



# SEC 101

Analyse de risques  
Politiques et architectures de sécurité  
Sécurité opérationnelle

# le cnam Bretagne

## VEILLER : des risques identifiés aux vulnérabilités exploitables

Éléments de sécurité opérationnelle en cybersécurité d'entreprise

Eric DUPUIS

`eric.dupuis@lecnam.net`   `eric.dupuis@orange.com`

`http://www.cnam.fr`

Conservatoire National des Arts et Métiers  
Chaire de Cybersécurité

Publication DRAFT NOTES 2020-2021 du  
3 juin 2021, 10 h 27 CEST



# Sommaire





# Fragilités et Vulnérabilités

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.



# Identifier ses vulnérabilités

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.



# Exploitation des vulnérabilités

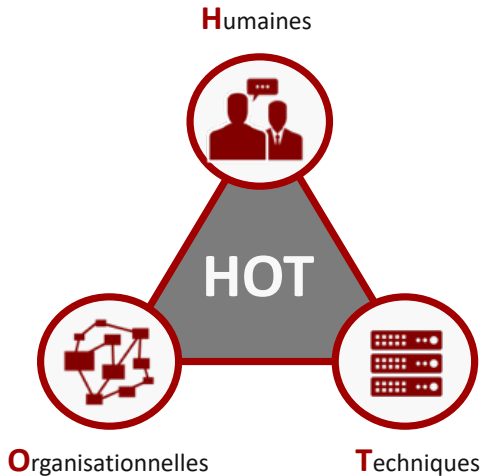
L'exploitation de ces fragilités, sont de deux grandes natures :

- 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.



# Fragilités HOT

les types de vulnérabilités





# Fragilités HOT

- 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).



## Zoom Fragilités TECHNIQUES

- **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.





# Groupes de failles

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.



# Typologie de failles

Les types de vulnérabilités

## Faillles « Systèmes »



Politique de sécurité / Configurations  
« laxistes » ou faibles

Conformité aux politiques  
de sécurité et/ou état de  
l'art / bonnes pratiques

## Faillles « applicatives »



Vulnérabilités  
Logicielles

Progiciels, Codes  
OpenSource  
(Editeurs,  
Auteurs)

CERT

Applications sur  
mesures  
(Développeurs)

Vérification  
et Validation



## AllowAll vs DenyAll

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



# Faillle, type XSS

Exemple

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

et permet de générer le code HTML suivant :

