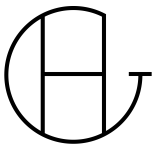

HoGent

Faculteit Bedrijf en Organisatie

Security en Managebility van Docker containers

Tomas Vercautter

Scriptie voorgedragen tot het bekomen van de graad van
Bachelor in de toegepaste informatica

Promotor:
Bert Van Vreckem
Co-promotor:
Bert Van Vreckem

Instelling: —

Academiejaar: 2015-2016

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Security en Managebility van Docker containers

Tomas Vercautter

Scriptie voorgedragen tot het bekomen van de graad van
Bachelor in de toegepaste informatica

Promotor:
Bert Van Vreckem
Co-promotor:
Bert Van Vreckem

Instelling: —

Academiejaar: 2015-2016

Tweede examenperiode

Samenvatting

Voorwoord

Inhoudsopgave

Hoofdstuk 1

Inleiding

Docker containers is een manier van virtualiseren van services. Hierbij worden services in een

In deze bachelorproef zal ik onderzoek doen naar de security en managebility van Docker conainers en deze vergelijken met de security en managebility van het draaien van de services rechtsreeks op de virtuele machine.

Deze bachelorprief is onderverdeeld in drie grote delen. In de eerste plaats bespreek ik de security zelf. In een tweede deel zal ik het hebben over de managebility van Docker containers. En als laatste onderdeel bespreek ik logging bij docker containers.

1.1 Probleemstelling en Onderzoeksvragen

Deel I

Corpus

Hoofdstuk 2

Security

Op vlak van security ben ik gestart van CIS Docker 1.6 Benchmark geschreven door **?**. Hierin is in samenwerking met de "Center for Internet Securityönderzoek gedaan naar de beveiliging van docker containers. Dit onderzoek is onderverdeeld in zes grote onderdelen namelijk Host configuration, Docker daemon configuration, docker daemon configuration files, Container images and build file, Container runtime en docker security operations. Hierbij wordt er gekeken welke van deze elementen standaard in docker worden uitgevoerd. In dit hoofdstuk zal ik dezelfde zes onderdelen behandelen en vergelijken. Hierbij zal ik kijken hoe we deze beveiliging aanpakken wanneer we onze services in een Docker container draaien tegenover wanneer we deze native in onze virtuele machine draaien.

Hierbij heb ik ook telkens deze methodes voor Docker containers uitgetest om te kijken hoe makkelijk deze kunnen geïmplementeerd worden in een nieuwe omgevinge of bestaande omgeving. Een grote hulp hiervoor was de github repository "docker-bench-security"van docker een grote hulp. Dit is een script die alle elementen die worden aangehaald in CIS Docker 1.6 Benchmark automatisch getest en feedback over gegeven.

2.1 Host Configuration

Een van de belangrijkste onderdelen in het beveiligen van de Docker stack is het beveiligen van de host machine waarop de docker containers zich zullen bevinden. Dit heeft drie onderdelen namelijk het up to date houden van het systeem, het juist

configureren van het systeem, en het auditten van het systeem.

2.1.1 Up to date houden van het systeem

Om de veiligheid van een systeem te kunnen garanderen moet er ten allen tijde gezorgd worden dat het systeem up to date blijft. Dit zorgt ervoor dat security flaws die gekend en gepatcht zijn ook op het systeem beveiligd zijn. Doordat we Docker gebruiken komt hier nog een extra laag bij. Bij het Updaten van een machine waarop de services rechtstreeks draaien moet enkel het host systeem, de kernel van het host systeem en de services up to date blijven. Doordat Docker een kernel deelt met het hostsysteem wordt het up to date houden van de kernel belangrijker. Naast het onderhouden van de voorvernoemde elementen moet er nu, doordat we Docker gebruiken, nog met twee extra dingen rekening gehouden worden. Enerzijds moet Docker zelf nu ook upgedate worden, maar een onderdeel die soms vergeten wordt is dat elke container nu ook zijn eigen besturingssysteem gebruikt. Hierover wordt er verder bij het beveiligen van Container images meer uitleg over gegeven.

2.1.2 Configureren van het systeem

Bij het configureren van het hostysteem wordt net zoals bij het up to date houden van het systeem dezelfde stappen van een native applicatie op een hostysteem configureren gevolgd. Maar door de extra laag komen daar nog extra stappen bij. Bij een systeem waar Docker op draait wordt er aangeraden om een aparte partitie te creëren voor de containers. Dit is een simpele operatie waardoor alle docker gerelateerde files zich niet meer tussen de files van het hostsysteem bevinden.

Daarnaast moet ook onder controle gehouden worden wie toegang krijgt tot de docker daemon. De Docker daemon heeft 'root' access, gebruikers die worden toegevoegd aan de 'docker' usergroup en geen 'root' privileges hebben kunnen hierdoor 'root' access verkrijgen tot het hostsysteem. Men directories sharen tussen het host systeem en een guest container. Doordat containers standaard altijd runnen als 'root' heeft de container als de '/' directory gemount is op deze container ongelimiteerde toegang tot het volledige host systeem. Hierdoor kunnen zonder ristricties aanpassingen aan het hostysteem gedaan worden. Dit betekend dat een gebruiker die toegang heeft tot de docker daemon verhoogde rechten kan verkrijgen door gewoon een container op te starten.

Zoals bij een virtuele machine waarop de services rechtstreeks wordt ook bij een

systeem waar docker containers op draaien aangeraden overbodige services uit te schakelen of te verwijderen. Moest een gebruiker door een van deze services toegang krijgen tot het hostsysteem. Zou hij makkelijker 'root' rechten kunnen vergrijgen volgens de manier beschreven in de vorige paragraaf.

2.1.3 Auditen van het systeem

2.2 Docker daemon configuration

Bij het beveiligen van de docker stack is ook de configuratie van de docker daemon een belangrijk element. Dit heeft twee grote onderdelen. Enerzijds de configuratie van de docker daemon zelf, dit houd in dat we de daemon zelf zo gaan configureren dat deze zo veilig en robuust mogelijk wordt. Aan de andere kant moeten we ook zorgen dat de configuratie files zelf de juiste file permissies en owners hebben. Hierbij is de vergelijking met een virtuele machine, waar de services rechtstreeks op draaien, moeilijk aangezien er hier geen docker daemon op te vinden is. Omdat het configureren van de Docker daemon dingen verandert doe voor elke container toegepast worden kunnen we het wel vergelijken met hoe de inviduele services globaal geconfigureerd worden.

2.2.1 Globale configuratie

2.2.2 netwerk configuratie

2.2.3 registry configuraion

2.2.4 Permissies voor Docker config files

Het configureren van de Docker daemon met de juiste instellingen voor zowel networking, registry gebruik en algemene configuratie heeft enkel nut als we ook de configuratie files waarin deze instellingen worden bewaard correct beveiligd zijn.

2.3 Security of Docker images

2.4 Container configuration

Hoofdstuk 3

Managebility

Hoofdstuk 4

Logging

Logging is een zeer belangrijk onderdeel van docker containers. Omdat we onze docker containers veel meer gaan isoleren en onze containers veel vluchtiger zijn dan gewone services op een virtuele host wordt loggen van

Hoofdstuk 5

Conclusie

Lijst van figuren

Lijst van tabellen