Afbeelding met tekening, teken

Automatisch gegenereerde beschrijving

**Machine**

Passage

**Date of done**

17-09-2020

**Author**

ChefByzen

**Omschrijving**

Passage is een Linux-based machine in Hack The Box. Deze box heeft in Hack The Box een *medium* rating. Vanuit de gebruikers die deze machine gedaan hebt verschilt de rating tussen *medium* en *easy*. Hij is 05-09-2020 uitgekomen en daarom is deze machine vrij nieuw en heeft die weinig *owns.*

**Moeilijkheidsgraad**

Medium

**Vlaggen**

Gebruiker: 0abbed6d9fd479f2dca08a5aeb89a944

Root: e5738259f9a716b1aca05dfcde149c23

Inhoud

[Introduction 3](#_Toc61430319)

[Reconnaissance 4](#_Toc61430320)

[Nmap 4](#_Toc61430321)

[DIRB 4](#_Toc61430322)

[Intrusion 6](#_Toc61430323)

[Webserver 6](#_Toc61430324)

[Rss Feed 6](#_Toc61430325)

[CuteNews 6](#_Toc61430326)

[Exploitation 8](#_Toc61430327)

[Path traversal 8](#_Toc61430328)

[RSS Feed 8](#_Toc61430329)

[CuteNews 8](#_Toc61430330)

[Privilege Escalation 11](#_Toc61430331)

[User 11](#_Toc61430332)

[Root 12](#_Toc61430333)

[Clean-up 14](#_Toc61430334)

# Introduction

Tijdens het hacken van deze machine wordt gebruik gemaakt van de cyber kill chain. Echter, zal tijdens het hacken van alleen maar een box niet alle fasen worden doorgaan. Zo zullen de volgende fasen niet worden doorlopen:

* **Lateral movement** aangezien je maar met 1 systeem te maken hebt.
* **Obfuscation** deze fase is niet nodig aangezien jij als enige op die box werkt.
* **Denial of Service** dit is onnodig aangezien je een machine wil hacken niet down wil brengen.

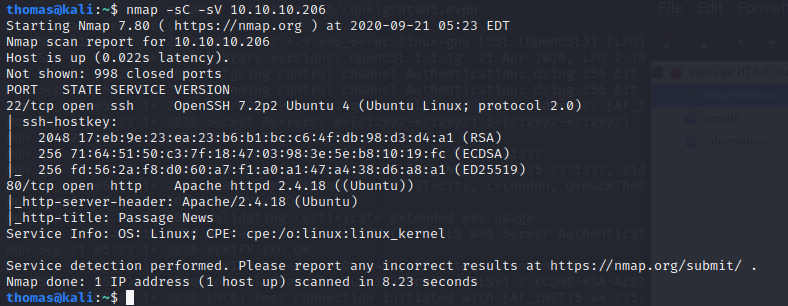
# Reconnaissance

*Reconnaisance* is een fase beginnend aan een penetratietest (in dit geval het hacken van een box). De pentester probeert zo veel mogelijk openbare informatie te verkrijgen over de targetomgeving. ... Dit betekent dat openbaar beschikbare informatie wordt opgezocht over de targetorganisatie met behulp van een webbrowser, newsgroups, DNS registraties, etc.

## Nmap

Om te beginnen heb ik het IP-adres gescand van Passage (10.10.10.206).   
De parameters die ik heb gebruikt zijn:

* **-sC** -> Dit heb ik gebruikt om default scripts uit te voeren
* **-sV** -> Dit heb ik gebruikt, zodat er bij open poorten wordt gezocht naar service/versie informatie
* Vaak gebruik ik ook de parameter **-oA** om de output van de *nmap* op te slaan in een bestand.