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



# Faible, type XSS

## Exemple

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>
```



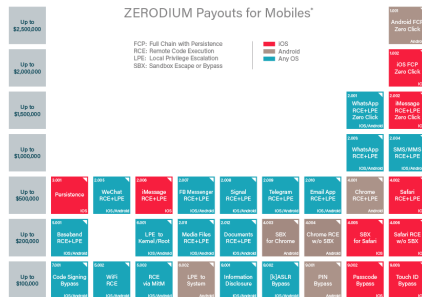
# zeroodium

Le marché des failles mobiles avec Zeroodium

## Changelog / Sep 3rd, 2019

Sep. 3, 2019 - Payouts for major mobile exploits have been modified. Changes are highlighted below:

Category	Changes
New Payouts (Mobiles)	<b>\$2,500,000</b> - Android full chain (Zero-Click) with persistence ( <b>New Entry</b> ) <b>\$500,000</b> - Apple iOS persistence exploits or techniques ( <b>New Entry</b> )
Increased Payouts (Mobiles)	<b>\$1,500,000</b> - WhatsApp RCE + LPE (Zero-Click) <b>without</b> persistence (previously: <b>\$1,000,000</b> ) <b>\$1,500,000</b> - iMessage RCE + LPE (Zero-Click) <b>without</b> persistence (previously: <b>\$1,000,000</b> )
Decreased Payouts (Mobiles)	<b>\$1,000,000</b> - Apple iOS full chain (1-Click) with persistence (previously: <b>\$1,500,000</b> ) <b>\$500,000</b> - iMessage RCE + LPE (1-Click) <b>without</b> persistence (previously: <b>\$1,000,000</b> )
Desktops/Servers	No modifications.



\*All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.

2019/09 © zeroodium.com



# Common Vulnerabilities & Weakness

Quelques concepts de gestion sur les vulnérabilités

## CVE

Common Vulnerabilities and Exposures



**Référentiel**, Base de données des vulnérabilités découvertes dans les produits et logiciels connus

**Méthode** pour évaluer la gravité d'une vulnérabilité

## CVSS

Common Vulnerability Scoring System



## CWE


Common Weakness Enumeration



**Base** de référence des sources et origine des fragilités, vulnérabilités informatiques



## MITRE et codification

*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://cve.mitre.org) <sup>1</sup>





Le *Common Vulnerability Scoring System (CVSS)* à sa version 3 issu des travaux du FIRST, Forum of Incident Response and Security Teams [↗](#)<sup>2</sup>, est un cadre méthodologique permettant d'évaluer en particulier la criticité d'une vulnérabilité.



# Cotation CVSS

Vecteur

Les notes et vecteurs CVSS sont toujours le résultat de trois groupes de critères d'évaluation (« Base », « Temporel » 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).
- Le groupe des critères « **Environnemental** » pondère le groupe « Temporel » en prenant en compte les caractéristiques de la vulnérabilité pour un Système d'Information donné.



# Cotation CVSS

criticité

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 :

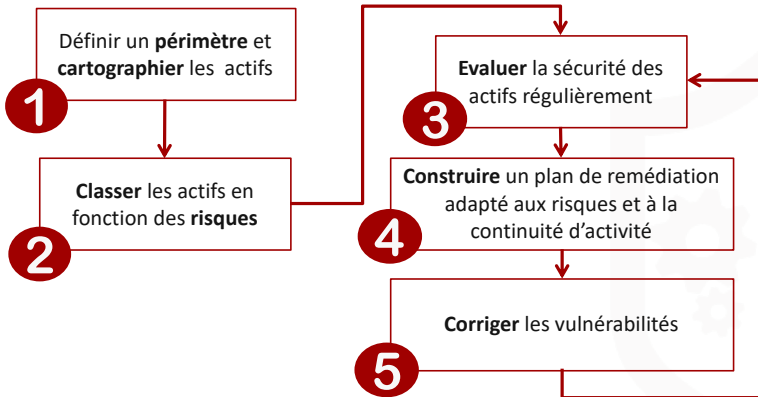
- Un score de 0 à 3.9 correspond à une criticité basse
