

Simulation de pentest

Meeting 01

Marine Bodson (Mayonnaise)

February 19, 2025

Overview

1. Avant de commencer

2. Walkthrough

3. Rédiger un rapport

4. Tips et bonnes pratiques

5. Ressources utiles

Quel "hacking environment" choisir ?

1. Machine virtuelle

- **Exemple (hyperviseur)** : VirtualBox, VMWare, Gnome Boxes,...
- **Exemple (OS)** : Kali Linux, ParrotOS, Ubuntu,...
- **Avantage** : Séparation entre l'environnement de testing et l'environnement de base, snapshots,...
- **Désavantage** : Restreinte en ressources, lag, distros peu stables,...

2. Docker

- **Exemple** : Exegol
- **Avantage** : Facile à installer et désinstaller, léger,...
- **Désavantage** : Moins d'isolation, problème de compatibilité avec certains outils, François le fan boy,...

3. Environnement de "base"

- **Exemple** : Ubuntu, Fedora, Debian,...
- **Avantage** : Utilise les capacités complète de la machine, simplicité,...
- **Désavantage** : Pas d'isolation, doit être plus conscient des risques de sécurités,...

Prendre des notes

Lors d'un pentest, il est très important de prendre des notes. Cette étape est souvent ignorée ou bâclée par les débutants. Voici pourquoi il est important de prendre des notes (non exhaustif) :

1. **Savoir ce qu'on a déjà testé**

- Moins de temps perdu à tester des choses déjà faites
- On sait ce qui a été testé et donc ce qui ne l'a pas été

2. **Lors de problèmes techniques :**

- Permet de déterminer rapidement et efficacement ce qui a causé le problème
- Permet de déterminer si le client a été attaqué ou s'il s'agit de notre testing

3. **Lors de la rédaction du rapport :**

- Permet de fournir des descriptions de vulnérabilités plus complètes
- Évitant de demander au client de récupérer des accès si nous n'avons pas pris assez de notes ou de captures d'écran

Exemple

▼ Obsidian Pentest Template 1.0

- 1. Recon Notes
- 2. Recon Targets
- 3. Enumeration Notes
- 4. Enumerating Targets
- 5. Exploitation Notes
- 6. Exploitation Targets
- 7. Post Exploitation
- 8. Post Exploitation Targets
- 9. High Value Information_Reporting

▼ 3. Enumeration Notes

▼ Services

- 1. FTP
- 2. SSH
- 3. Active Directory (AD)
- 4. Email Services
- 5. SNMP
- 6. Web (HTTP_HTTPS)
- 7. Network Shares (SMB, SAMBA, NFS)
- 8. Other Services

General Notes

Impacket General Notes

Impacket Kerberoasting

Impacket NtlmRelayX

Inveigh

ntlm_theft

Password Spraying

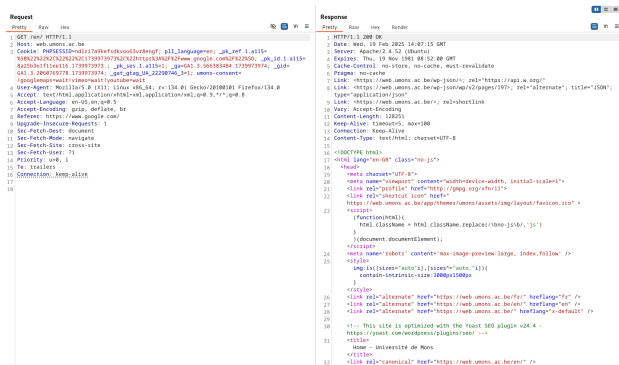
Pretender

Responder

Quelques conseils et bonnes pratiques

- **Utiliser un logger** : Enregistrez vos sessions de terminal, y compris toutes les commandes et leurs sorties.
 - **Exemple** : Utilisation de sessions Tmux.
 - Pas indispensable, mais très utile pour garder une trace détaillée, surtout pour les plus méticuleux.
- **Prendre des screenshots de qualité** : Un bon screenshot peut faire toute la différence.
 - C'est un véritable art d'obtenir des captures claires et pertinentes ! Oui monsieur !
- **Explorez d'autres bonnes pratiques** : N'hésitez pas à consulter des ressources en ligne pour des conseils de pro :3

Mauvais screenshot



- On ne voit rien et on se sait rien lire :/
- Qu'est-ce que vous voulez montrer sur ce screenshot ?

```

1 GET /en/ HTTP/1.1
2 Host: web.umons.ac.be
3 Cookie: PHPSESSID=ndlrI7a9kefsdkvoo63vr8engfj; pll_language=en; _pk_ref.1.a115=
   K5BwZ2%22%2C%22%22%2C%22%22%2F%2Fwww.google.com%2F%22%2D; _pk_id.1.a115=
   8a25b3e1f1ee116.1739973973.; _pk_ses.1.a115=1; _ga=GA1.3.566383484.1739973974; _gid=
   GA1.3.2860679778.1739973974; _gat_gtag_UA_22290746_3=1; umons-consent=
   !googlemaps-wait!vimeo-wait!youtube-wait
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:134.0) Gecko/20100101 Firefox/134.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://www.google.com/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Priority: u=0, i
15 Te: trailers
16 Connection: keep-alive

