services et de recherches de vulnérabilité, se doit de prendre des précautions vis à vis de vulnérabilités découvertes, en particulier les délais de publication. On parle généralement de *Vulnerability Disclosure Policy*.

En effet, développeurs de matériel et de logiciels ont souvent besoin de temps et de ressources pour corriger ces vulnérabilités.

Dans certains cas, lorsque la découverte n'a pas été faite via une recherche comman-



ditée (Audit, Pentest, BugBounty), la communauté sécurité et les scientifiques estiment qu'il est de leur responsabilité sociale de sensibiliser le public aux vulnérabilités ayant un impact important en les publiant. Cacher ces problèmes pourrait créer un faux sentiment de sécurité. Pour éviter cela, les parties impliquées unissent leurs forces et s'accordent sur un délai pour réparer la vulnérabilité et prévenir tout dommage futur. En fonction de l'impact potentiel de la vulnérabilité, du temps requis pour qu'un correctif d'urgence ou une solution de contournement soit développé et appliqué, ainsi que d'autres facteurs, cette période peut varier de quelques jours à plusieurs mois.

Par ailleurs, la confidentialité des découvertes est généralement requise lors des audits. Le commanditaire et l'expert signent un accord dénommé *Vulnerability Non Disclosure Agreement*, qui permet de s'assurer que la publication des vulnérabilités restera à la main du commanditaire.

Dans un mode public avec mode de divulgation de vulnérabilités ouvertes les experts en sécurité s'attendent à être indemnisés financièrement, mais avec le risque que signaler ces vulnérabilités au fournisseur avec l'exigence d'une indemnisation soit considéré comme une extorsion.

Un marché des vulnérabilités s'est développé, on peut citer en particulier la société Zerodium ⁶, ou les sociétés de BugBounty, mais la commercialisation des vulnérabilités reste un sujet très controversé lié au concept de divulgation des vulnérabilités. C'est normalement dans le rôle d'un CERT d'assurer cette coordination des divulgations.

Fiche Techno-Blog : Le marché des vulnérabilités

Le marché des failles de sécurité est un marché particulier dans lequel des hackers de toute nature trouvent le moyen de financer leurs activités de R&D et de hacking. Les grands éditeurs commerciaux et libres y trouvent leur compte. C'est un sujet intéressant pour une fiche TECHNO.

1.9 CVE, CVSS et CWE

1.10 Common Vulnerabilities and Exposure (CVE)

De nombreuses vulnérabilités sont découvertes chaque jour dans des produits et logiciels. Les informations techniques sur ces vulnérabilités permet de les détecter, et de les caractériser. Il était important dans le monde des technologies de l'information qu'elles puissent être identifiées et décrites de manière unique, et que ces caractérisations soient accessibles à tous.

L'objectif fondamental de la création du CVE est de constituer un dictionnaire qui recense toutes les failles avec une description succincte de la vulnérabilité, ainsi qu'un ensemble de liens que les utilisateurs peuvent consulter pour plus d'informations. Cette base est proposée pour consultation et reste maintenue par le Mitre Corporation. Cet organisme à

6. <https://zerodium.com>



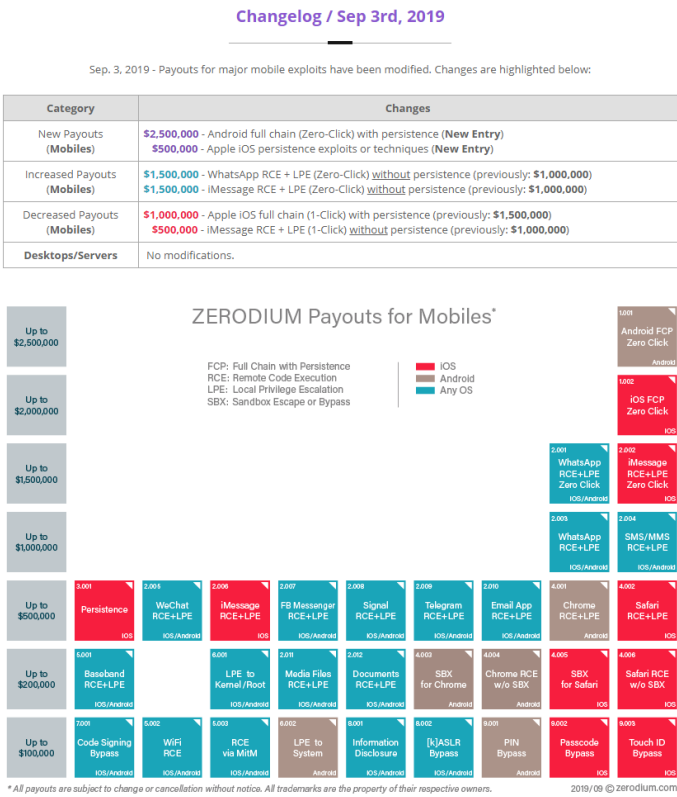


FIGURE 4. Le marché des failles mobiles avec Zerodium

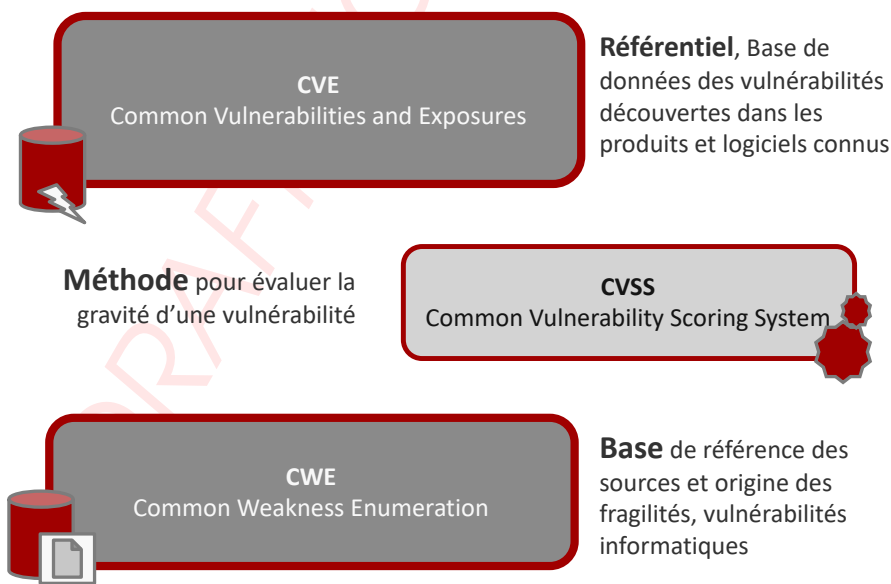


FIGURE 5. Quelques concepts de gestion sur les vulnérabilités



but non lucratif américaine a pour objectif de travailler dans des domaines technologiques comme l'ingénierie des systèmes, les technologies de l'information, la sécurité.

Common Vulnerabilities and Exposures ou CVE est une base de données (Dictionnaire) des informations publiques relatives aux vulnérabilités de sécurité. Le dictionnaire est maintenu par l'organisme MITRE. Les identifiants CVE sont des références de la forme CVE-AAAA-NNNN

Pour consulter les CVE, il suffit de se rendre sur [CVE.mitre.org](https://www.cve.mitre.org) ⁷


Le système CVE permet de recenser toutes les failles et les menaces liées à la sécurité des systèmes d'information avec un identifiant unique attribué à chaque faille.

