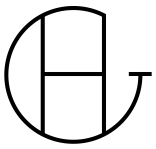


HoGent

Faculteit Bedrijf en Organisatie

# Vergelijkende studie: Vulnerability scanners

Niels Van Driessche

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Bert Van Vreckem

Instelling: —

Academiejaar: 2016-2017

Tweede examenperiode

Faculteit Bedrijf en Organisatie

# Vergelijkende studie: Vulnerability scanners

Niels Van Driessche

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Bert Van Vreckem

Instelling: —

Academiejaar: 2016-2017

Tweede examenperiode

# Samenvatting

# Voorwoord

Eerst en vooral wil ik de mensen bedanken die mij ondersteund hebben voor deze bachelorproef:

Ik heb dit onderwerp gekozen voor een paar redenen:

- Ik heb me deze vraag ook gesteld, maar nog niet veel neutrale standpunten gehoord.
- Dit onderzoek kan handig zijn voor bedrijven die hun vulnerability managment willen uitbreiden.
- Ik kan mijn passies gebruiken voor het schrijven van deze bachelorproef.
- Nessus was vroeger een deel van kali, maar is vervangen door openvas door licentie redenen, ik wil weten of de kwaliteit van beiden even hoog is.

# Inhoudsopgave

# 1. Inleiding

Security is de laatste jaren een echte trend geworden. Elk bedrijf is hiermee bezig, en ook in het nieuws komen er steeds meer berichten over beveiliging. Maar hoe zou een bedrijf hier zich best tegen kunnen verdedigen? Is een gewone firewall en antivirus niet genoeg meer?

De volgende stap is voor vele bedrijven een 'Vulnerability management' systeem. Dit is een continu proces dat bestaat uit 4 delen: 'discovery', 'reporting', 'prioritization' en 'response'. Tijdens het 'discovery' deel moet ELK systeem getest worden en in een databank geplaatst worden. Deze databank wordt gebruikt in de volgende stap 'reporting'. Hier moet er een rapport gemaakt worden zodat alle risico's in kaart gebracht kunnen worden. Eens dat de rapporten klaar zijn, moeten er 'priorities' opgesteld worden. Dit gebeurt door elk risico te bekijken en studeren wat de invloed is op het netwerk als deze misbruikt word. Uiteindelijk moeten deze risico's opgelost worden in het 'response' deel. „What is Vulnerability Management Anyway?" (2013)

Voor we beginnen uitleggen wat men hiervoor kan doen als verdediging, zuller er een paar termen uitgelegd worden dat essentieel zijn voor het verdere verloop van deze bachelorproef. Een vulnerability is een zwakheid in een stuk code of een design fout (de meeste vulnerabilities komen van verouderde software en verkeerd geconfigureerde software) waardoor het mogelijk is om een actie uit te voeren wat niet de intentie van de software en/of het design was Rouse (2005). Vulnerabilities die het mogelijk maken om informatie te krijgen van het systeem krijgen een CVE nummer. Dit is een uniek nummer voor een publieke vulnerability dat begint met het jaar waarin deze ontdekt werd. Deze nummers en namen zijn nodig omdat voor dit systeem bestond, elk bedrijf een andere naam had voor een vulnerability. Dit was onduidelijk en zorgde voor veel verwarring „About CVE" (g.d.). De acties dat gebruik maken van een vulnerability om bepaalde commandos uit te voeren

op een machine (Arbitrary code execution) of een server niet meer toegankelijk maken voor legitieme gebruikers (Denial of Service), noemen we exploits Rouse (2014).

Tools dat voor vulnerability management gebruikt kunnen worden noemen we scanners. In deze bachelorproef zal ik alleen de 'Vulnerability scanners' bespreken. Dit is een beveiligingstechniek dat fouten in een systeem kan opsporen „Vulnerability Scanning" (g.d.) over een netwerk. Deze tools worden zowel gebruikt door 'black hats' als 'white hats'. Het verschil tussen deze 2 is dat de 'white hat' ethisch correct is. Deze zal op voorhand toegang vragen om een systeem te mogen testen, en als deze test lukt het bedrijf melden wat er mis is. De 'black hat' vraagt geen toestemming en gebruikt de testen om binnen te dringen op een systeem en de data hiervan te verkopen, of het systeem te laten crashen Hoffman (2013). Voor een volledig vulnerabiliy management bij te houden, moet men ook nog andere scanners implementeren zoals een 'web application scanner' (bv. Arachni). Deze scant een website op mogelijke problemen als SQL injecties en cross-site scripting.