```

```

1 HTTP/1.1 200 OK
2 Date: Wed, 19 Feb 2025 14:07:15 GMT
3 Server: Apache/2.4.52 (Ubuntu)
4 Expires: Thu, 19 Nov 1981 08:52:00 GMT
5 Cache-Control: no-store, no-cache, must-revalidate
6 Pragma: no-cache
7 Link: <https://web.umons.ac.be/wp-json/;>; rel="https://api.w.org/"
8 Link: <https://web.umons.ac.be/wp-json/wp/v2/pages/197?>; rel="alternate"; title="JSON";
   type="application/json"
9 Link: <https://web.umons.ac.be/>; rel=shortlink
10 Vary: Accept-Encoding
11 Content-Length: 128251
12 Keep-Alive: timeout=5, max=100
13 Connection: Keep-Alive
14 Content-Type: text/html; charset=UTF-8
15
16 <!DOCTYPE html>
17 <html lang="en-GB" class="no-js">
18 <head>
19   <meta charset="UTF-8">
20   <meta name="viewport" content="width=device-width, initial-scale=1">
21   <link rel="profile" href="http://gmpg.org/xfn/11">
22   <link rel="shortcut icon" href="
   https://web.umons.ac.be/app/themes/umons/assets/img/layout/favicon.ico" >

```

- Comme on a pris l'essentiel dans la capture, celle-ci devient plus lisible ! :)
- Les box permettent d'attirer l'attention sur ce qu'on veut montrer !
- N'hésitez pas à utiliser d'autres figures telles que des flèches pour aider le lecteur à se concentrer sur les aspects importants.

1. Avant de commencer

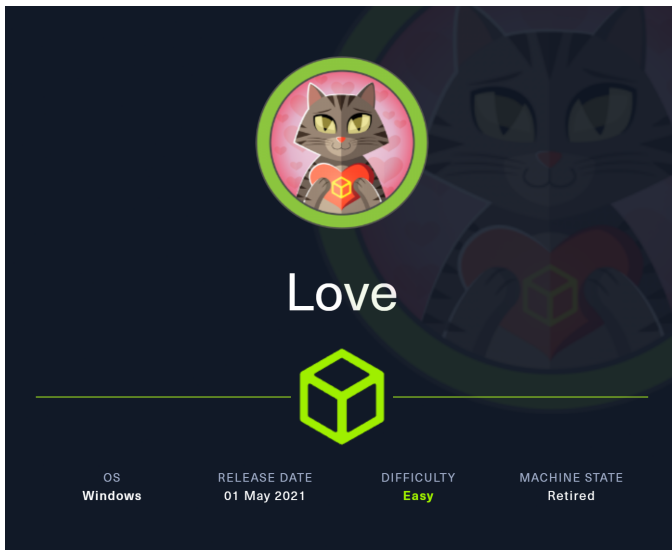
2. Walkthrough

3. Rédiger un rapport

4. Tips et bonnes pratiques

5. Ressources utiles

HTB Machine



The image shows a card for the HTB machine 'Love'. At the top is a circular avatar of a grey cat with yellow eyes holding a red heart with a yellow cube on it. Below the avatar is the name 'Love' in white text. Underneath the name is a large yellow cube icon. At the bottom, there are four columns of information: OS (Windows), RELEASE DATE (01 May 2021), DIFFICULTY (Easy), and MACHINE STATE (Retired). The background of the card is dark blue with a faint, larger version of the cat avatar.

Love

OS
Windows

RELEASE DATE
01 May 2021

DIFFICULTY
Easy

MACHINE STATE
Retired

Hacking Time



Figure: Me btw

1. Avant de commencer

2. Walkthrough

3. Rédiger un rapport

4. Tips et bonnes pratiques

5. Ressources utiles

Que contient un rapport ?

- **Le blabla légal** : Mentions légales et conditions de confidentialité.
- **Executive Summary** : Résumé succinct des problèmes majeurs et de leur impact.
- **Récapitulatif des vulnérabilités** : Présenté sous forme de graphes ou de tableaux pour une visualisation claire.
- **Findings** : Détails des vulnérabilités identifiées.
 - **Titre + CVSS** : Nom de la vulnérabilité et score de criticité.
 - **Description et contexte** : Explication de la vulnérabilité et son impact dans le cadre du pentest.
 - **Walkthrough** : Reproduction de la vulnérabilité pour démontrer son exploitation.
 - **Impact** : Évaluation des conséquences possibles de la vulnérabilité.
 - **Recommandations** : Bonnes pratiques et solutions pour corriger la vulnérabilité.
 - **Références** : Documentation et sources supplémentaires utilisées.