On trouve ainsi dans cette base par exemple :

- ▶ l'identifiant de l'une des vulnérabilités qui a permis une attaque massive via le rançongiciel **Wannacry** : CVE-2017-0144, faille dans le protocole SMB découverte en 2017 et la 144^{ième} faille découverte de l'année.
- ▶ ou **Heartbleed** l'une des failles les plus importantes des années 2010 : CVE-2014-0160 présente dans la couche logicielle open source OpenSSL. OpenSSL est une librairie avec deux bibliothèques, libcrypto et libssl, respectivement une implémentation des algorithmes cryptographiques et du protocole de communication SSL/TLS fortement utilisé par internet.

1.11 Common Vulnerability Scoring System (CVSS)

Bien entendu, disposer d'un identifiant d'une vulnérabilité est important, mais un gestionnaire de sécurité dans l'entreprise, doit aussi disposer d'éléments pour juger de la gravité de cette vulnérabilité.

Le *Common Vulnerability Scoring System (CVSS)* à sa version 3 issu des travaux du FIRST, Forum of Incident Response and Security Teams ⁸, est un cadre méthodologique permettant d'évaluer en particulier la criticité d'une vulnérabilité.

C'est un système permettant de calculer une note évaluant la criticité d'une vulnérabilité, et de construire une chaîne de caractères (un vecteur) présentant les caractéristiques de cette vulnérabilité, et les critères utilisés pour ce calcul.

Les notes et vecteurs CVSS sont toujours le résultat de trois groupes de critères d'évaluation (« Base », « Temporal » et « Environnemental ») ayant chacun leur note ainsi que leur vecteur :

- ▶ Le groupe des critères de « **Base** » évalue l'impact maximum théorique de la vulnérabilité.
- ▶ Le groupe des critères « **Temporel** » pondère le groupe « Basic » en prenant en compte l'évolution dans le temps de la menace liée à la vulnérabilité (par exemple, l'existence d'un programme d'exploitation ou d'un correctif).

7. <https://www.cve.mitre.org>

8. <https://www.first.org/cvss/>



- ▶ Le groupe des critères « **Environnemental** » pondère le groupe « Temporel » en prenant en compte les caractéristiques de la vulnérabilité pour un Système d'Information donné.


La richesse du modèle apporte une complexité dans sa lecture rapide, toutefois globalement, on peut lire un score CVSS en terme de criticité avec la grille de lecture suivante :

- ▶ Un score de 0 à 3.9 correspond à une criticité basse
- ▶ Un score de 4 à 6.9 correspond à une criticité moyenne
- ▶ Un score de 7 à 10 correspond à une criticité haute

Un autre exemple que je vous engage à explorer pour bien comprendre le fonctionnement CVE sont les vulnérabilités :

- ▶ CVE-2020-1023, CVE-2020-1024, and CVE-2020-1102 - SharePoint Remote Code Execution Vulnerability
- ▶ CVE-2020-1067 - Windows OS Remote Code Execution Vulnerability
- ▶ CVE-2020-1058 (VBScript), CVE-2020-1060 (VBScript), CVE-2020-1064 (Trident=>I.E.) – Internet Explorer Remote Code Execution Vulnerability (which could be used during Web Browsing)
- ▶ CVE-2020-1096 - Microsoft Edge PDF Remote Code Execution Vulnerability

1.12 Common Weakness Enumeration (CWE)

L'énumération des faiblesses ordinaires ou classiques c'est ainsi qu'il faudrait traduire CWE publié par le MITRE ⁹ est un site qui liste par ailleurs le top des erreurs de programmation dangereuses et les plus fréquentes. En effet, malheureusement les développeurs font souvent les mêmes erreurs, et malgré une évolution des formations en développement, les développeurs juniors continuent à reproduire les mêmes erreurs de conception et de programmation qui conduisent à des failles logicielles. La plupart des vulnérabilités applicatives viennent en particulier de quelques erreurs bien connues, qui reviennent régulièrement et pour lesquelles une grande partie des attaques existantes se basent. C'est le but de la CWE (Common weakness Enumeration) qui est de recenser les erreurs de programmation commises. On y retrouve des grands classiques, comme la validation des champs d'un formulaire, la célèbre injection SQL, les problèmes de gestion du système, les contrôles d'accès mal gérés, les tests réalisés par le client plutôt que par le serveur... Le but du top 25 est d'attirer l'attention des programmeurs sur leurs propres erreurs les plus courantes, mais également de faire réfléchir les formateurs : trop souvent, ces problèmes courants sont oubliés des cours de programmation et de sécurité. Après une brève présentation de chaque problème, la CWE propose des principes généraux pour l'éviter ; le tout est clarifié autant que possible et devrait être compréhensible avec un peu d'effort par la plupart des développeurs. On explorera un peu plus ces éléments dans le chapitre sur la sécurité applicative.

En 2019, les 3 premières faiblesses ordinaires ont été :

9. <http://cwe.mitre.org>



- ▶ CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer à 75.56%
- ▶ CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') à 45.69%
- ▶ CWE-20 Improper Input Validation à 43.61%

1.13 Les services de veille en vulnérabilités

Vous trouverez quelques éléments sur les CERTs (Computer Emergency Response Team) dans le chapitre sur les vulnérabilités, toutefois le périmètre de fonctions et services des CERTs s'est rapidement élargie ces dernières années. Au delà de la diffusion et alertes sur des vulnérabilités, ils couvrent maintenant avec précision les menaces (analyse et alerte sur codes malveillants, ...) et les incidents. Les CERTs restent les acteurs principaux de cette veille et capacité d'alerte. Nous trouvons toutefois de nombreux services de veille en vulnérabilités qui ne sont pas des CERTs, mais qui offrent des services dédiés à des typologies de produits, ou des secteurs

1.13.1 Les CERT de l'ANSSI

le Computer Emergency Response Team (CERT) gouvernemental de l'Agence Nationale de la Sécurité des systèmes d'information (ANSSI) (CERT GOUV FR ¹⁰) publie régulièrement plusieurs types d'information :

- ▶ Alertes de sécurité ;
- ▶ Rapports sur des menaces et incidents ;
- ▶ Avis de sécurité Indicateur de compromission ;
- ▶ Bulletins et notes d'information.

1.13.2 Les CERTs commerciaux

Fiche Techno-Blog : Fiche TECHNO

Les CERTs commerciaux est un bon sujet d'exploration des sociétés qui délivrent des services de veille. Vous trouverez les Certs en France ^a sur le site de l'ANSSI. Excellent sujet pour une fiche TECHNO.

a. <https://www.ssi.gouv.fr/agence/cybersécurité/ssi-en-france/les-cert-francais/>

1.13.3 La relation avec un CSIRT Interne

Une méthodologie efficace de gestion des vulnérabilités comprend une équipe d'intervention en cas d'incident de sécurité informatique (CSIRT). Le CSIRT est responsable de la publication des avis de sécurité, de la tenue d'informations régulières pour échanger sur

10. <https://www.cert.ssi.gouv.fr>



les activités malveillantes et des dernières attaques du jour zéro, de la simplification et de la diffusion des alertes de sécurité et de l'élaboration de directives compréhensibles et efficaces en matière de réaction aux incidents pour tous les salariés. De cette manière, chacun sera en mesure de réagir aux indicateurs de compromis potentiels conformément aux pratiques recommandées par l'équipe CSIRT.