Een SQL injectie, is een vulnerability dat voorkomt als men data moet invullen op een website dat een actie uitvoert op de databank. Deze kunnen gaan van een wachtwoord van een gebruiker omzeilen, tot toevoegen of verwijderen van data **acunetix**

Cross-site scripting (vaak afgekort als XSS), is een techniek waarbij een hacker legitieme gebruikers kan aanvallen via een legitieme website. Deze techniek wordt vooral misbruikt bij javascript, omdat deze op bijna elke site aanwezig is. Dit gebeurt als er input van een gebruiker gebruikt wordt door bijvoorbeeld HTML of JS. Hier kan de aanvaller de gebruiker bijvoorbeeld doorverwijzen naar een valse website, en daar persoonlijke data van de gebruiker proberen te krijgen „Cross-site Scripting (XSS)" (g.d.).

Het resultaat van een vulnerability scan is meestal een lijst met 'cvss' scores van de gescande targets. Een cvss score is een framework om te bepalen hoe erg een bepaalde vulnerability is. Dit word bepaald door een complexe formule waarbij er rekening gehouden wordt met de impact op de target server, welke privileges je op de server reeds moet hebben, hoe complex de vulnerability is en van waar deze vulnerability misbruikt kan worden (lokaal netwerk, internet, fysische toegang, ...). Deze scores liggen tussen 0 en 10 en hebben aan de hand hiervan een bepaald label. Volgens versie 3 van cvss zijn de labels: low (0-3.9), medium (4-6.9), high (7-8.9) en critical (9-10) „NVD CVSS Support" (g.d.). Naast een cvss score, heeft het rapport ook een oplossing voor de vulnerability (indien deze bestaat). Deze kan bestaan uit het updaten van een bepaalde service, tot een configuratie file aanpassen.

Momenteel zijn er erg veel vulnerability scanners, maar voor deze bachelorproef zal er gekeken worden naar 'openvas' en 'nessus'. De reden hiervoor is dat nessus momenteel 1 van de populairste en bekendste tools hiervoor is dankzij zijn kwaliteit/kost verhouding „Top 125 Network Security Tools" (2015), maar deze heeft voor bedrijven wel een betalende licentie nodig „Nessus Professional" (2017). Openvas daarintegen is open-source en is gebaseerd op de laatste code die van nessus is vrijgegeven. Hierdoor kunnen we ook vergelijken of het mogelijk is om een open-source product te gebruiken in plaats van betalende software, en deze dezelfde kwaliteit kan leveren. Een andere scanner die we

wouden testen genaamd CORE impact (deze wordt gezien als een van de beste en meest volledige scanners) heeft geen gratis versie en deze konden we dus ook niet testen.

## 1.1 Stand van zaken

In het volgende deel zal ik meer uitleg geven over hoe vulnerability scanners werken, ook refereer ik naar gelijkaardige onderzoeken en wat de verschilpunten zijn met mijn onderzoek. Hierna geef ik meer uitleg over nessus en openvas zelf, en wat de theoretische verschillen hiertussen zijn (zoals licenties,hardware,...). De praktische verschillen zullen we in het volgende hoofdstuk bespreken.

Elk onderzoek dat verband heeft met security is nooit lang up-to-date. Dit komt omdat deze constant veranderd en er steeds nieuwe technieken worden ontdekt waar men zich moet tegen beschermen. Een interessant voorbeeld van deze innovatieve technieken is over „Linkedin" (g.d.). Hierin slaagde een persoon om met css (Cascading Style Sheet), een webpagina van linkedin over te nemen. Dit werd mogelijk gemaakt door een bestaand css regel dat het hele scherm in gebruik nam. De aanvaller gebruikte via injecties deze regel om het hele scherm aanklikbaar te maken, en naar een wilkeurige site te gaan.

Dit onderzoek is hier dus geen uitzondering. Na een paar maanden zal sommige informatie irrelevant zijn. Dit wil zeggen dat ook vorige onderzoeken al oude informatie kunnen gebruiken, waardoor men zich hierop niet kan baseren.

