**Machine**

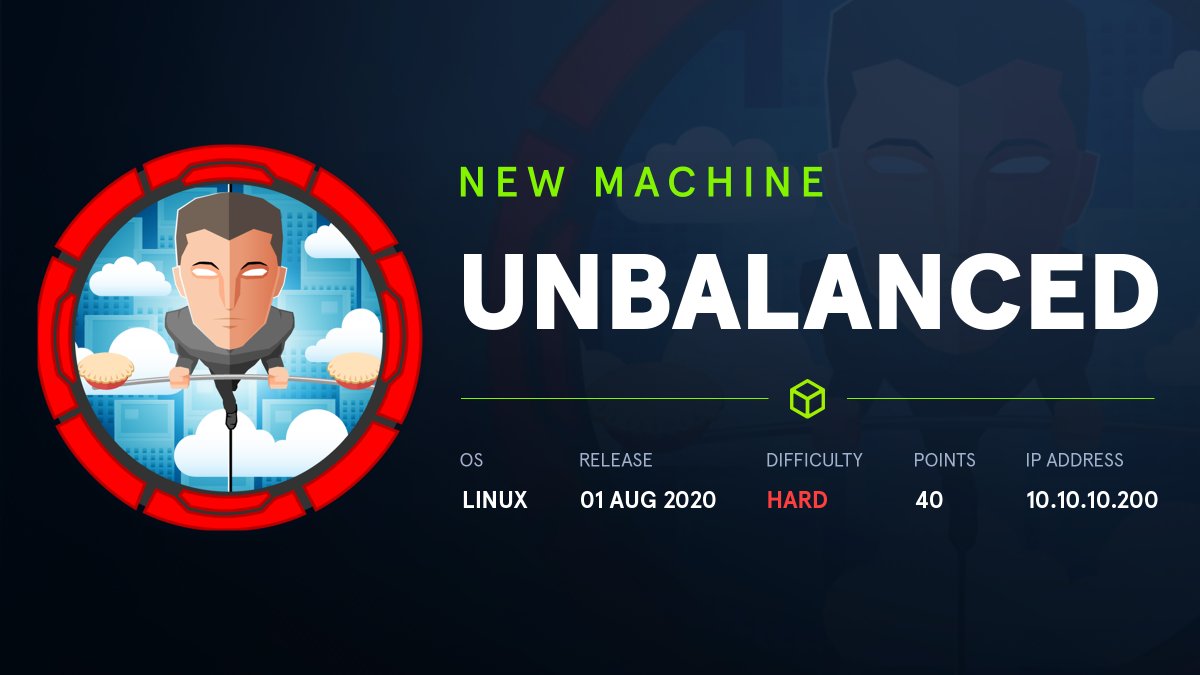
Unbalanced

**Date of done**

17-09-2020

**Author**

Polarbearer & GibParadox



**Omschrijving**

Unbalanced is een Linux-based machine in Hack The Box. Deze box heeft in Hack The Box een *hard* rating. Vanuit de gebruikers die deze machine gedaan hebt verschilt de rating tussen *medium* en *hard*. Hij is 01-07-2020 uitgekomen en daarom is deze machine vrij nieuw en heeft die relatief weinig *owns.*

**Moeilijkheidsgraad**

Hard

**Vlaggen**

Gebruiker: *<md5>*

Root: *<md5>*

Inhoud

[Enumeration 8](#_Toc56431684)

[Informatie 9](#_Toc56431685)

[Exploitation 10](#_Toc56431686)

[Privilege Escalation 11](#_Toc56431687)

[Extra informatie 12](#_Toc56431688)

