Onderzoeksvoorstel

Dynamisch Docker-omgevingbeheer voor CTF-events op on-prem infrastructuur.

Bachelorproef, 2025-2026

Manu Devloo

E-mail: manu.devloo@student.hogent.be
Project repo: https://github.com/Manu-Devloo/Bachelorproef
Co-promotor: Laurens Singier (HoWest, laurens.singier@howest.be)

Samenvatting

Dit onderzoeksvoorstel behandelt het dynamisch beheer van Docker-containers voor Capture The Flag (CTF) events op on-premise infrastructuur. Hierdoor focust de probleemstelling op het aanmaken en beëindigen van containers op basis van gebruikersacties. De hoofdonderzoeksvraag onderzoekt hoe dit proces geautomatiseerd kan worden met behulp van CTFd en Docker. Hierbij wordt specifiek gewerkt binnen de constraints van een self-hosted omgeving, zonder terugkerende licentiekosten en met een focus op open-source dependencies. Door middel van een literatuurstudie en een proof-of-concept wordt gekeken naar de beveiligingsaspecten, resource-limieten en communicatie tussen CTFd en Docker. Het doel is een schaalbare oplossing te bieden die misbruik voorkomt en performance garandeert.

Keuzerichting: Development (full stack)
Sleutelwoorden: Docker, CTFd, Infrastructure, Security, Automation

Inhoudsopgave

1. Introductie

Dit onderzoeksvoorstel richt zich op het dynamisch beheer van Docker-omgevingen tijdens Capture The Flag (CTF) events (Innovirtuoso, z.d.) op on-premise infrastructuur. Het idee voor dit onderzoek komt voort uit de noodzaak om efficiënt en veilig challenges te beheren (Singier, 2025).

1.1. Probleemstelling

De centrale probleemstelling luidt: Bestaat er een manier om dynamisch Docker-omgevingbeheer te doen op on-prem infrastructuur tijdens Capture The Flag events? Het beheren van containers voor challenges vereist een robuust systeem dat kan omgaan met gebruikersacties en variabele belastingen, zonder in te boeten op veiligheid of performance.

Daarnaast is het een vereiste dat de oplossing volledig self-hosted kan worden ingezet. Hiervoor zijn drie belangrijke redenen: flexibiliteit (mogelijkheid tot uitbreidingen en features), kostprijs (het betalen van on-premise infrastructuur versus cloud-diensten is voor veel organisaties een andere financiële flow) en de voorkeur voor open-source oplossingen.

1.2. Onderzoeksvraag

De hoofdonderzoeksvraag van dit bachelorproefvoorstel is:

> Hoe kunnen we dynamisch onze docker containers gelinkt aan CTFd-challenges beheren, aanmaken en beëindigen op basis van gebruikersacties en ingestelde variabelen?

Om deze hoofdvraag te beantwoorden, worden de volgende deelvragen geformuleerd, opgesplitst in probleem- en oplossingsdomein:

Probleemdomein:

- Welke fundamentele en direct toepasbare beveiligingsmaatregelen ('low-hanging fruit') zijn noodzakelijk om de veiligheid van de Docker-host en -containers te waarborgen?
- Welke specifieke beveiligingsrisico's brengt het dynamisch aanmaken van containers door gebruikers met zich mee op on-premise infrastructuur?
- Wat is de impact van frequente container-creatie en -vernietiging op de performance van de host-server?
- Welke communicatieprotocollen zijn geschikt voor de interactie tussen het CTF-platform en de container-runtime?

Oplossingsdomein (gericht op de Proof-of-Concept):

- Hoe kan een CTFd-plugin ontwikkeld worden die rechtstreeks communiceert met de Docker Engine API zonder zware externe dependencies?
- Op welke manier kunnen resource-limieten (CPU, RAM) en netwerkisolatie technisch worden afgedwongen in de PoC?
- Hoe kan de architectuur zo worden opgezet dat deze volledig self-hosted en modulair uitbreidbaar is?
- Hoe valideren we de stabiliteit van de oplossing onder gesimuleerde zware belasting (load testing)?