### 1.1.1 Gelijkaardige onderzoeken

Volgens de meeste onderzoeken die gevoerd zijn, is core impact de scanner met recenste NVTs, en heeft deze een paar handige functies die kunnen gebruikt worden om te kijken of de gevonden vulnerability geen 'false positive' is. Maar de licentiekosten bedragen rond de 30.000$ / jaar. De meest populaire is zoals eerder vermeld nessus. „Vulnerability Scanners" (2016) && „Top 125 Network Security Tools" (2015)

Voor een vergelijking tussen openvas en nessus, is het aantal papers beperkt. Men geeft steeds dezelfde link naar het onderzoek „Nessus, OpenVAS and Nexpose VS Metasploitable" (2012). Dit onderzoek is gebeurt rond Juni 2012, voor security standards is dit dus heel outdated. Hoewel de vergelijking goed beschrijft hoe de scans gebeurt zijn en in welke environment dit gebeurt is, zijn er een aantal zaken die ontbreken. Zoals een windows target, in het onderzoek heeft men alleen metasploitable gebruikt. Ook waren de 'extra' tools voor openvas niet geinstalleerd en werd voor de nessus scan niet een volledige diepe scan uitgevoerd. In de comments werd het nessus probleem aangesproken door Paul Asadoorian. Deze werkt voor tenable en heeft een uitleg gegeven hoe je een volledige scan doet met nessus Asadoorian (2012). Merk op dat deze referentie partijdig is, sinds deze gegeven werd door een medewerker van nessus.

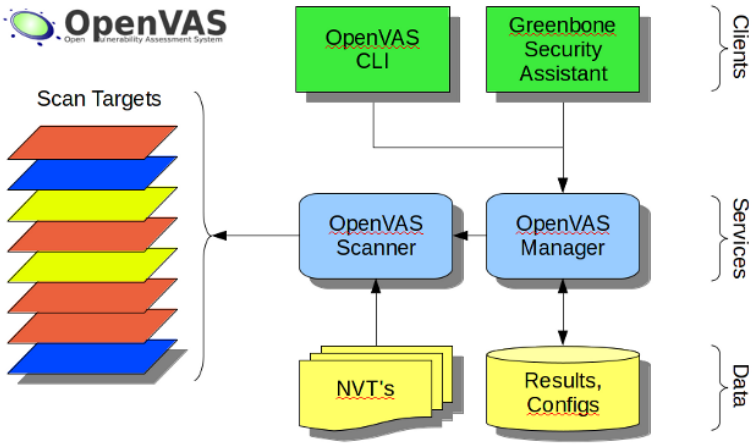
Een ander onderzoek „OpenVAS vs Tenable Nessus" (2011) gebruikt wel windows en linux targets, maar geeft geen informatie over hoe alle scans uitgevoerd zijn. Dit is een

groot probleem als men niet voor elke scanner een gelijkaardig profiel gekozen heeft. Wel is dit onderozek interessant omdat er een methode gebruikt werd om 'false positives' eruit te filteren.

Andere gevoerde onderzoeken werden uitgevoerd als nessus nog open source was en zijn dus niet toepasbaar op dit onderzoek, of waren van een te lage kwaliteit om deze te analyseren.

### 1.1.2 Hoe werkt een vulnerability scanner

Een vulnerability scanner bestaat traditioneel uit 3 delen. Een user interface, een manager en een scanner. Eerst zal er een kleine uitleg gegeven worden over hoe alle delen met elkaar werken, en daarna een gedetailleerde beschrijving hoe een target gescanned word.




„About OpenVAS" (g.d.)

De user interface (hierboven vermeld als 'Greenbone Security Assistant') is in veel gevallen een webinterface dat acties doorstuurt naar de manager, zoals een scan starten, of rapporten maken. Het is ook mogelijk om acties door te sturen naar de manager via een command line interface (hierboven vermeld als 'OpenVAS CLI'), dit is handig voor als men scans wil automatiseren.

De manager (hierboven vermeld als 'OpenVAS Manager') staat in voor de communicatie tussen de user interface en de scanner engine. Deze onderhoud ook een database waarin alle configuraties, resultaten, SCAP data, CERT data en NVT's zitten. Het is de job van de manager om deze databank up-to-date te houden zodat deze steeds een volledige vulnerability management oplossing kan bieden.

De scanner (hierboven vermeld als 'OpenVAS Scanner') staat in voor de effectieve scan van een target. Deze heeft een cache waar alle NVT's aanwezig zijn die de scanner nodig heeft om een target te kunnen scannen. Deze geeft zijn resultaten terug aan de manager.

Indien men een scan start voor een range van machines, zal de scanner engine eerst een 'alive check' uitvoeren op alle hosts in de range. Dit kan bestaan uit een simpele ICMP ping naar de host, tot een range van poorten scannen en kijken of de host reageert op 1 van de requests.

Indien de host reageert, zal er een port scan gestart worden, de tool die men hiervoor gebruikt is in veel gevallen 'nmap'. De range van poorten wordt op voorhand gedefinieerd door de gebruiker en indien er een firewall aanwezig is, zal hier ook rekening mee gehouden worden.