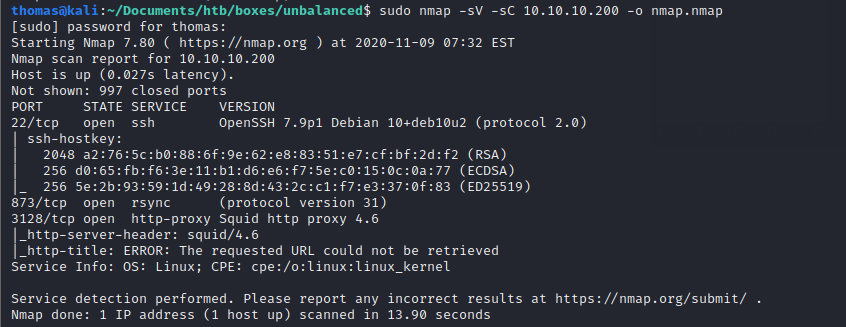
# Enumeration

*Enumeration* is een proces beginnend aan een penetratietest (in dit geval het hacken van een box). In deze fase wordt een actieve connectie met het doelwit gemaakt om potentiële aanvalsmogelijkheden/kwetsbaarheden te ontdekken. Dit kan dan gebruikt worden voor exploitatie in een verdere fase van de penetratietest.

## 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 Squid http proxy draait en dat er een rsync op staat op poort 873.

## 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.

Dit leverde niks nuttigs op.

# Foothold

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

## Squid http-proxy

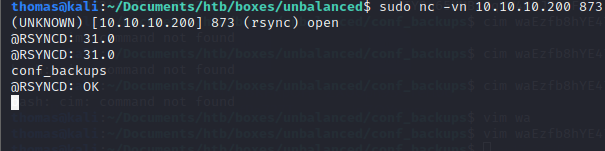
Squid is een veelgebruikte proxyserver die onder de GNU GPL is vrijgegeven. Squid is in 2001 ontwikkeld voor Unix/Linux-omgevingen en wordt als stabiele en betrouwbare software gezien.

Deze proxy bevat een exploit: [Squid exploit 4.6](https://www.secpod.com/blog/alert-squid-proxy-remote-code-execution-vulnerability-cve-2019-12527/). Echter naar hier even naar gekeken te hebben bleek dit een *rabit hole*, want hier was geen namelijk geen PoC voor te vinden.

## Rsync

Rsync is een computerprogramma dat bestanden en directory's synchroniseert van de ene computer naar de andere.

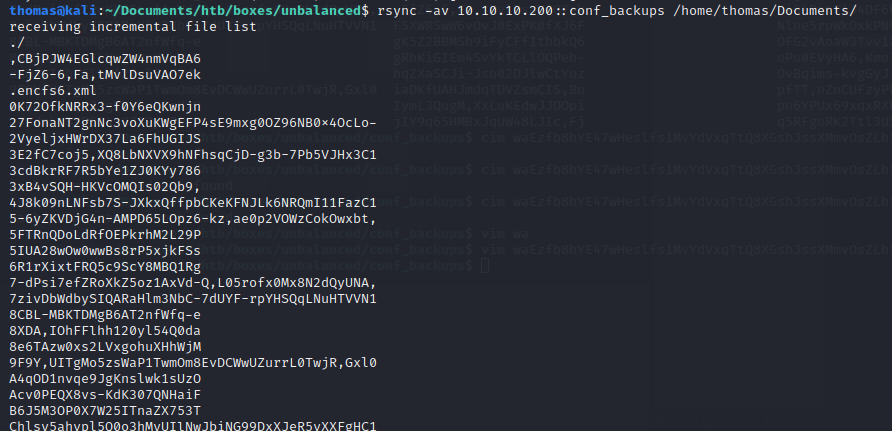
Via rsync heb ik geprobeerd bestand op te halen via het onderstaande commando, maar dit leverde niet veel op.



Figuur 2 nc rsync

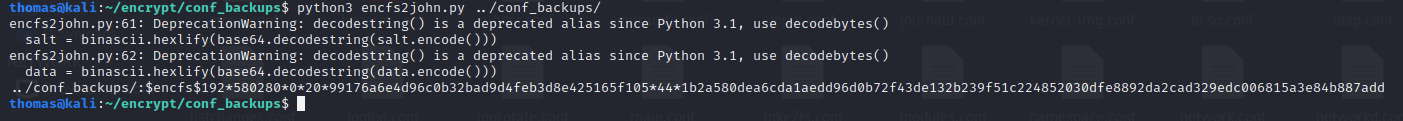
Via het onderstaande commando is het me wel gelukt om de bovenstaande folder op te halen. Deze bestanden zijn echter wel allemaal ENCFS encrypted.



