**Machine**

Worker

**Date of done**

22-09-2020

**Author**

ekenas



**Omschrijving**

Worker is een Windows-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 is e rating tussen *medium*. Hij is op 15-08-2020 uitgekomen en daarom is deze machine vrij nieuw en heeft die weinig *owns.*

**Moeilijkheidsgraad**

Medium

**Vlaggen**

Gebruiker: 1b52f826f6e1b037f7265b20bc25148d *<md5>*

Root: 6012370fc6382c6930d904f8af9be8f4 *<md5>*

Inhoud

[Introduction 3](#_Toc61430417)

[Reconnaissance 4](#_Toc61430418)

[Nmap 4](#_Toc61430419)

[Fuzzing 4](#_Toc61430420)

[Intrusion 6](#_Toc61430421)

[Webserver 6](#_Toc61430422)

[Svnserve 6](#_Toc61430423)

[TRACE-methode 6](#_Toc61430424)

[Exploitation 7](#_Toc61430425)

[Webserver 7](#_Toc61430426)

[TRACE-methode 7](#_Toc61430427)

[Svnserve 8](#_Toc61430428)

[Privilege Escalation 12](#_Toc61430429)

[User 12](#_Toc61430430)

[Clean-up 15](#_Toc61430431)

[Extra informatie 16](#_Toc61430432)

[Svnserve 16](#_Toc61430433)

[Clean-up 17](#_Toc61430434)

# 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.

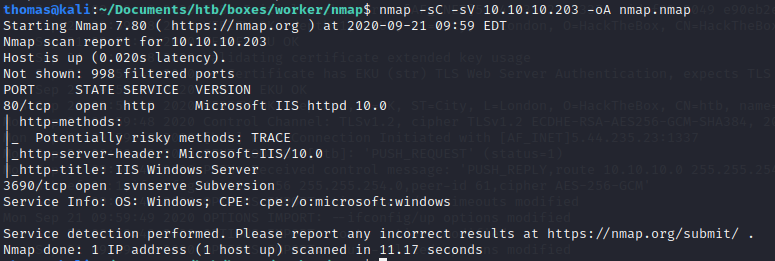
# Reconnaissance

*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 nmap

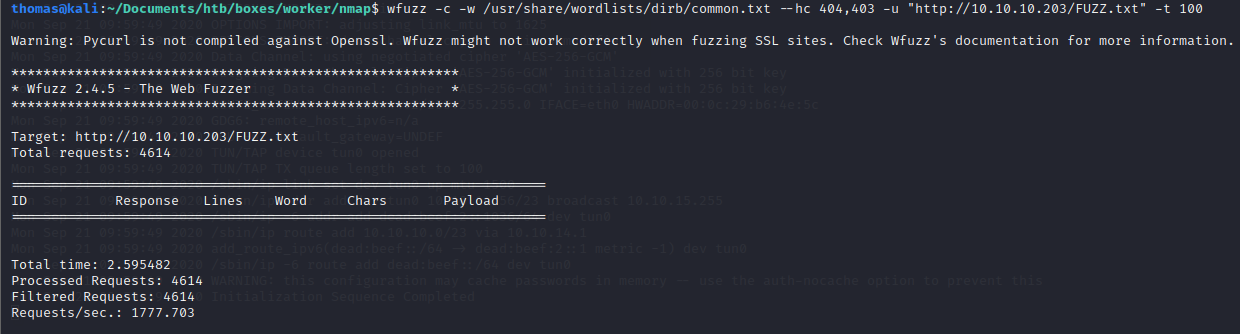
Met de output is te zien dat er een webserver draait op poort 80 (<http://10.10.10.203:80>). Ook staat de poort 3690 open voor *svnserve*, aangezien ik dit nog niet kende ben ik hier ingedoken : Svnserve.