De scanner zal hierna een aantal dingen analyseren, zoals het besturingssysteem dat op de host draait, en welke services overeen komen met de open poorten. Als al deze feiten uiteindelijk bekend zijn, zal de scanner op basis van deze analyse NVTs naar de hosts sturen en kijken of deze vatbaar is voor bepaalde exploits „How does vulnerability scanning work?" (2015).

Uiteindelijk zal de scanner alle gevonden informatie doorsturen naar de manager, en op basis hiervan een rapport maken die leesbaar is voor de gebruiker.

### 1.1.3 Requirements

In de volgende delen, zullen we Nessus en Openvas vergelijken op een aantal vlakken dat belangerijk zijn in een scanner. Een aantal belangerijke zaken (snelheid van de scanner, accuraatheid, etc...) zullen pas besproken worden in hoofdstuk 2, omdat deze tot mijn onderzoek behoren. Eerst zal er kort een aantal zaken besproken worden zoals de geschiedenis van de scanner, de licenties, de support dat de scanner krijgt en de hardware vereisten. Hierna bespreken we de belangerijkste zaken.

Een volwaardige vulnerabiltiy scanner moet aan een paar voorwarden voldoen, het belangerijkste onderdeel zijn de NVTs. Als deze niet tijdig worden geüpdate, is de scanner niet meer betrouwbaar voor de nieuwste vulnerabilities, en geeft dit een vals veilig gevoel. Hiervoor zullen we 2 grote vulnerabilities (1 voor linux, en 1 voor windows) opzoeken en kijken wanneer beide scanners deze NVT toegevoegd hebben. Voor linux heb ik gekozen voor de 'shellshock' vulnerability (CVE-2014-6271), voor windows is dit 'Server Service Vulnerability' (CVE-2008-4250)

Uiteindelijk bespreken we ook de functionaliteiten van beide scanners. Eerst zullen we bepaalde funtionaliteiten bespreken die een betere performantie geven, of een accurater beeld geven van het systeem. Daarna bespreken we bepaalde functies die niet direct bijdragen tot het eindrapport, maar de scanner gemakkelijker laten implementeren met de bestaande infrastructuur of het grootste deel van het onderhoud automatiseren.

Voor het eerste deel gaan we na of beide scanners een 'credential scan' ondersteunen, deze test gebeurt door het inloggen op een bepaalde dienst (meestal ssh) en heeft dus ook de mogelijkheid om configuratie files te controleren op fouten Peterson (2010). Een basisvereiste is ook vaak dat scanner meerdere scans tegelijkertijd kunnen uitvoeren, Hoewel dit bij moderne scanners zo goed als steeds het geval is gaan we voor volledigheid dit ook nakijken. Uiteindelijk kijken we ook of de scanner 'agent based' scanning toestaat, dit is een klein programma dat op een host geïnstalleerd word, en commandos van van een centrale machine kan ontvangen. Deze zal een scan uitvoeren op de lokale machine, en zijn resultaten doorsturen naar de centrale machine waar deze beschikbaar is via alle

interfaces.

Uiteindelijk kijken we of de server zichzelf onderhoud (zoals NVTs updaten), en wat er nog manueel moet gebeuren. Ook kijken we of er een REST interface, of een API aanwezig is.

### 1.1.4 Nessus

#### Geschiedenis

Nessus is in 1998 opgericht door Renaud Deraison, met de bedoeling om een gratis vulnerability scanner te maken dat over het internet machines kon scannen. Rond deze software werd het bedrijf tenable opgericht en werd al snel 1 van de populairste oplossingen voor vulnerability managment. Vanaf nessus versie 3 (2005) werd de source code niet meer publiek gemaakt omdat er misbruik van gemaakt werd door de concurrentie. De nessus scanner werd doorverkocht door bedrijven met hun eigen logo en vermelden dat het hun scanner was, ook was er weinig tussenkomst van de community (vooral op gebied van de scanner engine) LeMay (2005).

#### Licenties

Tenable heeft een aantal nessus producten, maar voor deze bachelorproef kijken we naar 'Nessus Professional'. De reden hiervoor is dat dit product zo goed als volledig overeenkomt met een openvas installatie (met een paar limitaties). De licentiekosten hiervoor zijn 2190 $ / jaar (ongeveer 2010 euro) en is proprietary software.

#### Support

Als men een licentie van 'Nessus Professional' heeft, is het mogelijk om support te krijgen via live chat, email en een support portal. Deze zaken zijn 24/7 te benaderen „Nessus Support" (g.d.).