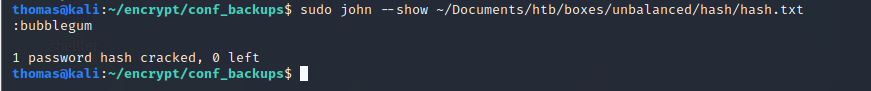


Na research gedaan te hebben, ben ik erachter gekomen dat de ENCFS een password nodig heeft. Dit wachtwoord kan gekraakt worden door het <encfs2john.py>.

# Exploitation

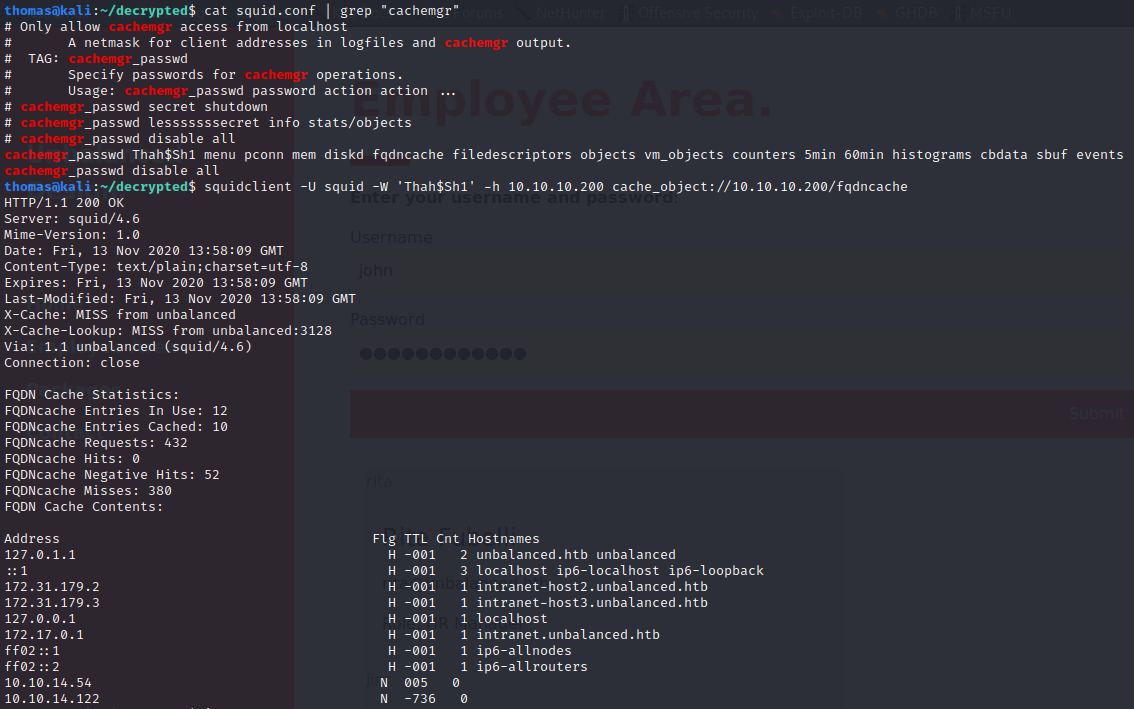
Om dus deze ENCFS folder te kraken heb ik de bovenstaande tool gebruikt, deze tool gebruikt john om de hash te krijgen van deze folder.

Nu kan via *john* het wachtwoord opgehaald worden die gekraakt is vanuit de hash die verkregen is.