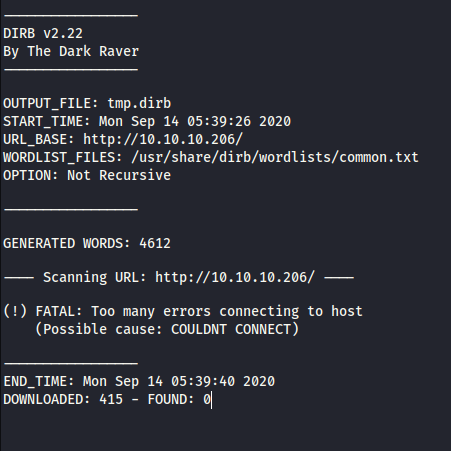
Figuur 1 nmap

Met de output is te zien dat er een webserver draait op poort 80 (<http://10.10.10.206:80>). Ook staat de SSH poort open en is deze mogelijk te gebruiken voor een *remote* connectie.

Zie Webserver voor de diepgang over de informatie verzameling omtrent deze webserver.

## DIRB

*DIRB* is een Web Content Scanner. Het zoekt voor bestaande (of verborgen) web objecten. Simpel gezegd werkt het door het uitvoeren van een *dictionary* aanval tegen de webserver en analyseert het resultaat.   
Hiermee heb ik als doel om alle pagina’s te verkrijgen die bestaan op de webserver.

De eerste poging leidde tot niks, omdat de webserver beschermd was tegen een dergelijke aanval:  


Figuur 2 protection against dictionary attack(s)

Ook met de parameter voor extra delay leverde de *DIRB* niks gevoeligs op.

# Intrusion

Volgend op de *reconnaisance* komt de *intrusion*. De *intrusion* houdt in dat er gekeken wordt wat er met de *reconnaisance* gedaan kan worden en waar de mogelijke exploits te vinden zijn.

## Webserver

Nadat ik op de webserver rond gekeken had en de source code(F12 – *inspect* element) bekeken had viel me een aantal dingen op.

### Rss Feed

Zo zag ik dat [RSS](https://www.wat-is-een-rss-feed.nl/" \l ":~:text=De%20term%20RSS%20staat%20voor,favoriete%20websites%20hoeft%20te%20bezoeken) gebruikt werd op de website, alleen wist ik in eerste instantie niet welke versie. Om die reden kon ik niet overgaan naar een exploitatie fase, hier dient namelijk een versie voor nodig te zijn.  
Na een goede zoektocht ben ik tegengekomen in de source code dat er *RSS 2.0* werd gebruikt. (<http://10.10.10.206/CuteNews/rss.php>)

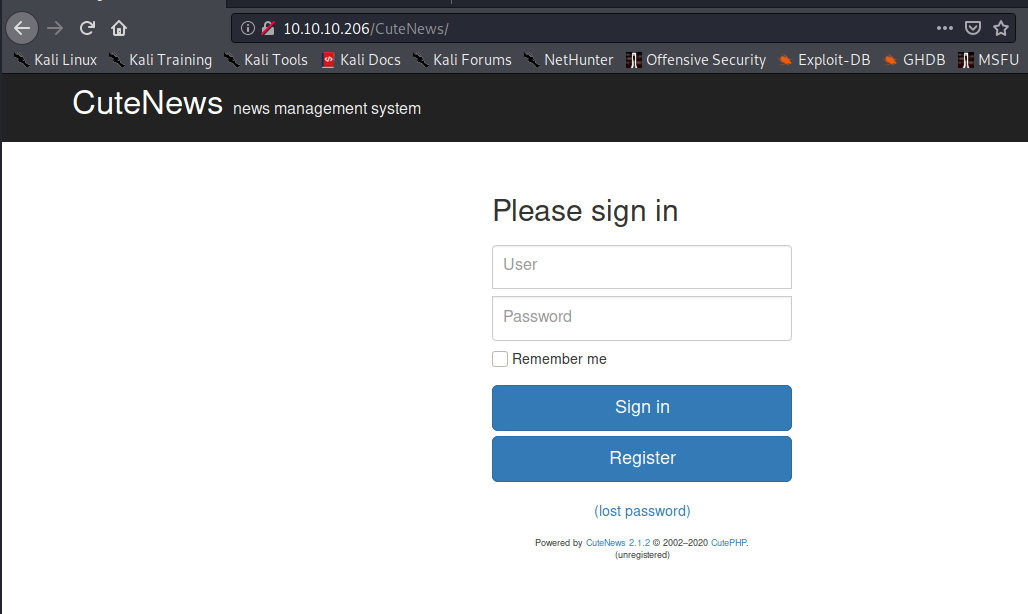


Figuur 3 RSS version

Echter, bleek dit in de exploitatie fase een *rabbit hole* voor mij te zijn.