#### Nvt development

Nessus heeft op moment van schrijven 87.071 plugins „Nessus plugins" (g.d.). Deze NVTS worden geschreven in .nasl (Nessus attack script language).

Voor de shellshock vulnerability is er een NVT verschenen op 24-09-2014 „Nessus shellshock" (g.d.). Dit is dezelfde dag als shellshock publiek gemaakt werd. De windows vulnerability NVT werd geüpload op 23-10-2008, dit is ook op de dag dat windows de vulnerability publiek maakte.

#### Functionaliteiten

***Credential scan:*** Deze functionaliteit is beschikbaar, met ondersteuning voor een aantal services. De belangerijste services worden hier opgelijst: database, mongoDB, ssh, windows en SNMPv1/v2c „Nessus products" (g.d.).

***Meerdere scans:*** Dit is mogelijk op elke versie.

***Agent based:*** Dit is niet mogelijk met de 'professional' licentie, maar wel met de 'manager' licentie „Nessus products" (g.d.). ***Onderhoud:*** Op de webinterface is het mogelijk om bepaalde of alle elementen te updaten. Ook is het mogelijk om een interval in te stellen wanneer deze moet geüpdate worden (dagelijks, wekelijks of maandelijks).

#### Hardware

De minimum systeemvereisten voor een scanner zijn:

- Dual core 2GHz CPU
- 2-4 GB RAM
- 30GB HDD

„Nessus Hardware Requirements" (g.d.)

### 1.1.5 Openvas

#### Geschiedenis

Openvas is een fork van de laatste source code die nessus publiek heeft gemaakt (2005), en begon onder de naam „GNessUs - GPL based Nessus" (g.d.) en werd begonnen door Tim Brown. Dit project is nog steeds volledig opensource „Source Code Documentation" (2017) en gratis voor zowel personen, als bedrijven met uitzondering op een (niet verplichte) betalende licentie van greenbone voor NVT's „About NVT Feed" (g.d.). De grootste bijdrager voor openvas is greenbone, deze zorgt voor een groot deel van de NVTs, de webinterface en bugfixes „Contributors" (g.d.).

#### Licenties

Openvas valt onder de GPL license, is open-source en is gratis te gebruiken voor zowel personen, als bedrijven.

#### Support

Een nadeel van open-source projecten is vaak dat je afhankelijk bent van de community voor support. Dit is bij openvas niet anders, buiten een mailing list is de support zo goed als nietbestaande. De openvas website bevat veel verouderde links zoals een wiki, IRC chat en screenshots. Op het moment van schrijven is de laatste entry in de wiki voor openvas 7

„OpenVAS Documentation” (2014), ook op de screenshots pagina is de laatste entry voor openvas 7 „Screenshots” (g.d.).

### Nvt development

Openvas heeft op het moment van schrijven 53.082 plugins „About NVT Feed” (g.d.) waarvan er 214 geüpload zijn in de laatste maand. Wat hier ook opvallend is, is dat alle plugins in openvas geschreven worden in nasl (dit staat voor 'nessus attack script language'). Openvas gebruikt dus nog steeds dezelfde code voor NVTs te schrijven als nessus.

Voor de shellshock vulnerability heeft openvas de NVT geupload op 25-09-2014 „Openvas shellshock” (g.d.). Dit is 1 dag later als de announcement van shellshock en is gemaakt door greenbone. Dit lijkt onschuldig, maar shellshock heeft 24 uur na de ontdekking al voor het ontstaan van botnets gezorgd Greenberg (2014).

De windows vulnerability werd beschikbaar gesteld op 24-10-2008, Dit is 1 dag na de bekendmaking door windows. Hoewel de gevolgen niet zo erg waren als shellshock, gebruikte de worm „Conficker” (g.d.) deze vulnerability om zich te verspreiden.

### Functionaliteiten

***Credential scan:*** Deze functionaliteit is beschikbaar, maar dit is niet op scan niveau. Dit betekent dat men 2 verschillende targets moet aanmaken als men een credential en een non-credential wil starten (ookal is dit hetzelfde IP). De ondersteunde services zijn hier: SSH, SMB, ESXi en SNMP.

***Meerdere scans:*** Dit is mogelijk.

***Agent based:*** Dit is mogelijk.

***Onderhoud:*** Het is niet mogelijk om in de webinterface updates te starten. De updates (voor alles) worden wel automatisch elke dag geüpdate om 1 uur snachts door middel van cronjobs.