1.14 les agences de notation

⚙️ chapitre en cours de rédaction, DRAFT non publiable ⚙️

2. GERER les fragilités

Dans le paysage numérique de plus en plus complexe, nous sommes exposés à des terminologies variées souvent soutenues par les modes du moment. Les termes «analyse de vulnérabilités», «évaluation des vulnérabilités» et «gestion des vulnérabilités» sont souvent utilisés et restent une source de confusion pour nombre d'entre nous. Pour nous assurer de se concentrer sur les tactiques les plus efficaces pour gérer les vulnérabilités, nous donnerons les principales différences entre l'évaluation des vulnérabilités et la gestion des vulnérabilités. Mais en posant comme principe que l'important est d'agir quand sont identifiées des failles dans un système. La gestion des vulnérabilités est donc un processus de gestion des risques associés à la présence de vulnérabilités qui se base sur la recherche de celles-ci, l'évaluation de leur impact et qui pilote le calendrier d'application des correctifs disponibles.

- ▶ La gestion des vulnérabilités (**Vulnerability Management**) est un processus continu servant à identifier, classer, corriger et réduire les vulnérabilités, en particulier dans les logiciels. La gestion des vulnérabilités fait partie intégrante des processus de gestion de la cybersécurité dans l'entreprise. Contrairement au projet d'évaluation ponctuelle des vulnérabilités, une stratégie de gestion des vulnérabilités fait référence à un processus ou programme complet et continu qui vise à gérer les vulnérabilités d'une organisation de manière globale et continue. Nous avons rassemblé quelques caractéristiques et éléments clés d'une approche standard de la gestion des vulnérabilités. La gestion des vulnérabilités comprend aussi le processus par lequel les risques associés à ces vulnérabilités sont évalués. Cette évaluation conduit à corriger les vulnérabilités et éliminer le risque ou une acceptation formelle du risque par la gestion d'une organisation (par exemple, au cas où l'impact d'une attaque serait faible ou la le coût de la correction ne dépasse pas les dommages éventuels pour l'organisation).
- ▶ Il est souvent confondu avec l'évaluation des vulnérabilités (**Vulnerability Assessment**), dont l'objectif est de rechercher les fragilités d'un système ou d'une entreprise. Ces vulnérabilités connues sont recherchées sur le système. Une évaluation de vulnérabilité n'est pas une analyse, c'est un projet ponctuel avec une date de début et une date de fin définies. En règle générale, un consultant externe en sécurité de l'information examine votre environnement d'entreprise et identifie diverses



vulnérabilités potentiellement exploitables auxquelles vous êtes exposés dans un rapport détaillé. Le rapport répertoriera non seulement les vulnérabilités identifiées, mais fournira également des recommandations concrètes pour la résolution. Une fois le rapport final préparé, l'évaluation de la vulnérabilité est terminée. Malgré le fait que les deux sont liés, il existe une différence importante entre les deux. La recherche de vulnérabilités consiste à utiliser par exemple un programme informatique pour identifier les vulnérabilités dans réseaux, infrastructure informatique ou applications. La gestion de la vulnérabilité est le processus entourant ce scan de vulnérabilités, prenant également en compte d'autres aspects tels que acceptation des risques, remédiation, etc. On verra en outre que le scan de vulnérabilités n'est qu'une sous partie de l'évaluation des vulnérabilités.

- L'analyse des vulnérabilités est un processus de recherche de ces fragilités et des scénario qui vont permettre de les exploiter. Les tests d'intrusion sont un exemple de cette dynamique d'analyse des fragilités afin d'en définir un scénario permettant d'atteindre l'objectif que l'attaquant s'est assigné.

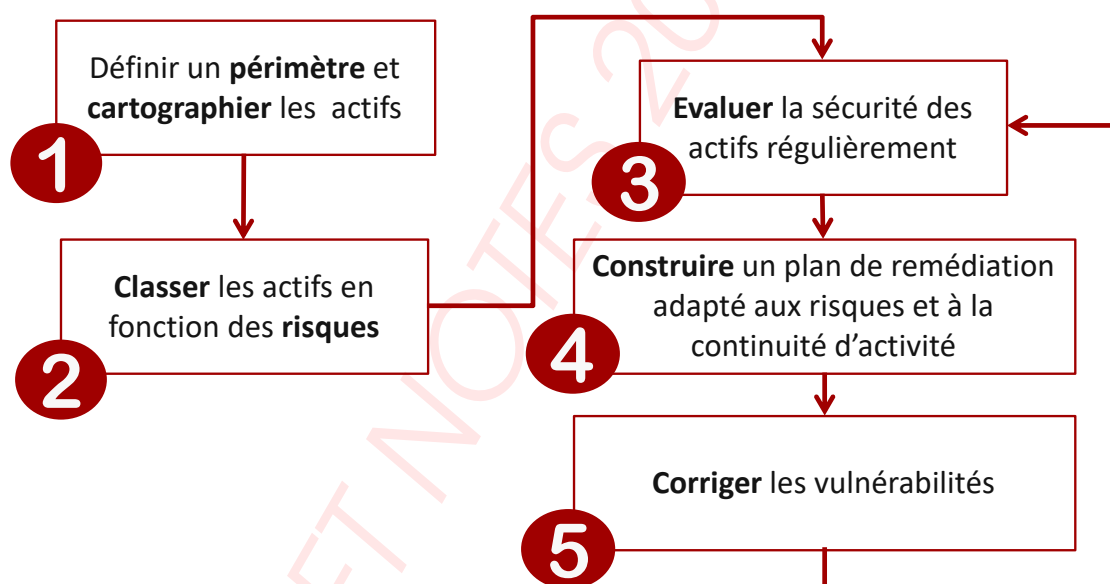


FIGURE 6. La gestion des vulnérabilités

2.1 Processus de gestion des vulnérabilités

La gestion des vulnérabilités est un processus continu. Elle apparaît en toile de fond du cycle de vie du Maintien en Condition de sécurité :

- Cartographier, cataloguer l'environnement ;
- Identifier les fragilités et les menaces ;
- Corriger, remédier, améliorer la protection et la défense ;



- ▶ Mesurer et suivre l'efficacité des mesures déployées.

2.1.1 ISO 27001

Un chapitre de la norme parle de Veille de la vulnérabilité, que nous pouvons classer dans le domaine de la gestion des vulnérabilités et donne des éléments méthodologiques :

- ▶ 1. **DÉCOUVRIR** : Catalogage de l'existant, des actifs, des ressources du système d'information.
- ▶ 2. **PRIORISER** : Classifier et attribuer des valeurs quantifiables aux ressources, les hiérarchiser.
- ▶ 3. **ÉVALUER** : Identifier les vulnérabilités ou les menaces potentielles sur chaque ressource.
- ▶ 4. **SIGNALER** : Signaler, publier les vulnérabilités découvertes.
- ▶ 5. **CORRIGER** : Éliminer les vulnérabilités les plus sérieuses des ressources les plus importantes.
- ▶ 6. **VÉRIFIER** : S'assurer que la vulnérabilité a bien été traitée.