Wat mij ook opviel was dat de TRACE methode een risico vormt op de website. De trace-mogelijkheid staat standaard aan op vrijwel alle webservers. Het commando stuurt direct dezelfde data terug als de client verzonden heeft, en is net als pingen bedoeld om een connectie te testen. Dit kan een gevaar vormen en dus de aanvalsvectoren van hackers vergroten

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

## Fuzzing

Om een website te fuzzen heb ik WFUZZ gebruikt, dit 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.

Het resultaat van de fuzzing is:

Figuur fuzzing

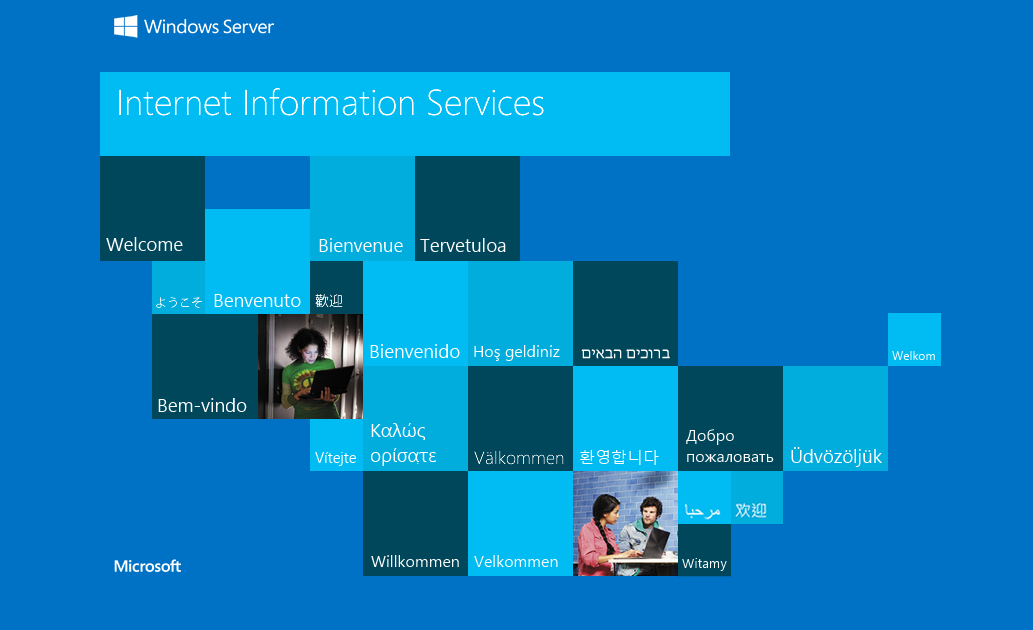
Hier is niks uitgekomen en hier liep de fuzzing van de website dood.

# 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

De webserver is een one-page, de pagina bevat niks aan gevoelige informatie. De enige informatie die ik kon achterhalen was dat het een Microsoft-IIS 10.0 is, wat in houdt dat deze vrij nieuw is.



Figuur webserver

## Svnserve

In de *nmap* scan kwam de poort 3609 – *svnserve*  naar voren, dit viel mij op en ben me hier in gaan verdiepen. ([svn command](https://www.thegeekstuff.com/2011/04/svn-command-examples/))

Dit commando kan gebruikt worden om onder andere code en bestanden in te zien. Aangezien deze open staat vormt dit wel een gevaar voor de ontwikkelaar, maar geeft het mij jeukende vingers.

## TRACE-methode

De TRACE-methode kan een gevaar vormen op de website als het door hacker uitgebuit kan worden. Deze methode kan worden gemanipuleerd door wellicht