### Hardware

De minimum systeemvereisten voor openvas staan niet op de site, er is wel een virtuele machine aanwezig voor openvas te testen. De requirements hieronder zijn van deze virtuele machine, in realiteit zullen deze requirements hoger liggen. De reden hiervoor is dat dit een minimale versie is dat bepaalde features niet heeft, ook zijn er nog meer beperkingen zoals dat het besturingssysteem kan niet updaten. Dit is dus niet gepast voor een productie omgeving.

- Dual core CPU
- 2 RAM

- 9GB HDD

„Openvas Hardware Requirements” (g.d.)

## 1.2 Probleemstelling en Onderzoeksvragen

## 1.3 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 3, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

# 2. Methodologie

## 2.1 Testen bepalen

In dit onderzoek zal er met elke scanner 4 scans uitgevoerd worden op elk target. De reden hierachter is dat er in een professionele omgeving 2 punten zijn die belangerijk zijn.Zowel nessus als openvas hebben hiervoor aparte profielen die focussen op een ander deel. Naast deze 2 scans is een credential scan ook een noodzaak om een inzicht te krijgen hoe goed een systeem lokaal beveiligd is (indien men toegang en permissie heeft om deze te scannen met bv. ssh). Voor elke scanner zal er dus een snelle credential, accurate credential, snelle uncredential en een accurate uncredential scan uitgevoerd worden. Dit zal gebeuren in een paar omgevingen die we hieronder bespreken.

Eerst testen we dit op honeypots, dit zijn servers die intentioneel een zwakke configuratie hebben die veel vulnerabilities toont op de scanner. Het doel hiervan is dat deze heel goed opgevolgd moeten worden, en van zodra men activiteit op de server opmerkt dit onderzoeken en kijken of er onbevoegde persoon / malware aanwezig is. Deze test zal bepalen hoeveel vulnerabilities op een zwak systeem gevonden worden voor zowel een linux, als een windows machine.

Hierna zullen de scanners gebruikt worden om een webserver te scannen over het internet. Deze test is een goede weergave voor een deel van een penetration test. We verwachten op de server zelf weinig tot geen fouten, maar de extra informatie die de scanner ons geeft zoals het besturingssysteem of welke diensten er draaien op het systeem. Deze informatie kan gebruikt worden voor andere aanvallen zoals social engineering.

## 2.2 Opzetten testomgeving

Voor dit onderzoek werd er gebruik gemaakt van 4 virtuele machines op Virtualbox versie 5.1.10. Voor de virtuele machines waar er een vulnerability scanner op moest draaien, hebben we gebruik gemaakt van een CentOS minimal image. Deze kregen ook dezelfde instellingen die hieronder vermeld worden:

- 8000 MB ram
- 4 CPUs
- NAT & host-only interfaces

Voor de honeypots gebruikten we een iets minder zware virtuele machine, de instellingen hiervoor worden hieronder ook vermeld.

- 1000 MB ram
- 2 CPUs
- Host-only interface

De fysieke computer waar deze testen op zijn uitgevoerd heeft 16GB ram en een cpu met 4 cores en 8 threads. Tijdens de testen waren er geen andere programmas actief die een impact zouden kunnen hebben op de resultaten. Ook werden alleen de machines die getest moeten worden aangezet zodat beide scanners geen impact op elkaar konden hebben.

## 2.3 Installatie

### Nessus

Voor nessus te downloaden op de virtuele machine (met enkel CLI), moeten we naar de productpagina gaan. Sinds we geen extra software op de virtuele machine wouden (zoals FTP of SMB), zullen we wget gebruiken voor de .rpm file te downloaden. Dit is niet zo simpel, sinds de download link een pop up weergeeft voor de juiste versie te kiezen. Na dit probleem op te zoeken, is het duidelijk dat we bepaalde waarden met wget moeten meegeven ***COMMAND***.

Na de installatie van de RPM file (38 MB), poort 8834/tcp te openen in firewall-cmd en de service nessusd te starten is de webinterface bereikbaar. Hier moeten we de licentiecode ingeven die we verkregen hebben bij het aanvragen van een nessus home licentie. Hierna volgt een download scherm dat +- 7 minuten duurt. Hierna kunnen we beginnen met scannen.

### Openvas

Voor openvas te downloaden op de virtuele machine (met enkel CLI), hebben we 2 opties. We kunnen deze van source compilen, of de „OpenVAS Binary Packages (3rd party)" (g.d.). Wij gebruiken de packages, omdat dit minder tijd in beslag neemt. Voor deze

packages te downloaden gebruiken we de atomicorp repository, omdat hiernaar verwezen word op de openvas site.