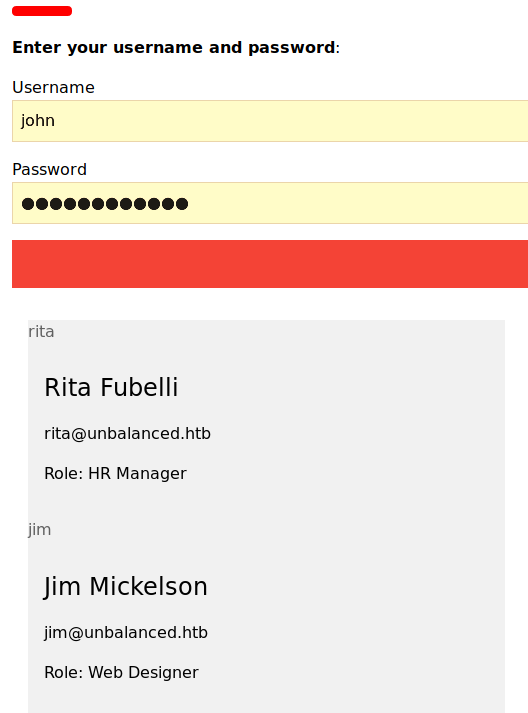


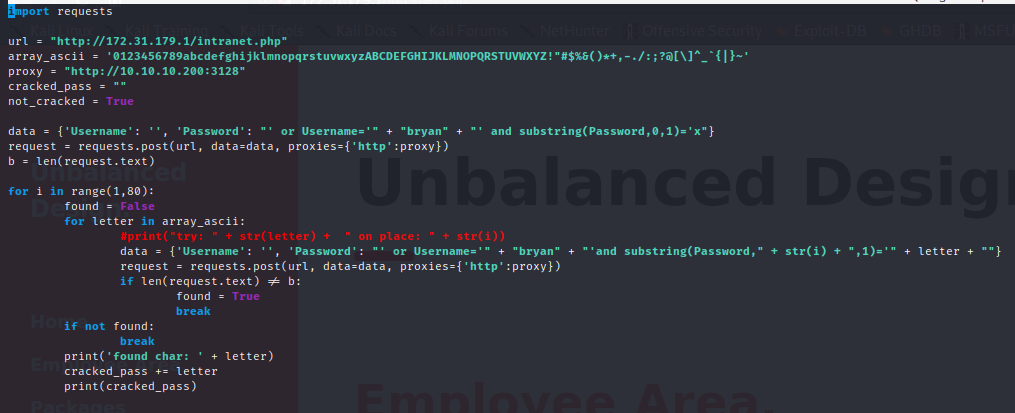
In de decrypte folder staat een bestand genaamd *squid.conf*. Hierin staat veel informatie omtrent domeinen, wachtwoorden etc. Na lang gezocht te hebben is dit het bestand wat ik nodig had. In de kwetsbaarheid die ik eerder gevonden had over de *cachemgr* in de Squid http-proxy.

Om die reden ben ik naar die syntax gaan zoeken in het bestand. Hierin staat een woord dat lijkt op een wachtwoord, namelijk **Thah$Sh1**. Bovendien is er een domein, genaamd **intranet.unbalanced.htb**.

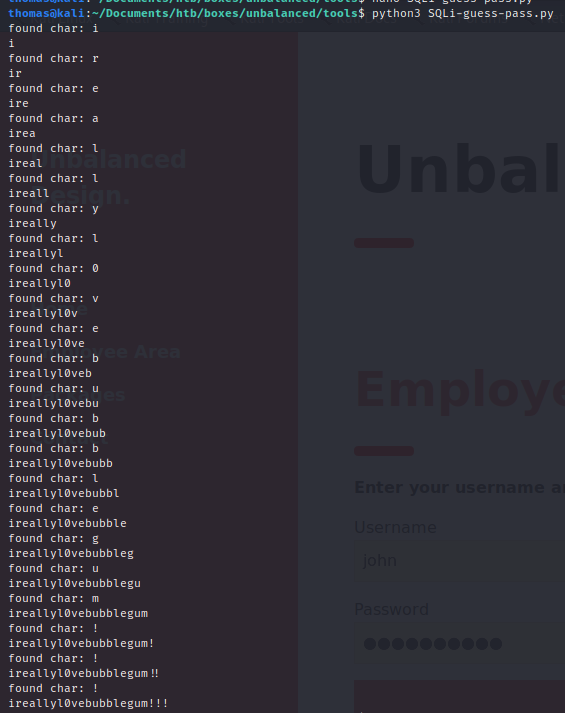
Met de gegevens in dit bestand heb ik een via de *squidclient* een request gemaakt naar de proxy en de cache uitgelezen. Deze cache is hieronder te zien:

In dit bovenstaande lijst zie je dat er 2 hosts staan, host 2 en host 3. Mist hier iets? Ja, host 1. Als je naar deze site gaat, heb je een zelfde site als die andere 2 hosts. Dus ben ik gewoon gaan kijken naar de site. Er was alleen een inlogscherm, maar hier was wel SQL injection mogelijk(**1’ or ’1’=’1**.