# Exploitation

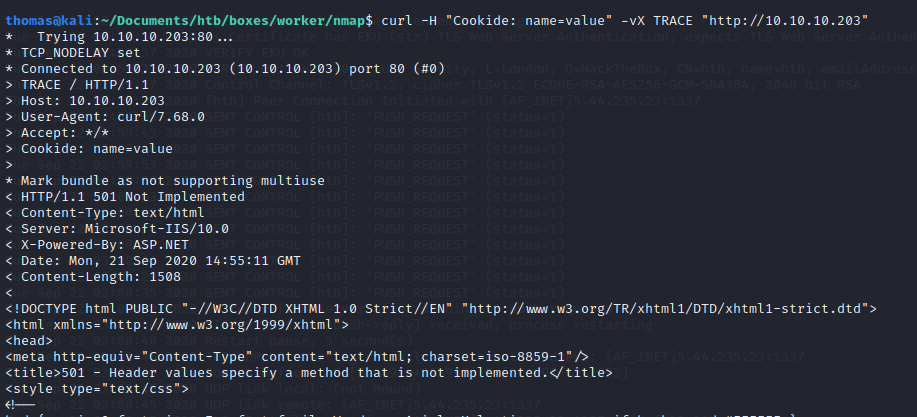
In de *Exploitation* fase proberen pentesters gevonden kwetsbaarheden uit te buiten door exploits uit te voeren. Exploits zijn ontwikkeld om, bijvoorbeeld gevoelige informatie te verzamelen of pentesters de mogelijkheid geven tot het binnendringen van het besturingssysteem.  
De databank voor alle publiek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

## Webserver

De webserver zelf is niet exploitable aangezien het een one-pager is en deze is goed geüpdatet is naar Microsoft-IIS 10.0. De gevonden exploits waren allemaal PoC’s en niet te gebruiken.

## TRACE-methode

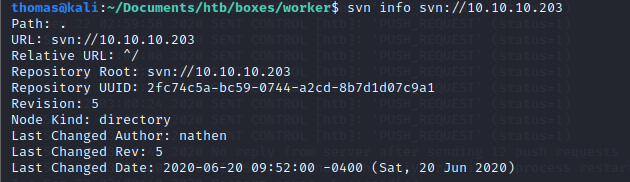
Om te kijken of de TRACE-methode een kwetsbaarheid vormt ben ik gaan kijken of mijn request inderdaad werd geëchood. Zoals in Figuur 4 te zien is, zie je dat dit inderdaad het geval is. Echter, heb ik er zelf niet veel mee kunnen doen. Ik zag meer potentieel in het versiebeheer systeem.



Figuur TRACE-methode

## Svnserve

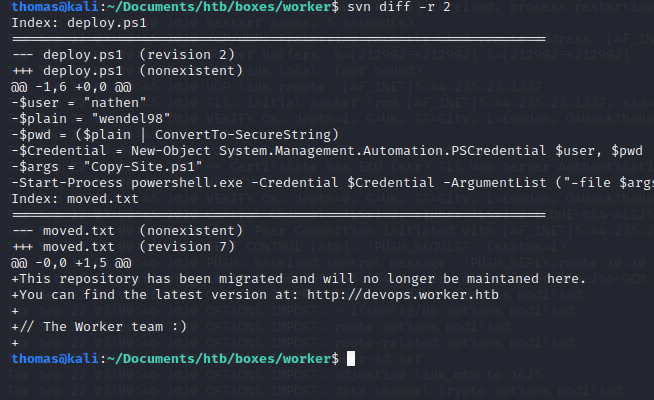
Kijkend naar het versiebeheersysteem heb ik me eerst verdiept in de mogelijkheden die **svn** biedt. De eerste optie die ik heb geprobeerd was om de **info** van het versiebeheer systeem op te halen.



Figuur info of svn

Hierin is veel informatie te zien, maar wat vooral opvalt is de “Last changed Author”. Dit houdt in dat er een gebruiker is met de naam Nathen.

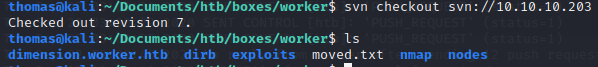
Wat **diff** doet is kijken naar het verschil tussen verschillende commits van de repository.