We installeren de packages voor openvas 9 via yum. De totale grootte van alle packages is 70 MB (185 dependencies). Na deze te downloaden moeten we alles installeren via command line, hiervoor gebruiken we 'openvas-setup'. Deze download eerst alle NVTs (28 MB, verplicht), maar na de download crashed het script omdat we de package 'bzip2' niet geïnstalleerd hebben. We starten het script terug op en merken dat we alle NVTs opnieuw moeten downloaden. Hierna geeft het script een niet fatale error 'certool not found', maar het script gaat verder en blijft vasthangen na 'verify admin password'. Op de achtergrond is het script een databank aan het maken met alle NVTs in, na deze stap is het script gedaan. Het script heeft 23 minuten in beslag genomen.

Na poort 9392/tcp open te zetten in firewall-cmd is het *niet* mogelijk om te connecteren op de webinterface. We gebruiken een ander script 'openvas-check-setup –v9' om te kijken wat er mis is. Deze geeft weer dat de redis server niet gestart is. Deze zit in een failing state, en wil niet herstarten. Na de log files te raadplegen hebben we selinux op 'permissive' gezet, en werkt de redis server nu wel.

Na de check-config nog eens te overlopen, blijkt dat de services van openvas momenteel niet aanstaan. Bij deze opmerkingen staan er tussen haakjes openvassd en openvasmd. Na deze commandos uit te voeren, blijkt de services nog steeds niet te draaien. Na dit verder te onderzoeken, blijkt dat de service openvas-scanner, openvas-manager en gsad noemen. we starten deze op en runnen de check-config opnieuw.

De config geeft nogmaals een probleem, deze keer omdat de database geen tot weinig gegevens bevat. Hiervoor moeten we het commando 'openvasmd –rebuild' starten. Na 4-5 minuten is deze gedaan en draaien we de check-config nogmaals. Nu geeft deze weer dat er geen SCAP en CERT data aanwezig zijn. Hoewel dit *niet* verplicht is, installeren we deze gegevens voor de volledigheid van de scanner. De sync moet gebeuren met rsync, maar sinds deze sync veel files bevatten die niet groot zijn gaat dit heel traag. Deze duurden samen ongeveer 22 minuten en nemen +-800MB in beslag.

Als al deze gegevens aanwezig zijn, draaien we het commando check-config nog eens. Deze geeft weer dat selinux moet gedisabled worden. We proberen deze eerst op 'permissive' te zetten, maar het script geeft nog steeds een foutmelding. Na een reboot van de server voor selinux uit te zetten, geeft het script weer dat er een paar packages die **optioneel** zijn niet aanwezig zijn (zoals alien en net-tools). We installeren deze ook zodat deze geen invloed kunnen hebben op de eindresultaten van de scans.

Uiteindelijk geeft het script weer dat de openvas installatie klaar is voor gebruik. Na het bezoeken van de webinterface blijkt dat dit nog steeds niet mogelijk is. Na te troubleshooten blijkt dat de error in het begin van openvas-setup (certool not found) hier schuldig voor is. Na de package 'gnutls-utils' te installeren en nieuwe certificaten te genereren, is het mogelijk om de webinterface te raadplegen.

Als we de CLI bekijken, blijkt dat deze niet werkt omdate we niet werken met socket files. We lossen dit op door openvas op localhost te laten luisteren in plaats van een .sock file te

gebruiken (openvasmd -a 127.0.0.1), hierna werkt de CLI zoals het hoort.

#### Honeypots

Voor de linux honeypot hebben we 'metasploitable' genomen. Dit is een linux machine die gebaseerd is op ubuntu 8.04 en een reeks services heeft draaien. Deze services zijn opzettelijk zo zwak mogelijk geconfigureerd zodat men op deze virtuele machines vulnerability scanners kunnen testen en zo leren een penetration test uit te voeren.

Voor de windows honeypot hebben we een windows xp machine zonder service packs geïnstalleerd, maar met een aantal servers (SMB,SMTP,SNMP, FTP en IIS). Vooral door deze missende service packs hebben deze veel problemen + er zijn geen opzettelijk slecht geconfigureerde machines zoals metasploitable omdat windows xp onder een licentie valt.

#### Webservers

Als 2de test zullen we een webserver scannen die momenteel in gebruik is door HoGent. ***URLS***. Hiervoor hebben we een schriftelijke toestemming gekregen door Van Vreckem Bert.

