Security problemen in container orcherstratie tools indentificeren en vermijden

Onderzoeksvoorstel Bachelorproef 2020-2021

Nick Heymans[1]

Samenvatting
In deze bachelorproef worden de moderne veiligheidsrisico's geanalyseerd die gepaard gaan met container virtualisatie en container orchestratie. Hierbij wordt er naar manieren gezocht om deze op te lossen of te vermijden. De laatste jaren is het gebruik van containers in de IT infrastructuur fors gestegen en dit zal zo blijven in de komende jaren. De kracht van container virtualisatie en container orchestratie is dat het administrators en ontwikkelaars ondersteund. Waarbij ze de mogelijkheid krijgen om hun applicaties op een draagbare en snelle manier uit te rollen en deze te beheren op grote schaal. Maar zoals bij alle technologieën zijn hier enkele veiligheidsrisico's aan verbonden. Het gebruik en de daaraan verbonden risico's van container virtualisatie en container orchestratie zullen in dit onderzoek onderzocht worden. In het tweede deel, waar er verder gebouwd wordt op de bevindingen uit het eerste deel, wordt er onderzocht hoe deze risico's vermeden of opgelost kunnen worden. Als laatste zal er een 'proof-of-concept' opstelling gemaakt worden. Hierin zal er gekeken worden naar de effecten van de mogelijke oplossingen op het uitrollen van een container cluster. Er zal gefocust worden op de snelheid en resource management.

Sleutelwoorden
Onderzoeksdomein. Security — Containerization — Container Orchestration

Co-promotor
?????????[2] (Bedrijfsnaam)

Contact: [1] nick.heymans@student.hogent.be; [2] piet.pieters@acme.be;

Inhoudsopgave

1. Introductie

Het uitrollen en schalen van applicaties wordt steeds vaker gedaan met behulp van containers. Tijdens de ontwikkeling van traditionele applicaties word de applicatie ontwikkeld in een specifiek testomgeving. Vervolgens wordt de applicatie overgezet naar de productieomgeving wat vaak voor problemen zorgt (bijvoorbeeld van een linux testomgeving naar een Windows productieomgeving). Een container is een pakket waar één enkele applicatie in zit, samen met alle nodige afhankelijkheden. Dit zorgt ervoor dat deze gemakkelijk en snel van de ene omgeving naar de andere kan overgezet worden. De containers maken gebruik van een 'runtime engine', dit is een laag die verantwoordelijk is voor de communicatie tussen het operating system van de host machine en de containers zelf. De meeste gebruikte 'runtime engine' is de 'Docker Engine'[1]. Deze is al sinds 2013 de industriestandaard als het gaat over container software. (McCarty, 2018) Naar mate het gebruik van containers steeg, steeg ook de nood naar op manier om deze vanuit één centrale locatie te beheren. Om aan deze vraag te voldoen werden container orchestratie tools, zoals Kubernetes [2], ontwikkeld. Deze tools helpen bij het opzetten, uitbreiden en verbinden van een grote hoeveelheid containers.

Container applicaties hebben enkele voordelen tegenover normale applicaties, ze draaien namelijk geisoleerd van de rest van het systeem. Ze kunnen dus perfect werken zonder afhankelijk te zijn van andere containers. Dit garandeerd dat als er één container gecompromitteerd is, de rest zonder interuptie kan verderwerken. De containers delen wel verschillende resources van het host systeem, wat de deur opent voor veiligheids inbreuken tussen containers.

Gartner (Pettey, 2019) voorspeld dat tegen 2022 maar liefst 75% van alle internationale organisaties gecontaineriseerde applicaties zullen gebruiken in hun productieomgeving. Dit zowel in lokale datacenters alsook in online cloud omgevingen. Uit een raport van Tripwire (2019) blijkt dat 94% van bevraagden bezorgd zijn over de veiligheid van hun containers. Uit hetzelfde raport blijkt ook dat 47% weet dat ze kwetsbare containers gebruiken in hun productieomgeving.

[1] https://docs.docker.com/engine/
[2] https://kubernetes.io/

In deze paper zal ik een onderzoek uitvoeren waar ik op zoek ga naar de belangrijkste bronnen van security inbreuken en hoe deze vermeden kunnen worden. Tegelijk krijg ik via dit onderzoek de kans om te testen of er vooral technische fouten zijn met de tools of er toch vooral menselijke fouten aan basis staan van de veiligheidsrisico's. In de volgende paragraaf staat er beschreven hoe ik te werk zal gaan.