Figuur svn diff

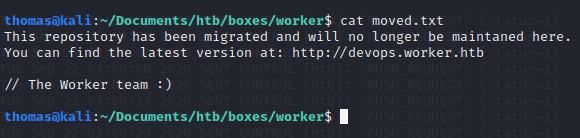
Wat hier duidelijk naar voren komt zijn de *credentials* van Nathen (namelijk wendel98). Bovendien is te zien dat de *moved.txt* is toegevoegd aan deze commit. Dit bestand ziet er interessant uit, maar er moet dus een manier zijn om deze bestand op mijn systeem te krijgen.   
Verder is te zien dat e

Toen ik me heb verdiept in *svn,* zag ik dat er een **checkout** functie was. Wat deze functie doet is de huidige repository ophalen en downloaden in je huidige directory.



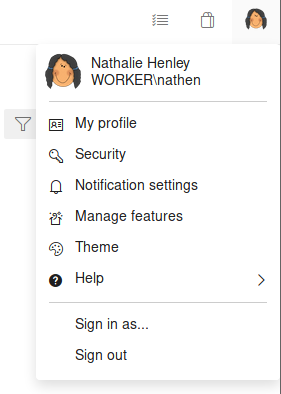
Figuur 7 svn checkout

Op deze manier kon ik bekijken wat in het bestand zat, namelijk:



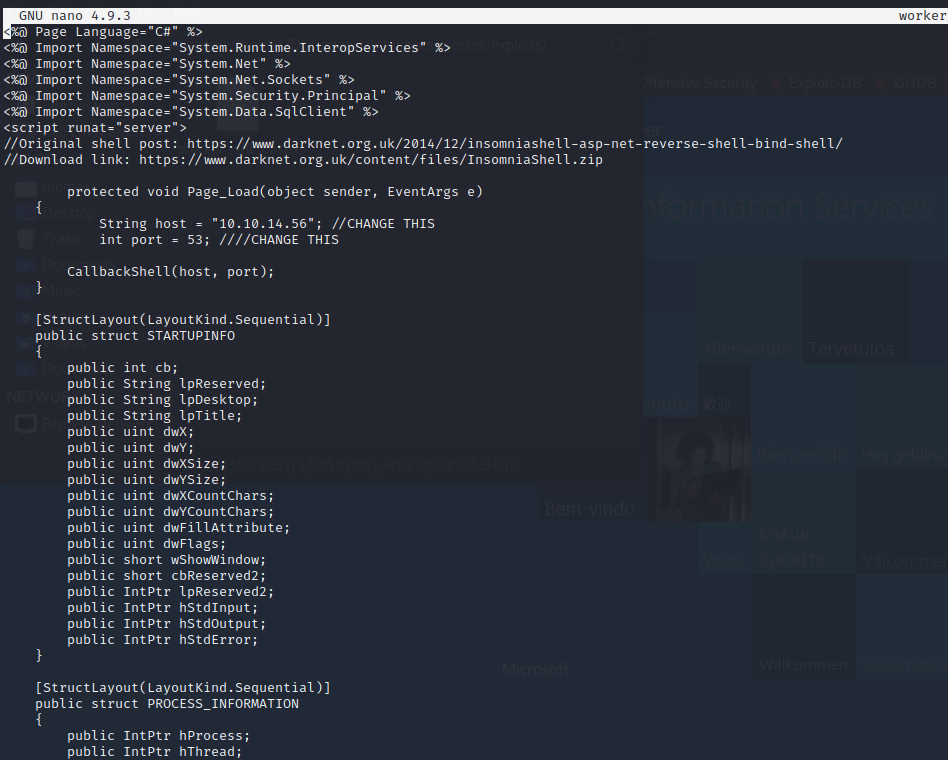
Figuur moved.txt

Dit document geeft de informatie weer dat er op de **devops** omgeving gewerkt wordt. Nadat ik dit toegevoegd heb aan mij hosts, kon ik deze website bereiken. Hier had ik wel inloggegevens voor nodig, dus heb ik die van Nathen geprobeerd. Dit lukte met succes, zie Figuur 9.



Figuur ingelogd op de devops names Nathen

Nu ik op de **devops** omgeving zat van Azure moest ik een manier vinden op een reverse shell op te zetten. Nu wist ik nog van de *Enumeration* fase dat ze met ASP.NET deze website opgezet hebben, dus ben ik gaan kijken naar een .aspx reverse shell.