## 2.4 Instellingen scans

## 2.5 Scan resultaten

## 2.6 Scan resultaten valideren

## 2.7 Analyseren van de data

# 3. Conclusie

# Bibliografie


Asadoorian, P. (2012). The Right Way To Configure Nessus For Comparison. Verkregen van http://securityweekly.com/2012/08/24/the-right-way-to-configure-nes/

Greenberg, A. (2014). Hackers Are Already Using the Shellshock Bug to Launch Botnet Attacks. Verkregen van https://www.wired.com/2014/09/hackers-already-using-shellshock-bug-create-botnets-ddos-attacks/

Hoffman, C. (2013). Hacker Hat Colors Explained: Black Hats, White Hats, and Gray Hats. Verkregen van https://www.howtogeek.com/157460/hacker-hat-colors-explained-black-hats-white-hats-and-gray-hats/

LeMay, R. (2005). Nessus security tool closes its source. Verkregen van https://www.cnet.com/news/nessus-security-tool-closes-its-source/

About CVE. (g.d.). Verkregen van https://cve.mitre.org/about/

About NVT Feed. (g.d.). Verkregen van http://www.openvas.org/openvas-nvt-feed.html

About OpenVAS. (g.d.). Verkregen van http://www.openvas.org/about.html

Conficker. (g.d.). Verkregen van https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Win32%2fConficker

Contributors. (g.d.). Verkregen van http://www.openvas.org/contributors.html

Cross-site Scripting (XSS). (g.d.). Verkregen van https://www.acunetix.com/websitesecurity/cross-site-scripting/

GNessUs - GPL based Nessus. (g.d.). Verkregen van http://www.securiteam.com/tools/6V00B0AEAW.html

Linkedin. (g.d.). Verkregen van https://security.linkedin.com/blog-archive#11232015

Nessus Hardware Requirements. (g.d.). Verkregen van https://docs.tenable.com/nessus/Content/HardwareRequirements.htm

Nessus plugins. (g.d.). Verkregen van http://www.tenable.com/plugins/index.php?view=all

Nessus products. (g.d.). Verkregen van http://www.tenable.com/products/nessus-vulnerability-scanner

Nessus shellshock. (g.d.). Verkregen van https://vulners.com/nessus/BASH_REMOTE_CODE_EXECUTION.NASL

Nessus Support. (g.d.). Verkregen van http://www.tenable.com/products/nessus/nessus-faq#Nessus_Support

NVD CVSS Support. (g.d.). Verkregen van https://nvd.nist.gov/vuln-metrics/cvss

OpenVAS Binary Packages (3rd party). (g.d.). Verkregen van http://www.openvas.org/install-packages.html

Openvas Hardware Requirements. (g.d.). Verkregen van http://www.openvas.org/vm.html

Openvas shellshock. (g.d.). Verkregen van https://vulners.com/openvas/OPENVAS:804489

Screenshots. (g.d.). Verkregen van http://www.openvas.org/screenshots.html

Vulnerability Scanning. (g.d.). Verkregen van https://www.techopedia.com/definition/4160/vulnerability-scanning

OpenVAS vs Tenable Nessus. (2011). Verkregen van http://rageweb.info/2011/04/13/openvas-vs-tenable-nessus/

Nessus, OpenVAS and Nexpose VS Metasploitable. (2012). Verkregen van https://hackertarget.com/nessus-openvas-nexpose-vs-metasploitable/

What is Vulnerability Management Anyway? (2013). Verkregen van https://www.tripwire.com/state-of-security/vulnerability-management/what-is-vulnerability-management-anyway/

OpenVAS Documentation. (2014). Verkregen van https://wiki.openvas.org/index.php/Main_Page

How does vulnerability scanning work? (2015). Verkregen van https://community.qualys.com/docs/DOC-1068

Top 125 Network Security Tools. (2015). Verkregen van http://sectools.org/tag/vuln-scanners/

Vulnerability Scanners. (2016). Verkregen van https://www.concise-courses.com/hacking-tools/vulnerability-scanners/

Nessus Professional. (2017). Verkregen van https://store.tenable.com/index.php?main_page=product_info&cPath=1&products_id=94

Source Code Documentation. (2017). Verkregen van http://www.openvas.org/src-doc.html

Peterson, D. (2010). credentialed-scanning. Verkregen van http://www.digitalbond.com/blog/2010/01/25/3-reasons-you-should-be-using-credentialed-scanning/

Rouse, M. (2005). vulnerability. Verkregen van http://whatis.techtarget.com/definition/vulnerability

Rouse, M. (2014). exploit. Verkregen van http://searchsecurity.techtarget.com/definition/exploit

# Lijst van figuren

# Lijst van tabellen