### CuteNews

Wat me opviel aan de URL van de *rss.php* was dat er *CuteNews* voor stond. Toen ik me hier in verdiepte bleek dit een management systeem te zijn, als een dergelijk systeem een verouderde versie heeft kan deze exploitable zijn.

Toen ik naar <http://10.10.10.206/CuteNews/> ging, kwam ik op de inlogpagina van dit systeem.

Figuur 4 CuteNews index.php

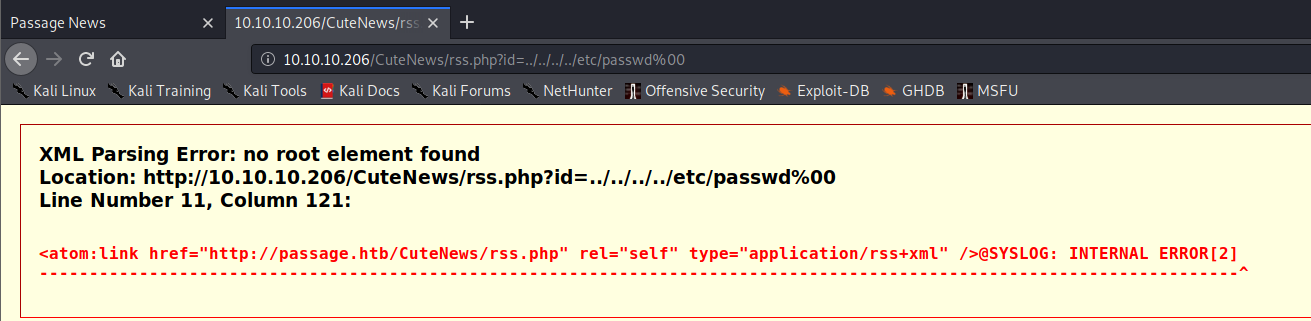
Wat me hier meteen opviel was dat onderaan de pagina het versienummer stond (CuteNews 2.1.2).

# Exploitation

In de *Exploitation* fase proberen pentesters gevonden kwetsbaarheden uit te buiten door exploits uit te voeren. Exploits zijn ontwikkeld om, bijvoorbeeld gevoelige informatie te verzamelen of pentesters de mogelijkheid geven tot het binnendringen van het besturingssysteem.  
De databank voor alle publiek bekende kwetsbaarheden, die ik gebruik is [exploit-db](https://www.exploit-db.com/).

## Path traversal

Toen ik path traversal wilde toepassen op de *rss.php* kreeg ik een *Interal server error:*



Figuur 5 internal server error

Hier kon ik met mijn kennis op dat moment nog niet veel mee, dus ben ik verder gegaan met andere mogelijke exploits.

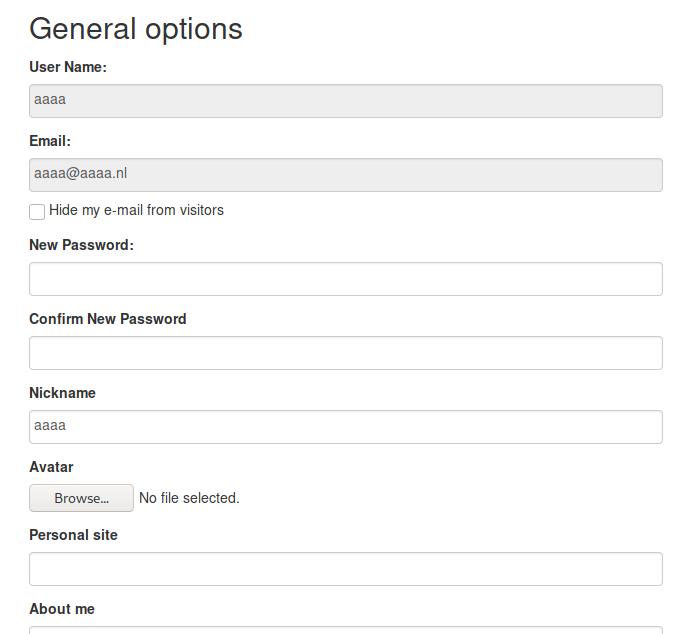
## RSS Feed

In de exploit-db heb ik gezocht naar exploits omtrent RSS feed 2.0. Veel van deze exploits heb ik geprobeerd, maar niks leidde tot resultaat.  
Op dat moment zat ik even vast en wist ik niet wat ik verder nog kon doen, dit was voor mij dus een *rabit hole*.