2.1.2 Fenêtre d'exposition

Dans un monde idéal où en temps réel, on accéderait à l'apparition d'une vulnérabilité, ou on pourrait la détecter sur son SI, et patcher avec le correctif publié par l'éditeur, la gestion des vulnérabilités se cantonnerait à constater la fenêtre d'exposition générée par le temps nécessaire à l'éditeur pour publier un correctif immédiatement déployé.

L'unique décision pour un RSSI serait une gestion du risque simple liée à la décision de déconnecter ou pas un élément vulnérable du SI pendant la fenêtre d'exposition. Tous les correctifs disponibles seraient appliqués dès qu'ils sont disponibles.

Même si l'information peut être accessible en temps réel, la détection de la présence d'une vulnérabilité sur le SI dépend de la fréquence des « scans » ou des audits de l'organisation, ou la connaissance parfaite du SI via une CMDB à jour. Ainsi, la fenêtre d'exposition apparaît naturellement à plusieurs niveaux quelle que soit la réactivité du service de gestion des vulnérabilités.

L'application d'un correctif nécessite une fenêtre de maintenance et un arrêt du service. Au mieux, ces fenêtres de maintenance sont prévues et planifiées, au pire, l'application de correctifs est interdite hors mode projet (notamment sur certains réseaux opérationnels et industriels). Par ailleurs, la planification et l'application d'un correctif, même dans le meilleur des cas, est extrêmement chronophage pour les équipes techniques.

Tous les correctifs pour une même vulnérabilité ne peuvent pas être déployés au même rythme sur tous les systèmes. La gestion des vulnérabilités se structure donc autour de la gestion des priorités basée sur les risques contextuels évalués sur la base



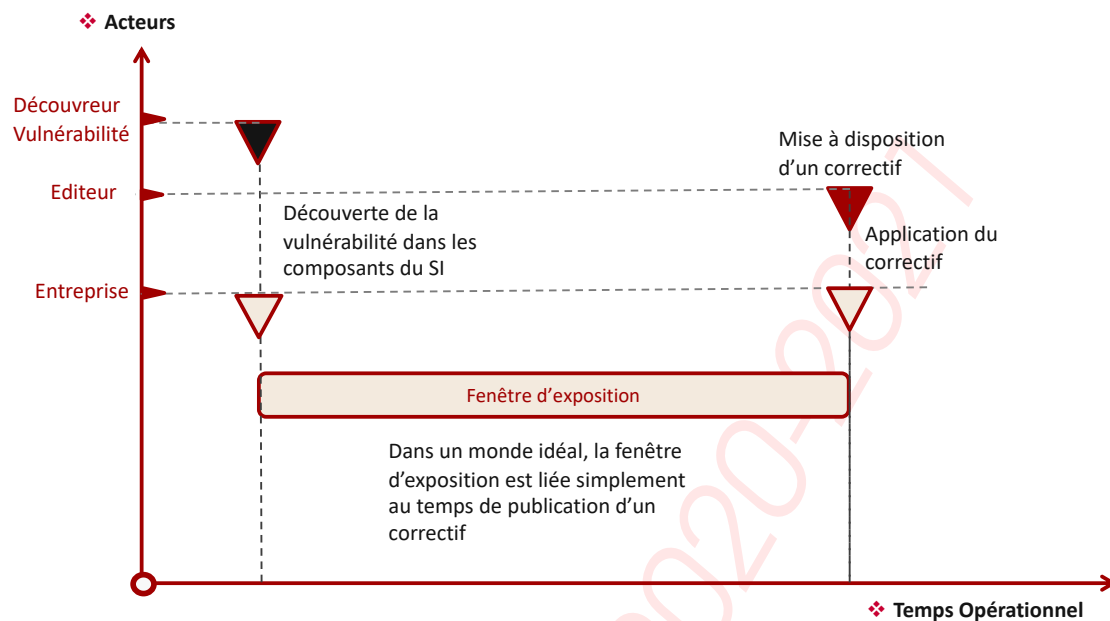


FIGURE 7. Fenêtre d'exposition idéale

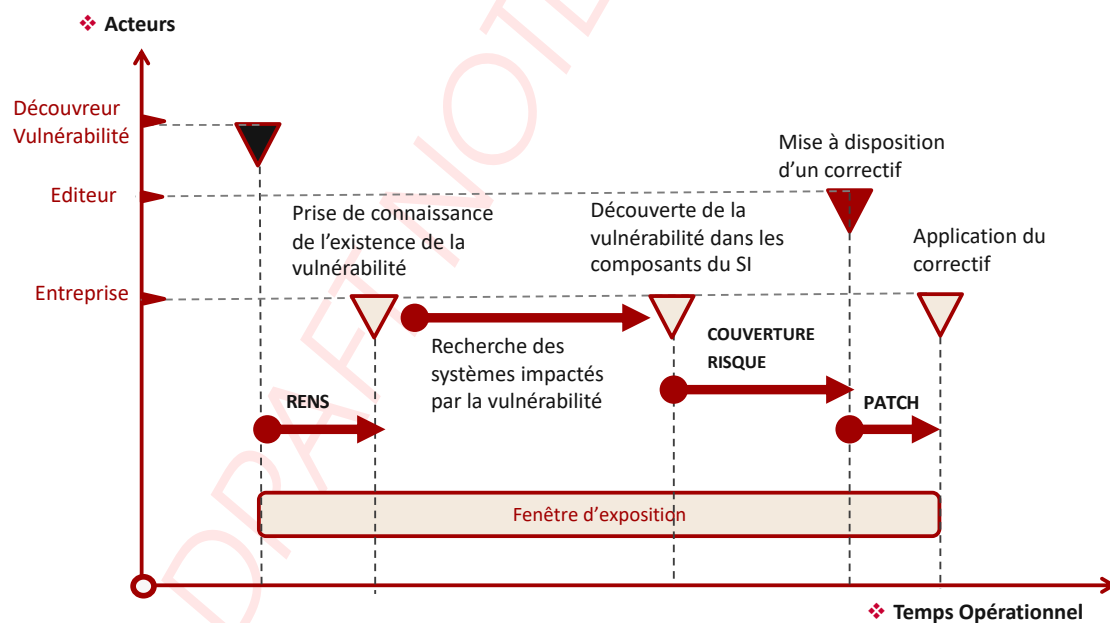


FIGURE 8. Fenêtre d'exposition



- ▶ du niveau de criticité de la vulnérabilité (c'est à dire ce que son exploit permet de réaliser ou d'obtenir – de nombreux services de veille s'appuient sur le Framework CVSS pour classer les vulnérabilités)
- ▶ du niveau de criticité pour l'organisation des systèmes et services avérés vulnérables.

Il est important de disposer d'un accès à un service de d'information (renseignement) sur les vulnérabilités et sur les Menaces (cf. CERT commercial) afin de collecter et d'organiser le renseignement extérieur et de le croiser avec le renseignement issu de l'interne (détections, incidents, ...) afin d'identifier l'information prioritaire, de l'enrichir et de la diffuser vers les opérations de sécurité.

