Sécurité web opérationnelle et investigation numérique

LAB / D34

DIFFUSION : **RESTREINT**

Destinataires : Stagiaires en session DFS

SOMMAIRE

Contexte

En tant que développeur full stack, tu seras très certainement la première ressource à laquelle on fera appel dans le cas d'une situation de crise concernant le site ou application web sur laquelle tu travail habituellement.

Tu as étudié la sécurité web lors de la séquence D25, et cette session est l'occasion de parfaire tes connaissances en sécurité. Elle est dédiée à l'apprentissage par l'exemple au travers d'exercices correspondant chacun à un cas d'étude réel. Elle a pour objectif de t'enseigner les bons réflexes qui te seront indispensables lorsque tu seras confronté à une situation d'urgence (site/application web ou serveur web compromis).

Il s'agit d'un travail individuel mais les échanges entre pairs sont largement encouragés afin de mieux progresser. En cas de blocage total, l'équipe est à ta disposition à l'adresse pedagogie@it-akademy.fr pour te fournir des indices et des pistes nécessaires pour continuer d'avancer.

Le rendu final s'effectue sur Google Classroom.

Important : Le but n'est pas d'aller vite, mais de comprendre les détails de chaque situation simulée. Aussi, nous n'attendons pas de toi d'avoir résolu tous les challenges en fin de séquence et nous te laissons 30 jours pour rendre ton fichier de résultats.

Nous te souhaitons un bon entrainement !

Le matériel d'étude, propre à chaque cas, est fourni dans le fichier « D34-material.infected.zip »figurant en pièce jointe de ce sujet dans Google Classroom.

*Le mot de passe de l'archive est « **infected** ».*

Les réponses aux questions posées sont à reporter dans le fichier « answers.md » contenu dans le fichier « D34-material.infected.zip ».

1. Cas d'étude 01

Le site web de ta société vient d'être attaqué ! Tu as immédiatement stoppé l'application et mis en oeuvre une page « *en cours de maintenance* ». Il est maintenant nécessaire d'analyser et de comprendre la nature de l'attaque subie afin d'identifier la ou les failles exploitées et d'apporter les correctifs nécessaires pour pouvoir remettre l'application en production dans les plus brefs délais.

Pour cela, tu procèdes à une extraction du dernier fichier de journalisation du serveur web : « *access.log* ». Nous te conseillons l'utilisation d'un logiciel de visualisation tel que https://cloudvyzor.com/ pour une lecture simplifiée.

Pour t'aider dans la démarche, nous t'invitons à répondre aux questions suivantes en précisant ton mode opératoire (outils, méthode, étapes…) :

1. Quel outil a été utilisé pendant la phase de reconnaissance (scan) ?

2. Après la phase de reconnaissance, quelle technique a été utilisée pour découvrir la liste des répertoires accessibles sur le serveur web ?

3. Après la fin de la phase de découverte des répertoire, de quel type d'attaque l'application est-elle la cible ?

4. Cette dernière attaque a-t-elle fonctionné ou a-t-elle échoué ?

5. Quel est le type de la quatrième attaque (son nom) et à quelle heure débute-t-elle ?

6. Quelle est la charge malveillante (payload) utilisée en premier lors de la quatrième attaque ?

7. Y'a-t-il une trace démontrant la persistance de l'attaque ? si oui, quelle est la charge malveillante (payload) utilisée ?

8. Quels sont les actions à entreprendre avant la remise en ligne de l'application ?

2. Cas d'étude 02

Plusieurs utilisateurs de votre société sont les victimes d'un incident de sécurité touchant leur poste de travail depuis hier. Dans 100% des cas, ils expliquent avoir constaté l'infection après l'ouverture des documents Word téléchargés depuis le back office de l'application web que vous développez.

Même si tu n'es pas expert en cybersécurité, tu es le seul à pouvoir intervenir rapidement pour aider ton entreprise à comprendre ce qui s'est passé et essayer de trouver une solution pour éviter que cela ne se reproduise.

Les fichiers incriminés ont été isolés et placés dans le répertoire Case02/files.

ATTENTION : CES FICHIERS CONTIENNENT DES CHARGES MALVEILLANTES ACTIVES, NE LES UTILISEZ PAS SUR UN AUTRE MATÉRIEL QUE CELUI FOURNI PAR IT-AKADEMY.