## CuteNews

Nadat ik me verdiept had in CuteNews zag ik wel een aantal mogelijke exploits in de database staan. Na een paar geprobeerd te hebben had ik eindelijk succes. Namelijk met de exploit die [hier](https://www.exploit-db.com/exploits/48458) te zien is.

**Stap 1**Allereerst heb ik een account aangemaakt bij CuteNews en ben naar mijn profiel gegaan.



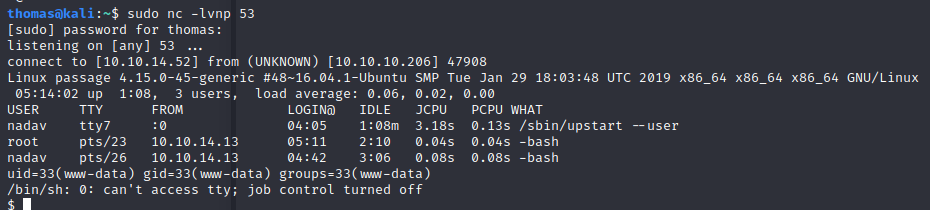
**Stap 2**Ik heb een bestand aangemaakt genaamd “empty.php.jpg”. Dit bestand begint met *magic bytes*, dit houdt in dat de server denkt dat het bestand een image is. Dit komt doordat de eerste regel van het bestand bestaat uit bytes van een image. Echter, kan je hier van profiteren door achter deze bytes code te zetten. In mijn geval heb ik hier een *reverse shell* in gezet.

**Stap 3**Vervolgens heb ik dit bestand geselecteerd als avatar en gesubmit. Echter, heb ik deze request onderschept en de naam veranderd van “empty.php.jpg” naar “empty.php”. Zo kan ik dit bestand uitvoeren als een PHP script.



Figuur 6 CuteNews/uploads

**Stap 4**Open het bestand vanuit <http://10.10.10.206/CuteNews/uploads> en de reverse shell wordt geopend.[[1]](#footnote-1)



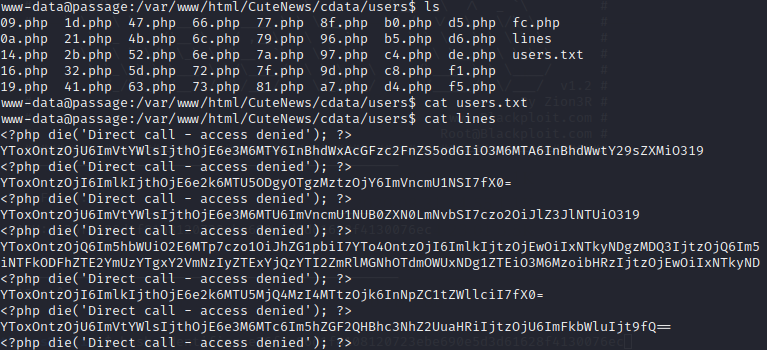
Figuur 7 reverse shell

# Privilege Escalation

In de *Privilege escalation* fase wordt er misbruik gemaakt van een bug, ontwerpfout of foute configuratie in het besturingssysteem van het doelwit. De bedoeling is om zo meer toegang te krijgen tot bronnen(bijvoorbeeld bestanden) die normaal gesproken worden beschermd tegen een toepassing of gebruiker. In deze fase is de aanvaller al in een zogeheten *reverse shell*[[2]](#footnote-2).