Un tel service peut être en mesure d'associer une vulnérabilité à un adversaire, à une phase d'attaque, et à une technique particulière employée dans cette phase d'attaque (Cf. Framework MITRE ATTACK facilite notamment un tel niveau de précision et d'analyse).

Il est ainsi possible de mettre en exergue les vulnérabilités exploitées par les adversaires et les campagnes jugées prioritaires ciblant l'organisation. Cette information sur l'adversaire peut être alors disséminée à des équipes de gestion des vulnérabilités et utilisée pour prioriser l'application des correctifs.

2.1.3 Processus d'analyse/recherche des vulnérabilités

Avant de se lancer dans la dynamique classique des audits de sécurité (qui permettent de trouver des vulnérabilités), on peut positionner des mécanismes de recherches de vulnérabilités dans le cycle de vie d'un système sous la forme des différentes étapes des cycles V et V (Vérification et validation) d'un projet.

- ▶ **Phase de conception** : recherche des défauts et fragilités de conception avec des techniques d'analyse de risque, de revue de conception avec des analyses de menaces
- ▶ **Phase de développement** : pendant la phase de développement il existe de nombreux outils d'audit de code statique qui offre l'assistance aux développeurs pour éviter les erreurs les plus classiques,
- ▶ **Phase de validation** : dans cette phase, il est possible d'utiliser des techniques et méthodologies classiques d'audit de sécurité (Pentest, analyse de code, ...)
- ▶ **Phase de vérification opérationnelle** en Pré-Production ou en production : dans cette phase c'est généralement de l'audit dynamique de type scan de vulnérabilité et tests d'intrusion.

Contrairement à l'évaluation des vulnérabilités, un programme complet de gestion des vulnérabilités n'a pas de date de début ni de fin définie, mais constitue un processus continue.

2.1.4 Processus d'évaluation des vulnérabilités (Vulnerability Assessment)

L'évaluation permet de définir l'impact d'une vulnérabilité sur les risques courus par l'entreprise dans une dynamique d'audit ponctuelle ou récurrente ?..



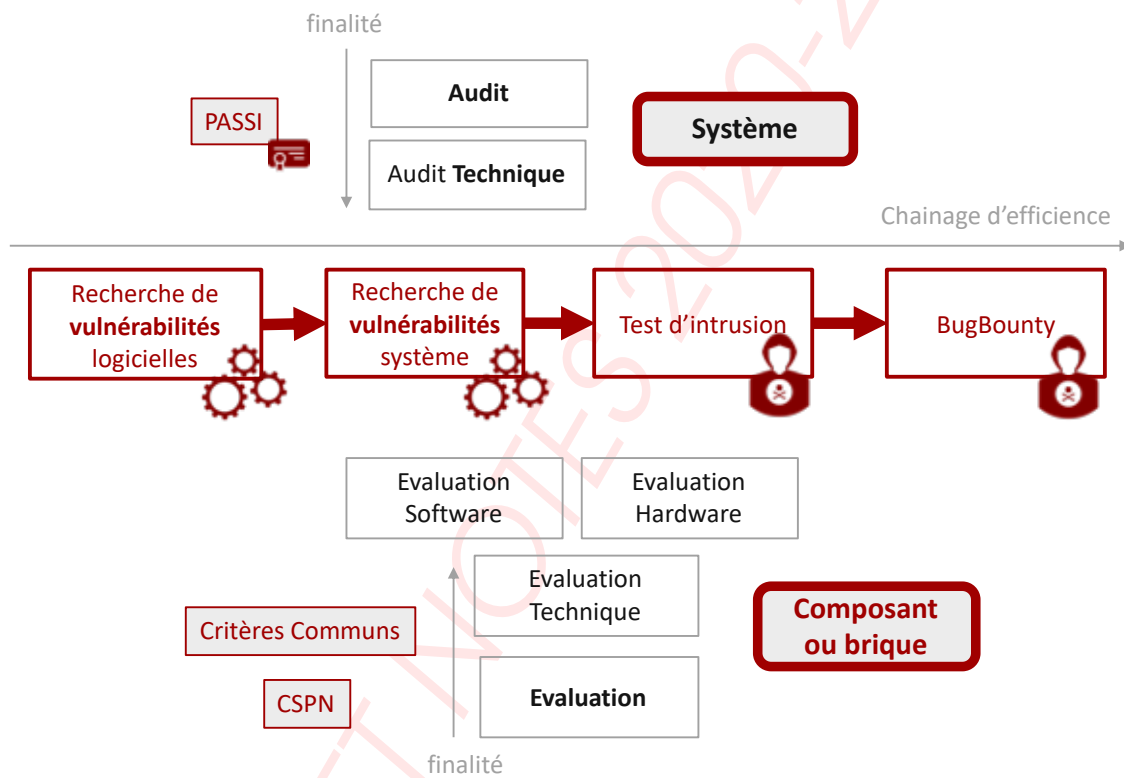


FIGURE 9. Rechercher ses vulnérabilités



2.2 Audit sécurité des vulnérabilités

Les audits de sécurité se conduisent souvent sur une base d'audit de vulnérabilités suivi de l'exploitation de ces vulnérabilités ou fragilités pour construire des scénarios plausibles et de caractériser des risques à forte probabilité ou fort impact. On distingue cependant :

- les scans de vulnérabilités, permettant de manière automatisée à rechercher les vulnérabilités sur un système donné, et qui se base sur des bases de vulnérabilités connues
- les audits techniques et pentests, qui peuvent se baser en premier lieu sur des scans pour identifier les vulnérabilités connues, mais qui travaillent aussi sur la recherche de vulnérabilités.

2.2.1 Scan de Vulnérabilités du système

Les vulnérabilités peuvent être découvertes à l'aide d'un scanner de vulnérabilités, qui analyse un système informatique à la recherche de vulnérabilités connues, telles que les ports ouverts, les configurations logicielles non sécurisées et la vulnérabilité aux infections par logiciels malveillants. Des tests de fuzz peuvent permettre de détecter des vulnérabilités inconnues, telles que le jour zéro, permettant d'identifier certains types de vulnérabilités, telles qu'un débordement de mémoire tampon avec des cas de tests pertinents. Une telle analyse peut être facilitée par l'automatisation des tests.

2.2.2 Scan de Vulnérabilités logicielles

La correction des vulnérabilités peut impliquer de différentes manières l'installation d'un correctif, une modification de la stratégie de sécurité du réseau, la reconfiguration du logiciel ou la formation des utilisateurs à l'ingénierie sociale.

2.3 La gestion des correctifs

Le patch management est un processus permettant cette gestion des correctifs de sécurité et leur déploiement en entreprise.

En anglais, ce patch management consiste à industrialiser les processus de détection, d'analyse et de déploiement des mises à jours de sécurité logicielles. En effet, lorsqu'un éditeur publie un nouveau patch de sécurité relatif à son produit, ses clients ne sont pas toujours en mesure d'évaluer l'importance de ce dernier ni les risques de son installation. Les solutions de gestion des correctifs proposent alors de stocker localement les correctifs sur un serveur du client, puis d'évaluer l'impact de celui-ci avant éventuellement de le tester puis de l'installer.