Om een privilege escalation toe te passen zijn er naast de username ook passwords nodig. Nu ik weet dat er een SQL injection mogelijk is, ben ik op zoek gegaan naar een script om een wachtwoord te achterhalen. Dit is echter wel alleen mogelijk als het plaintext wordt opgeslagen. Toch wil ik dit een kans geven en heb een script gemaakt hiervoor:

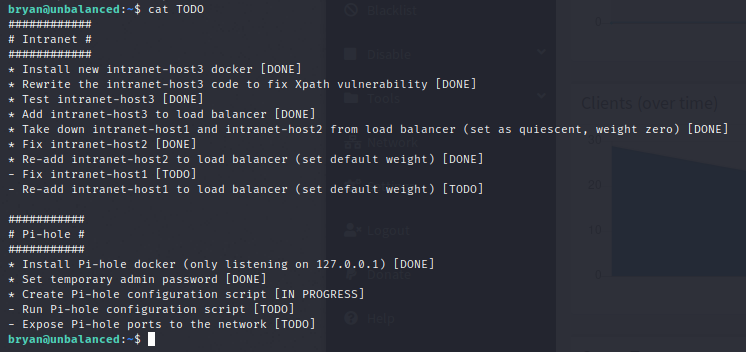
Dit script probeert alle ASCII mogelijkheden op het begin van het wachtwoord, als de site anders reageert (qua bytes lengte) dan alle andere pogingen , dan is dat de goede letter, nummer of teken. Daarna gaat die door naar de volgende plek etc. Hieruit kwam het wachtwoord.

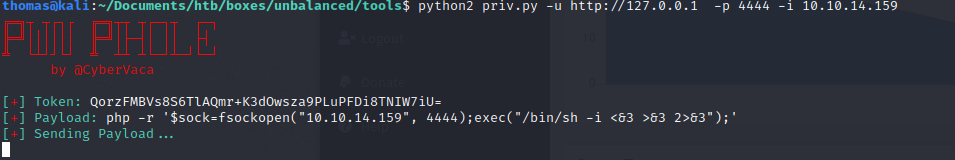


# Privilege Escalation

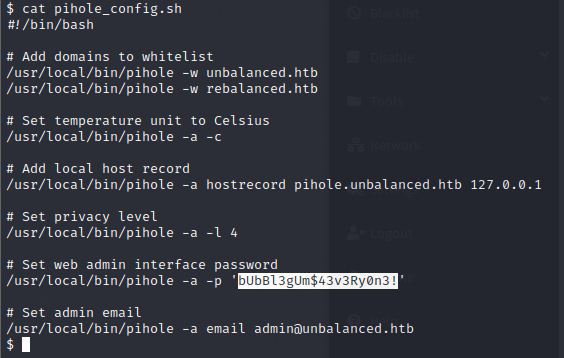
Via SSH verbinden met de server (**user: bryan, pass: ireallyl0vebubblegum!!!**), hierdoor kan de user flag behaald worden!

In de bestanden van Bryan staat een TODO. Dit bestand geeft aan dat er een pi-hole runt en dat er een tijdelijk admin wachtwoord gezet is(!).

Om deze pi-hole te exploiteren, is gebruik gemaakt van het tijdelijke admin wachtwoord. In deze pi-hole zat een kwetsbaarheid, deze ging met name om het Mac-adres input field. Hieruit kon namelijk een reverse shell worden opgezet.



Hiermee kreeg ik een user shell. Terug kijkend naar de TODO lijst, stond hierin dat het configuratie script van pi-hole nog uitgevoerd moet worden. Nu staat deze tussen de bestanden op de huidige shell het configuratie bestand. Hierin staat een web admin interface wachtwoord.



Dit was een rabit hole in mijn geval, want ik had helemaal niet geprobeerd dit als root wachtwoord te gebruiken. Dit bleek het wel te zijn!