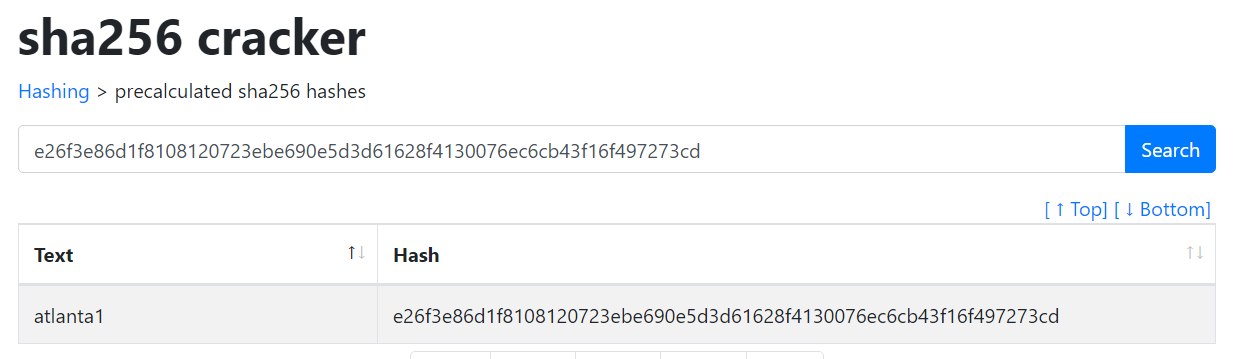
## User

Om de user te krijgen ben ik op zoek gegaan naar gevoelige informatie in bestanden. Hier heb ik heel lang op gezeten, omdat ik vergeten was dat er ook verborgen bestanden waren.  
Zo was er een *ssh* folder met veel gevoelige informatie hierin.



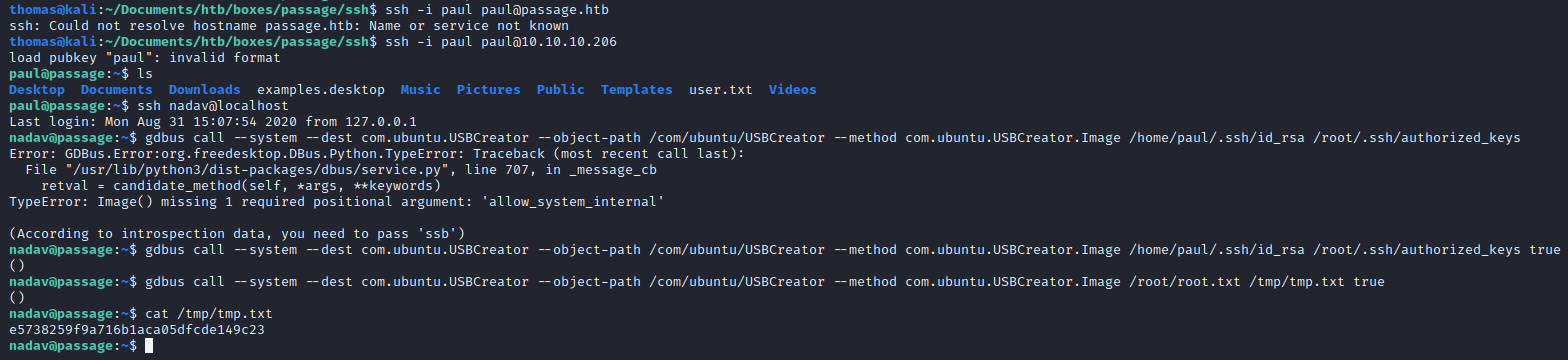
Figuur 8 lines bestand

Dit bestand bevat Base64 gedecodeerde informatie over de users. Onder andere stond hierin ook de wachtwoorden die gehasht zijn. Dit heb ik via een online tool weten te decrypten.



Figuur 9 decrypt user wachtwoord

Zo kon ik ook de **user.txt** achterhalen in zijn home directory.