2.3.1 De l'outillage

La force d'un outil de gestion des correctifs vient d'abord de sa base de données. C'est elle qui référence le parc informatique existant et conserve l'historique de l'installation des patches. Elle permet à tout instant de revenir en arrière après installation d'un correctif ou de déterminer précisément où doivent être installés les patches de sécurité. Par cette base de



connaissance, les logiciels de gestion de correctifs facilitent la détection d'incompatibilités logicielles ou matérielles avec un correctif de sécurité car chaque patch donne lieu à une validation fonctionnelle. Les correctifs installés ceux des éditeurs et dépendent donc de leurs délais de publication.

2.4 Les audits

Les audits de vulnérabilités s'inscrivent généralement dans des processus de sécurité d'entreprise ou de projets

Les audits peuvent être de natures différentes :

- ▶ Audit Organisationnel : pour découvrir les fragilités organisationnelles et humaines
- ▶ Audit technique : pour découvrir et analyser les fragilités

On peut avoir besoin de ces audits pour des enjeux différents :

- ▶ Audit de conformité
- ▶ Audit de vérification et de validation
- ▶ Audit de contrôle et d'inspection

Avec une dynamique d'audit :

- ▶ Audits ponctuels et campagnes d'audit
- ▶ Audit continu

Fiche Techno-Blog : Les audits

Les sujets autour des techniques, méthodologies d'audit informatique liés à la sécurité sont une bonne source de réflexion pour une fiche TECHNO.

3. RECHERCHER des vulnérabilités

3.1 Les tests d'intrusion

Il y existe de nombreuses activités d'évaluation des vulnérabilités. La littérature identifie par ailleurs des noms de métiers ou de compétences d'activité.

Un des cadres les plus courant pour conduire des audits est l'audit technique qui comprend les tests d'intrusion : PENTEST.

3.2 Généralités

Le terme PENTEST est devenu tellement courant, que l'on oublie quelques fois qu'il est l'abréviation de : « Penetration Testing », qui veut littéralement dire tests d'intrusion. L'expression « test de pénétration » est parfois rencontrée, mais les professionnels du PENTEST n'apprécient pas trop cette expression.



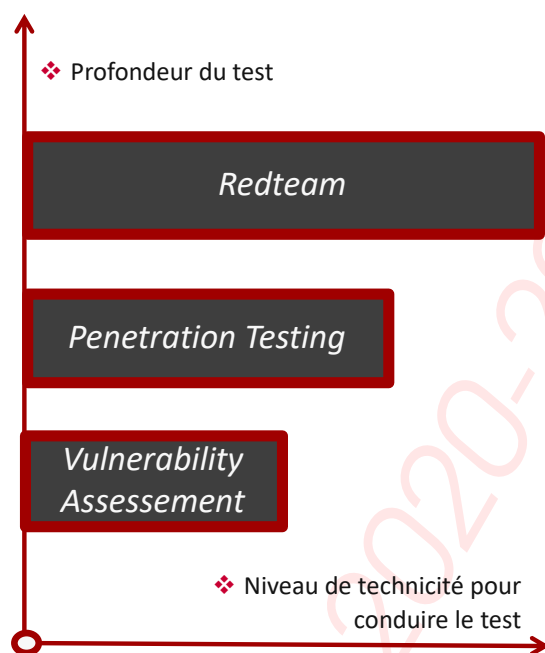


FIGURE 10. Les types de tests de vulnérabilités

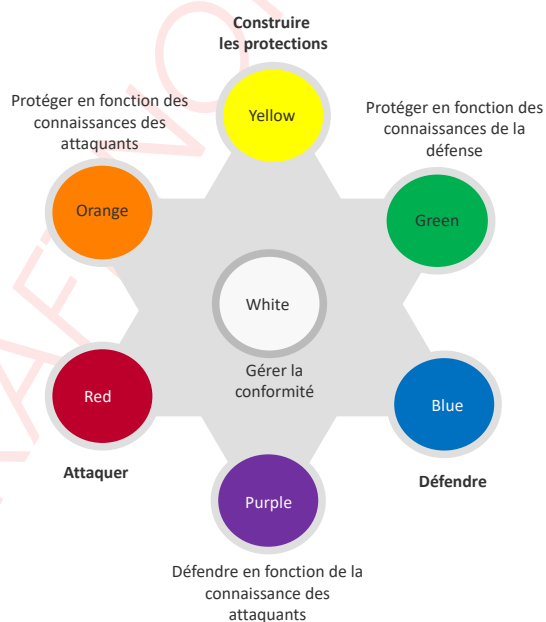


FIGURE 11. Les branches du test



Il est toujours un peu complexe de catégoriser l'activité de pentests. Il aura assez rapidement des détracteurs pour soulever le fait que la catégorie n'est pas la bonne, qu'elle n'est pas représentative du métier. Je vais donc faire rentrer cette activité dans plusieurs catégories (métier de l'audit, métiers d'expertise, métiers du tests). Dans le cadre de ce cours, je propose de relier cette activité métier comme un des outils du processus de gestion des vulnérabilités (*Vulnerability Management*). Mais il est plus courant de classer les activités de PENTESTS dans les activités d'audit.

3.3 Le métier de Pentesteur

Le métier du PENTEST est lié aux métiers techniques de l'informatique et télécom. Les origines des Pentesteur sont très variées. Ce sont des métiers qu'il est possible d'exercer avec différents niveaux de formation.

3.3.1 Ethical Hackers

Etre Ethical Hacker fait partie du mythe de la cybersécurité.

3.3.2 Peut-on faire confiance à des pentesteurs ?

Parmi les grandes questions que se posent les « commanditaires » de tests d'intrusion se trouve celle de la confiance. En effet, le principe des tests d'intrusion est d'ouvrir un peu les portes de ses systèmes informatiques à des « intrus » qui vont certainement découvrir des fragilités. Certains, peut être plus paranoïaques que d'autres, peuvent se poser la question de savoir ce que vont devenir ces informations sensibles dans « les mains » de Hackers. Parmi les commanditaires, on trouve bien entendu les RSSI, mais aussi les chefs de projet d'application ou de produits embarquant des technologies de informatique ou de communication (Objets intelligents, connectés).

3.4 Les sociétés de confiance

Le niveau de sécurité du système d'information et des outils permettant de réaliser les audits sont vérifiés et validés par une société de certification (LSTI, AMOSSYS ...). A l'issue de certification, l'ANSSI prononce la qualification de la société d'audit au titre de ce PASSI. Il existe une extension pour les audits liés à la loi de programmation militaire (LPM). c'est à dire pour les audits sur les SIIV (Système d'information d'importance vitale) des Opérateurs d'importance Vitale. (Voir chapitre sur la Cyberdéfense)

3.4.1 Formation des Pentests

3.5 Certifications professionnelles



CEH  ¹¹ Hacker Éthique Certifié

L'objectif est de savoir comment rechercher les faiblesses et les vulnérabilités des systèmes à partir des mêmes outils et de connaissances qu'un hacker malveillant, mais d'une

¹¹. <https://www.eccouncil.org/programs/certified-ethical-hacker-ceh/>



manière légale et légitime pour évaluer la sécurité du système. La certification CEH se veut par ailleurs indépendante et neutre vis-à-vis des fournisseurs de produits et solutions.