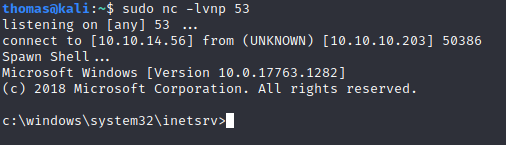
Figuur 10 .aspx reverse shell

Deze reverse shell kon ik via de **devops** omgeving uploaden. Dit heb ik gedaan door een branche aan te maken en daarna een bestand uploaden in dit geval de reverse shell. Daarna ben ik naar de URL gegaan, zie figuur 11. Hiermee wordt de reverse shell door middel van netcat geactiveerd.



Figuur activeer reverse shell

De reverse shell en dus exploit is met succes uitgevoerd, zie het onderstaande figuur.

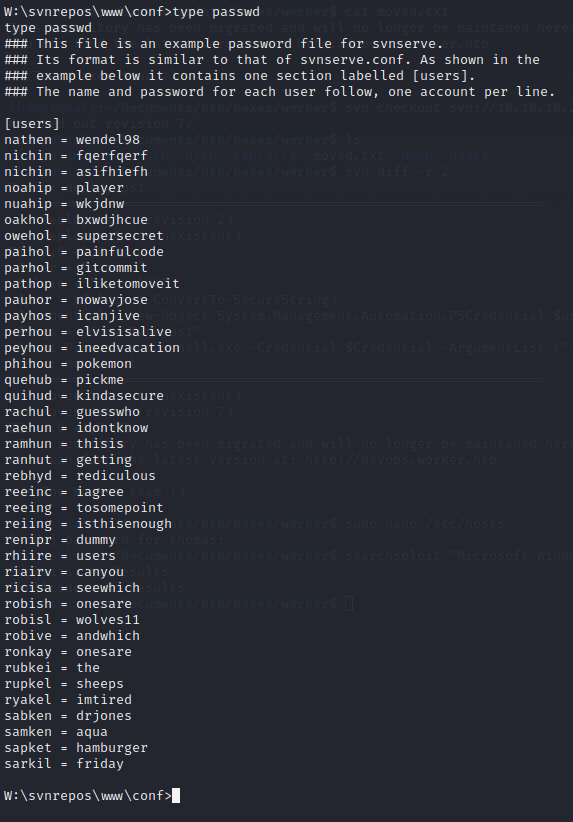


Figuur 12 reverse shell

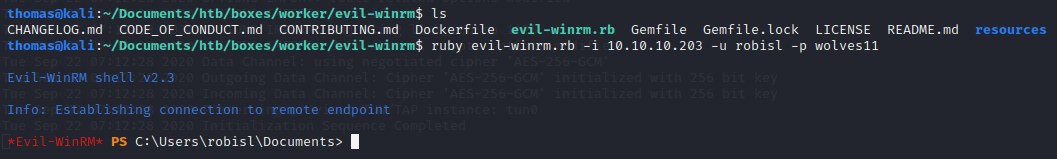
# Privilege Escalation

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*[[1]](#footnote-1).

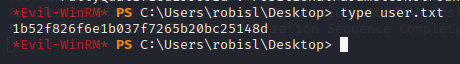
## User

Na veel zoekwerk ben ik bij een bestand gekomen genaamd *passwd.* Dit bestand stond vol met alle gebruikers en hun wachtwoord! Op die manier kon ik ook die van *robisl* achterhalen en via hem die *user.txt* achterhalen. *Robisl* was namelijk een gebruiker die een account had op de Windows machine.