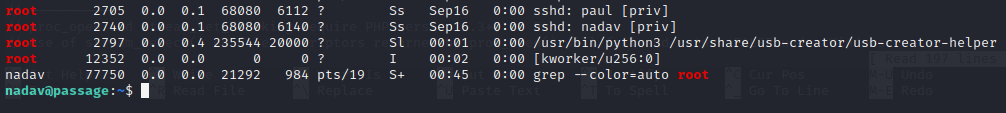
Aangezien er met dezelfde SSH sleutels gewerkt werd, kon ik gemakkelijk via **ssh user@10.10.10.206** naar de andere user wisselen. Zo kon ik zelfs ook een remote connectie leggen vanaf mijn eigen terminal.

Figuur 10 ssh via eigen terminal

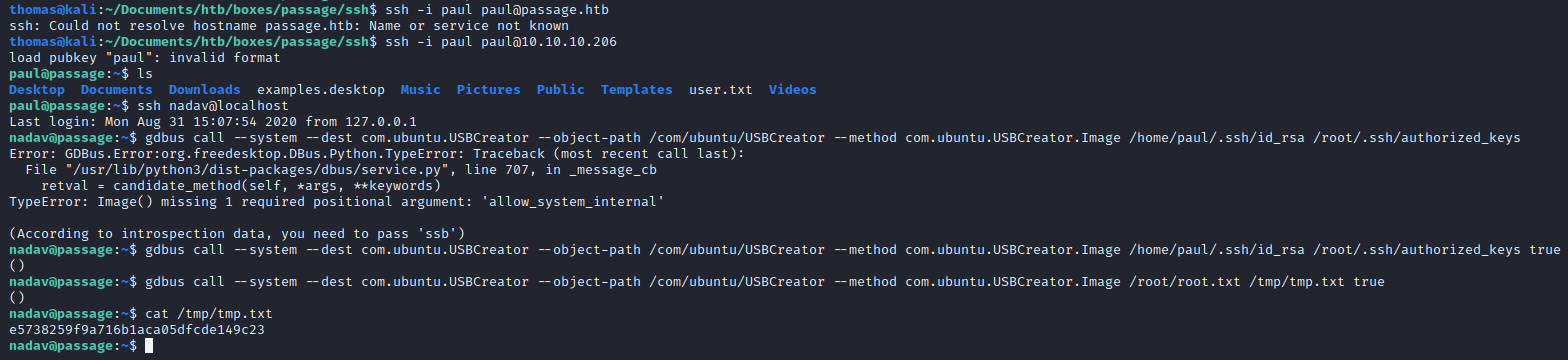
## Root

Om de root te krijgen heb ik gekeken hoe onder andere

* Hoe de privileges zijn ingesteld? **Linpeas.sh[[3]](#footnote-3)**
* Wat kan ik met sudo uitvoeren? **sudo -l**
* Welke processen worden als root uitgevoerd? **Ps -aux | grep root**

Bij *ps -aux* en via *Linpeas.sh* viel me iets op namelijk het volgende:

Figuur 11 ps-aux

Hier staat de usb-creator-helper wordt uitgevoerd als root, daarom ben ik gaan kijken of ik hier een exploit voor kon vinden. Met dank aan [deze](https://unit42.paloaltonetworks.com/usbcreator-d-bus-privilege-escalation-in-ubuntu-desktop/) exploit heb ik de root,txt kunnen verkrijgen, zie Figuur 12.

Figuur 12 root.txt

Wat deze exploit simpel weg doet is gebruik maken van de *gdbus*, die op zijn beurt de optie heeft bestanden to kopiëren. Doordat er geen gebruik gemaakt wordt van enige vorm van validatie(wachtwoord o.i.d.) kan de gebruiker bestanden overschrijven, uitlezen of kopiëren als root.

# Clean-up

Alle benodigde tools of bestanden die door mij op het systeem zijn gezet worden netjes verwijderd.

1. Deze shell heb ik geüpgraded naar een betere shell via [deze](https://medium.com/bugbountywriteup/pimp-my-shell-5-ways-to-upgrade-a-netcat-shell-ecd551a180d2) site [↑](#footnote-ref-1)
2. Een interactieve *shell* van het doelwit, verkregen door een uitgevoerde succesvolle exploitatie [↑](#footnote-ref-2)
3. Een script om privileges mee te controleren op een Linux systeem [↑](#footnote-ref-3)