OSCP ¹² Offensive Security Certified Professional Une des certifications reconnue pour être une référence dans le domaine des Ethical Hackers de métier. L'OSCP est une certification de l'offensive Security, organisme connu pour le système d'exploitation Kali Linux ¹³ (anciennement Backtrack), visant à vous fournir une certification attestant de vos compétences au niveau des tests de pénétration (Pentest). Cette certification se passe en ligne avec une dynamique de validation basée sur la mise en pratique des compétences au niveau d'un LAB accessible en VPN, avec le passage de différents niveaux de difficultés.

3.6 Les rapport d'audits, et cadre méthodologique

Au coeur de l'activité « professionnelle » des pentesteurs se situe le rapport d'audit. En effet, si l'objectif est bien d'identifier des fragilités (des vulnérabilités), et les scénarios qui permettent de les exploiter à des fins concrètes et relevant d'une menace présentant un risque pour l'entreprise, il n'en demeure pas moins important de « rendre compte » de ce qui a été trouvé. Ce rapport doit aussi contenir des préconisations, car il est important face à une ou des fragilité(s) de proposer des solutions.

3.7 Le Bug Bounty

Un bug bounty est une solution de « recherche de bugs et de vulnérabilités » proposée par des entreprises organisatrices qui permet à des personnes de recevoir une prime ou compensation après avoir remonté des bugs, surtout ceux concernant des failles et des exploits associés. Dans le domaine de la cybersécurité, on peut résumer de manière macroscopique cette activité à un déploiement de chasseurs de prime pour chasser les vulnérabilités « techniques » des systèmes informatiques.

La question est de savoir où trouver ces chasseurs de primes et d'organiser cette « chasse ». Le principe est d'organiser dans l'entreprise ou la structure un programme ciblé sur un système logiciel.

Un des programmes de Bug Bounty a été lancé au sein d'une entreprise pionnière de l'Internet, développeur du célèbre navigateur : Netscape. Créée en 1994 et rachetée par AOL en 1998, l'entreprise a disparue en janvier 2003. A l'époque 90% des internautes utilisaient ce navigateur et Jarrett Ridlinghafer, un ingénieur du support technique, avait constaté que les membres de la communauté « open source » corrigeaient d'eux-mêmes les bugs de l'application sans que l'entreprise ne soit dans la boucle de contrôle validation. C'est ainsi que naquit l'idée d'organiser et piloter cette chasse au bug et d'offrir des goodies au titre de récompenses. Aujourd'hui des entreprises intermédiaires proposent de réunir tous les conditions pour réaliser ces chasses au bug. Elle proposent des plateformes permettant de faire se croiser des « hackers » et des « éditeurs » autour de chasse au bugs

12. <https://www.offensive-security.com/information-security-certifications/oscp-offensive-security-certified-professional/>

13. <https://www.kali.org>



/ vulnérabilité.

L'utilisation de ces techniques pour trouver des vulnérabilités dans ses systèmes est conseillée en fin de cycle de recherche de vulnérabilités

3.8 Quelques entreprises

- ▶ HackerOne (<https://www.hackerone.com>)
- ▶ YesWeHack (<https://www.yeswehack.com>)

4. ANTICIPER et construire solide

Nous n'allons pas développer les concepts de *Security By design* qui englobent de nombreuses thématiques de la sécurité applicative. Les applications en particulier web et mobiles, de par leur complexité et les temps (et parfois budgets) restreints alloués à leur cycle de développement, contiennent souvent un grand nombre de vulnérabilités.

Comme nous l'avons abordé, tester de manière automatisée les vulnérabilités dans du code applicatif développé se décompose en deux typologies de tests :

- ▶ **Audit de code source automatisé** (SAST - Static Application Security Testing). L'audit du code source (SAST) des applications est important si vous souhaitez détecter et corriger leurs vulnérabilités pendant la phase de développement car en effet plus tôt une vulnérabilité est découverte et moins elle sera coûteuse à corriger. Un audit SAST est non intrusif par nature. Vous pouvez donc scanner en toute sécurité vos applications les plus critiques sans risque d'impacter leur performance.
- ▶ **Audit dynamique automatisé** (DAST - Dynamic Application Security Testing). Un audit dynamique (DAST) consiste à se servir d'un scanner pour interagir avec l'application (avec des requêtes malicieuses vers l'application auditée) afin d'y trouver des failles connues. Un scanner de vulnérabilités DAST est plus à même de détecter des erreurs de configuration au serveur web sur lequel est installée l'application.

Toutefois parmi celles-ci se trouve la recherche de vulnérabilités au sein des applications dès la phase de conception, et dans les phases de test et de production. Le terme APPSEC (*Application Security*) est souvent utilisé.

5. Produits de confiance

5.1 Les critères communs

⚙️ chapitre en cours de rédaction, DRAFT non publiable ⚙️

📖 chapitre à compléter, les éléments ne donnent qu'une vue trop réduite ou parcellaire du sujet

Pour affiner la gestion des vulnérabilités, il y a bien d'autres points à prendre en compte. Nous avons consigné ici ces points qui sont à développer. Nous ne donnons que des pistes de réflexion.



5.2 Périmètre sous responsabilité de l'entreprise

5.2.1 la notion de responsabilité

L'une des premières étapes d'un programme de gestion des vulnérabilités est un exercice de définition du périmètre de responsabilité et d'inventaire des actifs associés. En particulier au niveau de l'entreprise, les entreprises ont tendance à passer par une multitude de fusions, d'acquisitions et de nouvelles technologies et doivent donc combiner des systèmes incompatibles de manière native ou changer de personnel. Malheureusement, ces circonstances laissent souvent les entreprises confuses quant à la qualité de leur inventaire et beaucoup sont incapables d'identifier tous leurs actifs nécessitant un niveau de protection adéquat. Trop souvent, les entreprises possèdent une multitude d'actifs inconnus dans leur environnement qui pourraient compromettre leur sécurité sur le long terme.

Les experts de la sécurité considèrent que la gestion des actifs doit être confiée à une autorité unique qui assure la découverte pertinente dans tous les réseaux et services locaux, valide régulièrement l'inventaire des actifs et gère la gestion des modifications (dont les actifs nouveaux ou actifs retirés). Une fonction centralisée d'inventaire des actifs peut aider à clarifier l'inventaire des actifs d'une organisation et à renforcer le processus de gestion des vulnérabilités sécurité.

5.2.2 Inventaire des actifs

Une des vraie difficulté du déploiement d'une gestion de vulnérabilités efficace est la maîtrise des actifs vulnérables ou devant être surveillés et gérés en vulnérabilités. La gestion des systèmes d'information et des services informatiques (ITSM) est ainsi devenue un processus essentiel de la transformation digitale, considérée comme un outil privilégié qui va soutenir l'entreprise pour affronter sa propre complexité.

Dans un projet ITSM, le référentiel des actifs s'appelle CMDB (Configuration Management DataBase).

Cette base de données de gestion de configuration intègre tous les composants d'un système d'information pour avoir une vision d'ensemble sur l'organisation de ces composants et d'en piloter leur configuration en cas de besoin. Il est donc important de disposer de ce type d'outil pour pouvoir :

- ▶ Connecter cette CMDB à une solution de veille en vulnérabilités pour corréler les deux et avertir les bons acteurs sur l'apparition d'une vulnérabilité,
- ▶ Gérer les mécanismes de remédiation et de gestion des correctifs.