Pour t'aider, nous te proposons quelques outils qui te seront très utiles dans le répertoire Case02/tools, l'utilisation de VirusTotal.com, ainsi que quelques questions qui permettront d'orienter ta démarche :

1. Quelle est l'adresse IP malveillante utilisée dans le fichier Employees_Contact_Audit_Oct_2021.docx ?

2. Quel est le nom de domaine malveillant utilisé dans le fichier Employee_W2_Form.docx file ?

3. Quel est le nom de domaine malveillant utilisé dans le fichier Examing the Work_From_Home_Survey.doc ?

4. Quel est le nom de domaine malveillant utilisé dans le fichier income_tax_and_benefit_return_2021.docx ?

5. Quelle est la vulnérabilité technique exploitée dans cet incident ?

6. Quelle(s) solution(s) technique(s) proposes-tu au niveau du back-office de l'application pour prévenir ce type d'incident ? Quel pourrait-être le coût de mise en oeuvre, sachant que nous traitons environ 40Go de documents uploadés par mois ?

3. Cas d'étude 03

Le SIEM[1] de ton entreprise a remonté l'alerte suivante :

```
SEVERITY : HIGH
DATE : Feb, 25, 2022 11:34 AM
EVENTID : 115
RULENAME : SOC165 - Possible SQL Injection Pauload Detected
TYPE : Web attack
DETAILS :

Level : Security Analyst
Hostname : WebServer1001
Destination IP Address : 172.16.17.18
Source IP Address : 167.99.169.17
HTTP Request Method : GET
Requested URL : https://172.16.17.18/search/?q=%22%20OR%201%20%3D%201%20--
%20-
User-Agent : Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101
Firefox/40.1
Alert Trigger Reason : Requested URL Contains OR 1 = 1
Device Action : Allowed
```

Comme précisé dans le détail de l'alerte, le serveur web concerné est WebServer001 qui héberge une application de type « extranet » écrite en PHP.

À ce stade, rien n'indique que cette attaque a réussi mais ton manager te demande tout de même de t'en assurer. Afin de t'aider dans la démarche, nous te proposons les questions suivantes :

1. L'adresse IP source est-elle connue pour être source d'activités malveillantes ?
2. Comment penses-tu procéder pour évaluer la portée de l'attaque sur l'application (réussite ou non, effets constatés, conséquences,...) ? Décris la méthode et les outils utilisés.
3. Considérant que l'application est vulnérable à ce type d'attaque, propose plusieurs solutions techniques (complémentaires) à implémenter afin de corriger ce type de faille et renforcer l'application.

[1] https://www.microsoft.com/fr-fr/security/business/security-101/what-is-siem#:~:text=Un%20syst%C3%A8me%20de%20gestion%20des,elles%20ne%20perturbent%20leurs%20activit%C3%A9s.

4. Cas d'étude 04

Parmi les grands principes de sécurité, il y a la discrétion et même l'obscurité. Trop en dire est souvent une erreur qui facilite le travail de l'agresseur. En tant que lead développeur de HackyCorp, tu es garant de la bonne application de ce principe par ton équipe.

Pour les besoins de ce cas d'étude, nous te précisons que la surface d'étude/attaque est la suivante :

- **hackycorp.com**
- ***.hackycorp.com**
- **0x[%02x].a.hackycorp.com**
- **balancer.hackycorp.com**
- **z.hackycorp.com**
- Compte Github de Hackycorp

Pour chaque question, tu reporteras la clé secrète qui s'affiche lorsque la solution est trouvée dans le fichier « answers.md » et tu expliqueras en une phrase simple ta manière de procéder.

Commençons par étudier l'architecture générale et l'environnement serveur de l'application :

1. Retrouve le contenu du fichier robots.txt du site principal de Hackycorp.

Pour ton information, le fichier robots.txt contient des directives à destination des moteurs de recherches afin de leur dire comment indexer le contenu d'un site et surtout quoi exclure. Dans le cadre d'une attaque, il contient des informations souvent intéressantes sur la structure du site web.

2. Souvent, les pages d'erreur mal configurées révèlent de précieuses information. Que remarques-tu sur le site principal ?

3. De nombreux sites web contiennent, à la racine, un fichier security.txt permettant de dire aux chercheurs en sécurité comment faire remonter une information sur une faille ou une vulnérabilité. Mais est-ce vraiment une bonne idée ?

Pour en savoir plus : securitytxt.org et Security.txt.

4. Tu vas maintenant profiter d'une mauvaise configuration des serveurs web encore très courante : le directory listing. Elle te permettra de trouver, sur le site principal, un répertoire dont tu peux découvrir le contenu.

5. Tu vas pouvoir tenter de deviner le nom d'un répertoire couramment utilisé pour les pages d'administration. En effet, avant de lancer un brute-force sur l'arborescence d'un

site web, il est souvent payant de tester les noms de répertoires les plus communément répandus (applications connues, outils du marché).

6. Dans cet exercice, tu vas tenter de trouver un répertoire qui n'est pas directement accessible. Dans ce cas, des outils comme patator, FFUF ou WFuzz vous seront utiles.

7. Sauras-tu trouver le virtualhost par défaut du serveur principal ?

Pour ce faire, tu tenteras de remplacer le nom d'hôte par son adresse IP ou de modifier aléatoirement l'entête Host de la requête. Curl -H « Host: … » est ton ami.