- Un score de 4 à 6.9 correspond à une criticité moyenne
- Un score de 7 à 10 correspond à une criticité haute





# gestion des vulnérabilités

La gestion des vulnérabilités





## Cycle de vie VULMAN

- 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.



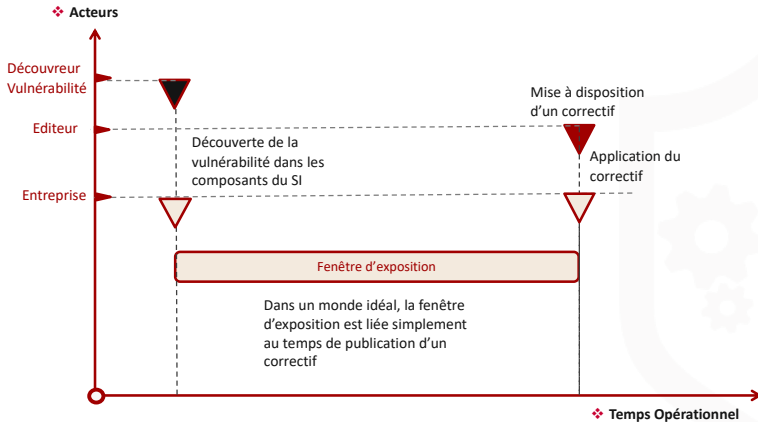


- 1. **DÉCOUVRIR** : Catalogage de l'existant, des actifs, des ressources du système d'information.
- 2. **PRIORISER** : Classifier et attribuer des valeurs quantifiables aux ressources, les hiérarchiser.
- 3. **ÉVALUER** : Identifier les vulnérabilités ou les menaces potentielles sur chaque ressource.
- 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.



# Fenêtre d'exposition idéale

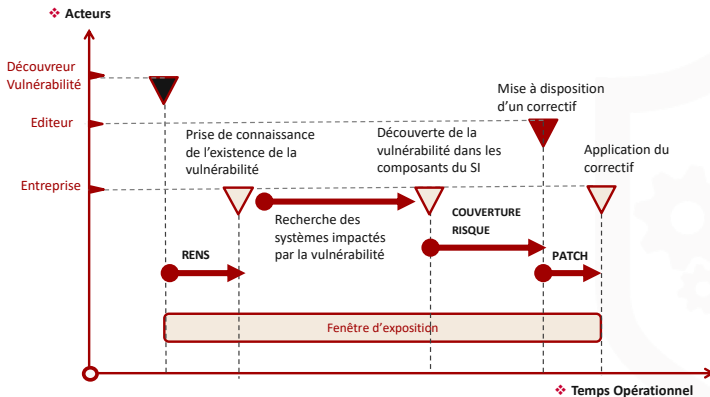
Fenêtre d'exposition idéale





# Fenêtre d'exposition

Fenêtre d'exposition







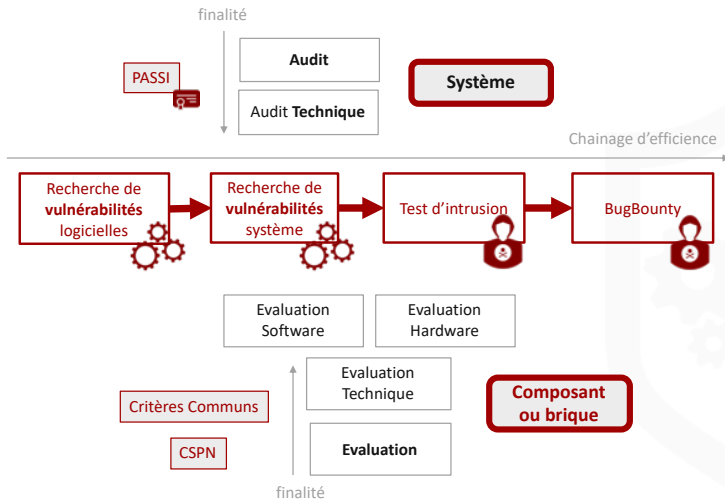
## Quand rechercher des vulnérabilités

- **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.



# Rechercher ses vulnérabilités

Rechercher ses vulnérabilités





# Caractéristiques d'audit

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 :

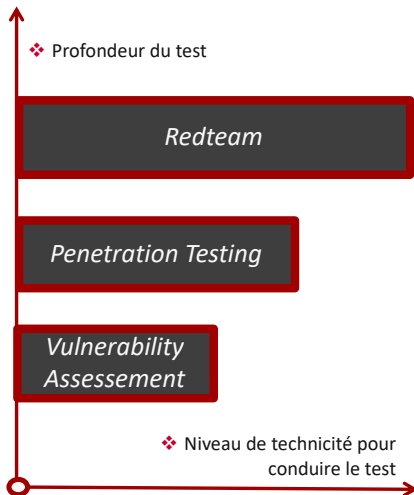
- Audits ponctuels et campagnes d'audit
- Audit continu





# Les types de tests

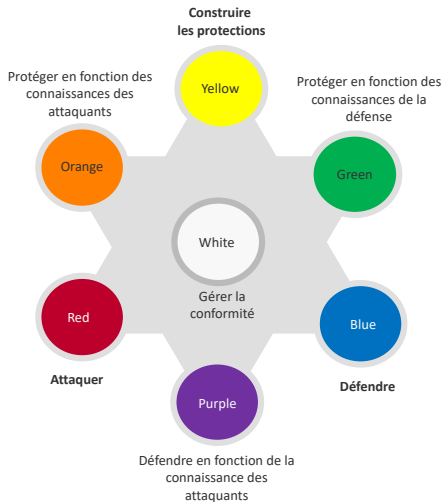
Les types de tests de vulnérabilités





# Les branches du test

Les branches du test





## 2 Certifications

reconnues pour les auditeurs techniques

### **CEH** ↗<sup>3</sup> 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 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** ↗<sup>4</sup> 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 ↗<sup>5</sup> (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.



- **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.



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.







des questions ?

contacter [eric.dupuis@lecnam.net](mailto:eric.dupuis@lecnam.net)

**CYBERDEF**



**101**

*Tous les documents publiés dans le cadre de ce cours sont perfectibles,  
ne pas hésiter à m'envoyer vos remarques !*



# Contributions

Les notes et les présentations sont réalisées sous  $\text{\LaTeX}$ .

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) [↗](https://github.com/edufaction/CYBERDEF)<sup>a</sup>. 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.

---

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



## Mises à jour régulières

Eduf@ction eric.dupuis@lecnam.net

Vérifiez la disponibilité d'une version plus récente de

**SEC101-C3a-VulMan.prz.pdf** sur GITHUB CYBERDEF <sup>6</sup>



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