Un exemple de rapport se trouve sur le dépôt GitHub du meeting :)

Quand commencer à écrire le rapport ?

Dès que vous commencez le pentest !!!

Écrire un rapport est un processus qui prend du temps, et personne n'aime écrire tout d'un coup.

- Commencez à ajouter les vulnérabilités dès que vous les identifiez :
 - Elles sont encore fraîches dans votre esprit, ce qui facilite la rédaction.
 - Plus vous attendez, plus il sera difficile de vous souvenir des détails.
 - Plus les vulnérabilités s'entassent, plus vous aurez de chose à écrire
- Plus vous documentez tôt, plus vous avez le temps d'affiner et d'améliorer le rapport.

Conseils pour la rédaction de l'executive summary

Pour rédiger un bon executive summary, imaginez que vous devez expliquer les problèmes à vos parents :

Quels sont les risques ? Quelle est leur portée ? Comment les résoudre ?

- **Évitez le jargon technique** : Présentez les problèmes de manière simple et compréhensible.
- **Mettez en évidence les points essentiels** : Attirez l'attention sur les problèmes majeurs et leur impact.
- **Soyez concis** : Résumez au maximum. Un executive summary trop long perdra rapidement l'attention du lecteur.
- **Ajoutez des éléments positifs** : Soulignez ce qui a été bien fait, pat pat sur la tête...

Conseils pour la rédaction des findings

- **Choisissez des titres explicites** : Privilégiez des titres qui décrivent clairement l'impact de la vulnérabilité.
 - *Exemple* : "SQLi in Login Form" \Rightarrow "SQLi in Login Form Leading to Dump of Admin Credentials".
- **Précisez l'impact dans le walkthrough** : Décrivez clairement les conséquences de la vulnérabilité dans le contexte de l'application pour montrer sa dangerosité.
 - *Exemple* : Pour un XSS, est-il stored ? Qui peut le trigger ? Peut-on voler des cookies de session ? Y a-t-il différents rôles d'utilisateurs (utilisateur, admin) ?
- **Soignez la section recommandations** : Ne négligez pas la partie recommandations, proposez des solutions concrètes pour aider les développeurs à corriger le problème.

Conseils généraux pour la rédaction

- **Utilisez l'IA à votre avantage** : Que ce soit pour la rédaction, la correction grammaticale ou les fautes d'orthographe. On est plus à l'école, facilitez-vous la tâche ;)
- **Mettez en évidence les éléments importants** : Utilisez des couleurs (par exemple, rouge ou vert) ou du texte en gras pour attirer l'attention sur les points clés.
- **Faites-vous relire** : Demandez à vos collègues ou camarades de relire votre rapport pour vous assurer qu'il est clair, précis et sans erreur.

La qualité du rapport ne dépend pas seulement des vulnérabilités trouvées !

1. Avant de commencer
2. Walkthrough
3. Rédiger un rapport
- 4. Tips et bonnes pratiques**
5. Ressources utiles

Bonnes pratiques

- Attention avec le bruteforcing et l'énumération automatisée. Vous risquez d'overload le serveur ou bloquer des comptes, → toujours demander aux clients ou à vos supérieurs/collègues.
- Si vous trouvez une vulnérabilité critique, toujours mieux de contacter le client pour le prévenir.
- Si vous obtenez un accès interne lors d'un pentest web/externe, sauf si demandez avant, demandez au client s'ils veulent que vous procédiez ou si on arrête pour fix asap.
- Si vous trouvez un RCE via file upload, ne créez pas un fichier avec le nom shell, webshell, etc... Extrêmement dangereux car d'autres personnes pourraient tomber dessus.

Bonnes pratiques (the encore)

- Notez les modifications que vous avez effectuées, afin de rétablir l'environnement à son état de base après le testing.
- Lorsque vous utilisez un outil, soyez certains de ce que fait celui-ci et ce que font les paramètres que vous avez entré.

⇒ **Soyez conscient des dégâts que vous pouvez faire en étant imprudent !**

My very few honest tips

- La méthodologie est aussi importante que les capacités techniques ! Améliorer votre méthodologie en vous entraînant.
- Il existe des checklists en ligne de telle méthodologie pour vous aider, mais toujours mieux de développer la sienne.
- Posez des questions même si elles vous paraissent bêtes, il vaut mieux être bête un jour que toujours !
- Abusez des plateformes en lignes qui proposent des labs et cours telles que Try Hack Me et Hack the Box pour vous améliorer techniquement mais également votre méthodologie
- Regardez les vidéos de IppSec. Il a une super méthodologie et ses vidéos sont très éducatives.
- Les bug bounty vous permettent de tester vos compétences sur des targets réelles (et c'est rémunéré :P

Ressources utiles

- Templates de notes Obsidian par TJ Null : <https://github.com/tjnull/TJ-OPT>
- Chaine YouTube de lppSec :
<https://www.youtube.com/channel/UCa6eh7gCkpPo5XXUDfygQQA>
- GitHub du meeting : <https://github.com/MarineB210/club-meeting-01>