8. Même chose que précédemment, mais avec le virtualhost par défaut du serveur principal sur transport TLS.

9. Dans cet exercice, nous allons nous intéresser aux entêtes de la réponse HTTP.

10. Dans cet exercice, nous nous intéressons aux noms alternatifs présents dans le certificat du site. Utilise ton navigateur ou encore openssl…

11. Dans cet exercice, tu vas effectuer une reconnaissance visuelle pour retrouver l'application avec la clé en rouge.

Les applications sont hébergées selon le schéma suivant :

`0x["%02x"].a.hackycorp.com` comme par exemple :

- **`0x00.a.hackycorp.com`**
- **`0x01.a.hackycorp.com`**
- …
- **`0x0a.a.hackycorp.com`**
- **`0x0b.a.hackycorp.com`**
- …

Il sera bien sûr nécessaire d'automatiser ta recherche…

12. Il s'agit maintenant de brute forcer un virtualhost en manipulant l'entête Host. Il n'y a pas de résolution DNS sur cet hôte. Tu dois donc ciblé hackycorp.com et bruteforcer le virtualhost qui termine par .hackycorp.com.

13. Accède maintenant au load balancer sur balancer.hackycorp.com.

Il est souvent interessant pour un assaillant d'émettre de multiples fois une même requête afin de découvrir si plusieurs back sont impliqués dans son traitement.

14. Intéresse-toi aux enregistrements DNS TXT de key.z.hackycorp.com.

Les enregistrements TXT contiennent régulièrement des informations précieuses…

15. Dans cet exercice, tu vas étudier les transferts de zone.

Ils sont souvent utilisés pour synchroniser de multiples serveurs DNS. Une liste prédéfinie des hôtes autorisés à effectuer cette opération devrait être définie, mais ce n'est pas toujours le cas, et il est possible d'avoir accès à de nouveaux hôtes.

Tu effectueras un transfert de zone pour z.hackycorp.com.

16. Est-il possible d'effectuer un transfert de la zone interne nommée « int » en utilisant le serveur de noms de z.hackycorp.com ?

Tu verras qu'il est parfois possible d'accéder aux informations d'une zone interne en passant par des serveurs publics.

17. Est-il possible de connaître la version de Bind exécutée sur le serveur z.hackycorp.com.

Imaginons maintenant qu'un assaillant prépare une campagne de phishing contre les développeurs de ton équipe (https://github.com/hackycorp) :

18. Est-il possible de trouver le nom du développeur qui a « commit » le code du dépôt « test1 » sur GitHub ? Quel est-il ?

19. Ce développeur possède-t-il des dépôts publics ? Si oui, laisse-t-il trainer des informations sensibles ?

20. Trop souvent encore, les développeurs fonts des *commits* avec une mauvaise adresse e-mail, ce qui peut conduire à faire fuite des adresses personnelles ou encore des comptes systèmes. Trouveras-vous une adresse différente des autres dans le dépôt « repo7 » ?

Astuce : la commande « git log » en local est ton amie.

21. Cette fois-ci, rien de sensible n'a été laissé par les développeurs dans le code source de la branche « master » du dépôt « repo4 ». Mais il y a souvent d'autres branches …

22. Souvent, lorsqu'on publie accidentellement une information sensible sur Github, on supprime cette information du code local puis on effectue un nouveau commit. Est-ce une bonne idée ? NON. Sauras-tu trouver les informations qui ont été supprimées dans le dépôt « repo9 » ?

Astuce : l'outil en ligne de commande « tic » en local est ton ami.

23. De même, les messages de commits peuvent contenir de précieuses informations… Regarde du côté du dépôt « repo0a ».

Astuce : ici encore, l'outil en ligne de commande « tic » en local est ton ami.

Revenons maintenant aux pratiques de coding de ton équipe :

24. Dans cette exercice, nous allons nous intéresser aux fichiers statiques disponibles publiquement sur les serveurs d'assets (Javascript, CSS,...). Il existe un fichier key.txt quelque part.

25. Dans cet exercice, nous allons vérifier si les développeurs que tu encadres font parfois preuve de flemme quant à la manipulation des mots de passe et autres informations sensibles et qui « hard codent » certaines d'entre elles.
Effectue une *code review* du Javascript.

Astuce: tu recherches la valeur de la clé ptlab_key_recon_26.

26. BILAN - À partir de tous les cas de figure passés en revue dans l'étude de cas N°4, tu es en mesure de déduire un ensemble de points de vigilance et de bonnes pratiques à mettre en oeuvre au niveau de ton équipe pour renforcer la sécurité. Tu dresseras une liste des actions à entreprendre et tu la reporteras dans le fichier « answers.md ».