Il n'en demeure pas moins complexe de disposer d'un CMDB à jour, d'autant plus que des services dans le Cloud ne sont pas encore totalement intégrés dans les principes des CMDB, et que le shadow IT sévit toujours dans les entreprises.

La maîtrise des actifs passe par des outils de d'autodiscovery et d'analyse comportement-



taille » qui permet de découvrir non seulement les usages du SI mais aussi découvrir des composants actifs dans l'environnement.

Au coeur de la gestion des vulnérabilités, la gestion des actifs est aussi une histoire de responsabilité des périmètres informatiques. :

- ▶ IT métier
- ▶ Informatique de gestion
- ▶ Bureautique communicante.
- ▶ Réseau

Fiche Techno-Blog : Périmètre et responsabilité

Qui a la responsabilité d'un périmètre, et dans quelle mesure ce périmètre est géré en configuration ? Quelles sont les adhérences entre les périmètres, quelle fragilité peut induire des niveaux de maturité différents entre ces périmètres, qui peuvent posséder des RSSI différents ... ? Telles sont les questions qui sont en elles-mêmes des sujets de fond dans la gouvernance de la sécurité.

L'une des premières étapes d'un programme de gestion des vulnérabilités est un exercice de définition du périmètre de responsabilité et d'inventaire des actifs associés. En particulier au niveau de l'entreprise, les entreprises ont tendance à passer par une multitude de fusions, d'acquisitions et de nouvelles technologies et doivent donc combiner des systèmes incompatibles de manière native ou changer de personnel. Malheureusement, ces circonstances laissent souvent les entreprises confuses quant à la qualité de leur inventaire et beaucoup sont incapables d'identifier tous leurs actifs nécessitant un niveau de protection adéquat. Trop souvent, les entreprises possèdent une multitude d'actifs inconnus dans leur environnement qui pourraient compromettre leur sécurité sur le long terme.

5.2.3 L'environnement digital externe

Surveiller les failles dans l'entreprise est fondamental, mais il est aussi nécessaire de surveiller les failles apparaissant dans les services Cloud et dans les réseaux sociaux. Ces failles peuvent avoir un impact sur l'entreprise.

5.3 Veille et alerte sur les vulnérabilités

5.3.1 Abonnement au CERT

S'abonner à un Cert pour être informé en temps réel des vulnérabilités est devenu un besoin primordial pour réagir au plus vite. Il est important que ces alertes soient contextualisées, mais avec la difficulté de ne recevoir que les vulnérabilités qui nous intéressent, le lien avec le patch et la disposition des solutions de corrections.



5.3.2 Le marché de la vulnérabilité

Il existe un marché de la vulnérabilité. Les éditeurs (et les états), achètent des vulnérabilités, en gros ils payent des acteurs pour trouver de vulnérabilités.

DRAFT NOTES 2020-2021



6. Contributions

6.1 Comment contribuer

Les notes et les présentations sont réalisées sous \LaTeX .

Vous pouvez contribuer au projet des notes de cours CNAM SEC101 (CYBERDEF101). Les contributions peuvent se faire sous deux formes :

- ▶ Corriger, amender, améliorer les notes publiées. Chaque semestre et année des modifications et évolutions sont apportées pour tenir compte des corrections de fond et de formes.
- ▶ Ajouter, compléter, modifier des parties de notes sur la base de votre lecture du cours et de votre expertise dans chacun des domaines évoqués.

Les fichiers sources sont publiés sur GITHUB dans l'espace : (edufaction/CYBERDEF) ¹⁴. Le fichier Tex/Contribute/Contribs.tex contient la liste des personnes ayant contribué à ces notes. Le guide de contribution est disponible sur le GITHUB. Vous pouvez consulter le document **SEC101-C0-Contrib.doc.pdf** pour les détails de contributions.

6.2 Les contributeurs/auteurs du cours

6.2.1 co-auteurs

(2019-2020) **David BATANY** - Cnam SEC101 : *Architecture et fonctionnement des Botnets*

6.2.2 contributeurs

(2020) **Céline JUBY** - Orange Cyberdefense : *Contributions d'amélioration et relectures*

¹⁴. <https://github.com/edufaction/CYBERDEF>



Table des matières

1	Fragilités numériques	2
1.1	Détecter les fragilités de l'entreprise	2
1.2	Anticiper et surveiller les menaces	3
1.3	Les basics sur les vulnérabilités HOT	3
1.4	Exemples de vulnérabilités	5
1.4.1	Faible type XSS	
1.4.2	Faible type SQL Injection	
1.4.3	Vulnérabilités WEB	
1.5	Faibles de programmation	8
1.5.1	Exemple : SMB etherblue	
1.5.2	Exemple : programmation erronée	
1.6	Vulnérabilités et configuration	9
1.7	Vulnérabilités et exploits	9
1.8	Vulnérabilités et divulgation	9
1.9	CVE, CVSS et CWE	10
1.10	Common Vulnerabilities and Exposure (CVE)	10
1.11	Common Vulnerability Scoring System (CVSS)	12
1.12	Common Weakness Enumeration (CWE)	13
1.13	Les services de veille en vulnérabilités	14
1.13.1	Les CERT de l'ANSSI	
1.13.2	Les CERTs commerciaux	
1.13.3	La relation avec un CSIRT Interne	
1.14	les agences de notation	15
2	GERER les fragilités	15
2.1	Processus de gestion des vulnérabilités	16
2.1.1	ISO 27001	
2.1.2	Fenêtre d'exposition	
2.1.3	Processus d'analyse/recherche des vulnérabilités	
2.1.4	Processus d'évaluation des vulnérabilités (Vulnerability Assessment)	
2.2	Audit sécurité des vulnérabilités	21
2.2.1	Scan de Vulnérabilités du système	
2.2.2	Scan de Vulnérabilités logicielles	
2.3	La gestion des correctifs	21
2.3.1	De l'outillage	
2.4	Les audits	22
3	RECHERCHER des vulnérabilités	22
3.1	Les tests d'intrusion	22
3.2	Généralités	22
3.3	Le métier de Pentesteur	24
3.3.1	Ethical Hackers	
3.3.2	Peut-on faire confiance à des pentesteurs ?	
3.4	Les sociétés de confiance	24
3.4.1	Formation des Pentests	
3.5	Certifications professionnelles	24
3.6	Les rapport d'audits, et cadre méthodologique	25
3.7	Le Bug Bounty	25



3.8	Quelques entreprises	26
4	ANTICIPER et construire solide	26
5	Produits de confiance	26
5.1	Les critères communs	26
5.2	Périmètre sous responsabilité de l'entreprise	27
5.2.1	la notion de responsabilité	
5.2.2	Inventaire des actifs	
5.2.3	L'environnement digital externe	
5.3	Veille et alerte sur les vulnérabilités	28
5.3.1	Abonnement au CERT	
5.3.2	Le marché de la vulnérabilité	
6	Contributions	30
6.1	Comment contribuer	30
6.2	Les contributeurs/auteurs du cours	30
6.2.1	co-auteurs	
6.2.2	contributeurs	

Table des figures