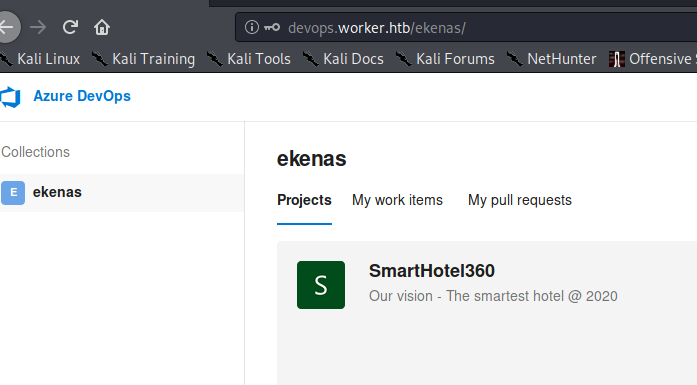
Figuur passwd bestand

Dit was mijn eerste Windows machine en ik kon geen manier vinden om via de command line van gebruiker te wisselen. Na wat zoekwerk ben ik op de “Evil Windows Remote management” tool gekomen. Deze is [hier](https://kalilinuxtutorials.com/evil-winrm-hacking-pentesting/) te vinden. Via deze tool kon ik op de host inloggen via de credentials(die ik dus had)

Figuur Evil-WinRM

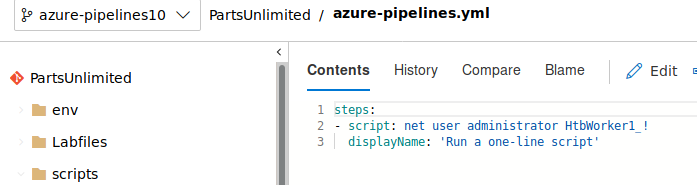
 Via hier kon ik de *user.txt achterhalen, zie figuur 15*

Figuur user.txt

Door middel van het *robisl* account kon ik ook hiermee inloggen op de **devops** omgeving.

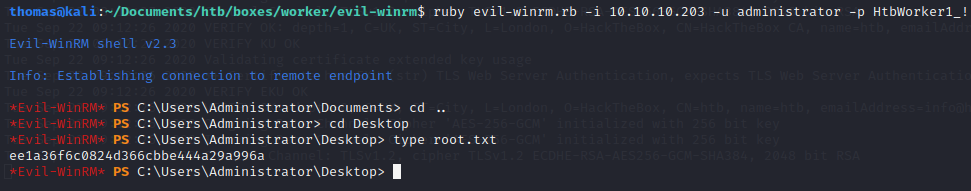
Figuur omgeving robisl

*Robisl* had de mogelijkheid om een pipeline aan te maken en deze te activeren. Nu was de truc om via deze nieuwe pipeline een yaml bestand op te zetten en via dit yaml bestand een script uit te voeren. Via dit script zou je dus het wachtwoord van de admin kunnen veranderen, zie figuur 17.



Figuur 17 yaml change admin password

Door deze succesvol gebuilde pipeline kon ik via de Evil-WinRM inloggen als admin.



Figuur admin & root.txt

# Clean-up

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

# Extra informatie

In dit hoofdstuk wordt extra informatie gegeven over onderdelen waar ik nog geen kennis van had. Deze onderdelen kunnen van belang zijn bij de daadwerkelijke exploit of gewoon verbreding van mijn algemene kennis.

## Svnserve

Toen ik deze poort open zag staan (ongeacht of ik wist wat het betekende), was het eerste wat door me hoofd ging: “exploit”. Echter, had ik geen idee wat die precies inhield. Na me erin verdiept te hebben is dit een versiebeheer systeem die via de command line kan worden gebruikt.   
Als ontwikkelaar wil je absoluut niet dat buitenstaanders deze kunnen bereiken en moet er dus voor gezorgd worden dat deze poort dicht staat.

Alleen waarom is dit dan gevaarlijk? Nou met deze tool kan je in de bestanden en eventuele code komen van de applicatie. Wellicht staan hier ook bestanden in met gevoelig informatie, dit vergroot allemaal de mogelijkheid aan aanvalsvectoren.

# Clean-up

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

1. Een interactieve *shell* van het doelwit, verkregen door een uitgevoerde succesvolle exploitatie [↑](#footnote-ref-1)