2. Methodologie

Voor dit onderzoek zullen er drie scenario's opgezet worden. Bij elke scenario zullen er verschillende 'security best practices en 'tools' gebruikt worden. Deze zullen getest worden op basis van de volgende criteria:

- Deployment snelheid
- Benodigde resources
- Stabiliteit

Voorbeelden van scenario's:

- S(0): Er wordt een container applicatie opgezet in een Kubernetes cluster zonder extra security configuratie.
- S(1): Er wordt een container applicatie opgezet in een Kubernetes cluster en enkele 'best practices' worden toegepast.
- S(2): Er wordt een container applicatie opgezet in een Kubernetes cluster waarin de grootste security risico's worden vermeden.

Door gebruik te maken van deze scenario's en criteria hopen we vast te stellen dat het toepassen van security 'best practices' een positieve invloed heeft op het gebruik van containers en orchestratie tools.

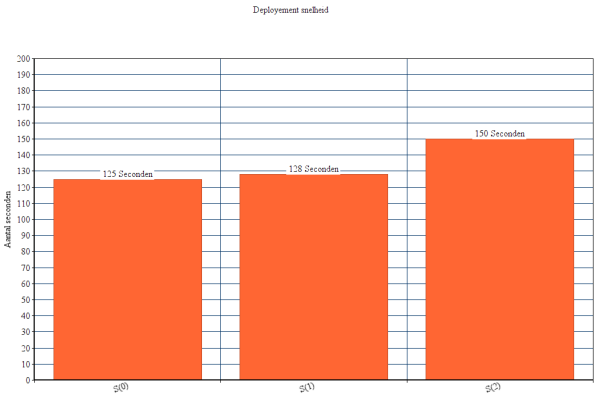
3. Verwachte resultaten

Op basis van de criteria wordt er verwacht dat Scenario 0 en 1 even snel op opgezet kunnen worden. Ze zullen beide kwetsbaarder zijn aangezien de 'best practices' vooral bestaan uit het correct gebruik van wachtwoorden en gebruiker privileges. Scenario 2 daarintegen zal iets meer tijd nodig hebben om opgezet te worden(zie Figuur1) en zal daarbij meer resources gebruiken(zie Figuur2). Dit zou te wijten zijn aan het delen van resources tussen de containers wat een security risico inhoudt en dus tot een minimum zal moeten beperkt worden(Education, 2019).

4. Verwachte conclusies

Uit dit onderzoek willen we concluderen dat het toepassen van 'best practices' en het correct gebruik van security tools een positief effect teweeg brengt bij het gebruik van container orcherstratie tools. We trachten daarnaast ook aan te duiden dat het omzeilen van security risico's een belangrijk aspect is bij het ontwikkelen van container applicaties. Tot slot kan er geconcludeerd worden dat het beveiligen van container clusters steeds belangrijker wordt. Daarnaast is het tevens van belang dat de persoon die een cluster opzet daarbij de onderliggende werkwijze goed kent en zich bewust is van de mogelijke valkuilen.

5. Bijlagen



Figuur 1. Verwacht eindresultaat

Benodigde resources	CPU %	Memory %
S(0)	45%	30%
S(1)	46%	32%
S(2)	55%	45%

Figuur 2. Verwacht eindresultaat

Referenties

Education, I. C. (2019, mei 25). *Containerization*. IBM. https://www.ibm.com/cloud/learn/containerization#toc-what-is-co-r25Smlqq

McCarty, S. (2018, februari 22). *A Practical Introduction to Container Terminology*. https://developers.redhat.com/blog/2018/02/22/container-terminology-practical-introduction/#h.6yt1ex5wfo3l

Pettey, C. (2019, april 23). *6 Best Practices for Creating a Container Platform Strategy*. https://www.gartner.com/smarterwithgartner/6-best-practices-for-creating-a-container-platform-strategy/

Tripwire. (2019, januari 1). *Tripwire State of Container Security Report* (onderzoeksrap.). Tripwire. Verkregen 30 november 2020, van https://3b6xlt3iddqmuq5vy2w0s5d3-wpengine.netdna-ssl.com/state-of-security/wp-content/uploads/sites/3/Tripwire-Dimensional-Research-State-of-Container-Security-Report.pdf