1.3. Onderzoeksdoelstelling

Het doel is om een werkende oplossing of proof-of-concept te realiseren die aantoont hoe CTFd-challenges dynamisch en veilig beheerd kunnen worden op eigen infrastructuur.

2. State-of-the-art

Er is al documentatie beschikbaar over het deployen van challenges binnen CTFd (CTFd, z.d.). Echter, het veilig en dynamisch beheren van deze containers op eigen infrastructuur brengt specifieke uitdagingen met zich mee.

Beveiliging is een cruciaal aspect; containers moeten geïsoleerd zijn om te voorkomen dat deelnemers dingen kunnen doen op de host of andere containers. Bronnen zoals de Docker Security Cheat Sheet van OWASP (OWASP, 2024) en artikelen over container security (Aikido, 2025) bieden richtlijnen hiervoor.

Daarnaast biedt een overzicht van het Docker-ecosysteem inzicht in de beschikbare tools en technologieën die ingezet kunnen worden (Arxiv, 2024). Het onderzoek zal zich baseren op deze bronnen om een veilige en schaalbare architectuur te ontwerpen.

3. Methodologie

Het onderzoek zal bestaan uit een combinatie van literatuurstudie en praktijkgericht onderzoek.

In de eerste fase wordt via literatuurstudie onderzocht welke 'low-hanging fruit' beveiligingsmaatregelen essentieel zijn. De focus ligt hierbij op praktische en effectieve best practices voor container security (zoals resource limits, netwerkisolatie en privilege management) in plaats van diepgaande theoretische beveiligingsmodellen. Daarnaast wordt gekeken naar de Docker Engine API en SDKs voor de integratie.

Vervolgens zal een proof-of-concept (PoC) worden ontwikkeld waarin een CTFd-omgeving wordt opgezet die communiceert met een Docker-daemon. Hierbij wordt gefocust op de implementatie van een API-interface, het instellen van resource limieten en het toepassen van de geïdentificeerde basisbeveiliging.

Tot slot zal de oplossing gevalideerd worden aan de hand van een uitgebreid testplan als onderdeel van de PoC-verificatie. Dit testplan beschrijft de strategie om zowel de stabiliteit als de veiligheid van het systeem te waarborgen. Concreet worden er load tests uitgevoerd om de schaalbaarheid en performance onder zware belasting te meten. Daarnaast worden specifieke beveiligingstests (zoals container escape pogingen en resource exhaustion simulaties) uitgevoerd om te verifiëren of de geïmplementeerde isolatie en restricties effectief zijn tegen misbruik.

4. Verwacht resultaat, conclusie

Het verwachte resultaat is een gedocumenteerde en werkende implementatie voor dynamisch containerbeheer binnen CTFd. Dit omvat:

- Een adviesrapport over de te gebruiken architectuur en beveiligingsmaatregelen.
- Een proof-of-concept implementatie.
- Testresultaten die de schaalbaarheid en stabiliteit aantonen.
- Installatie- en beheerinstructies om de reproduceerbaarheid van de PoC te waarborgen.
- Technische documentatie van de ontwikkelde architectuur, zodat derden hierop kunnen verder bouwen.

De meerwaarde van dit onderzoek ligt in het vereenvoudigen van het beheer van CTF-events op eigen hardware, met behoud van veiligheid en performance.

Referenties

Aikido. (2025, april 28). *Docker & Kubernetes Container Security Explained*. Verkregen november 28, 2025, van https://www.aikido.dev/blog/docker-kubernetes-container-security

Arxiv. (2024). Navigating the Docker Ecosystem: A Comprehensive Taxonomy and Survey. https://arxiv.org/pdf/2403.17940.pdf

CTFd. (z.d.). *Deploying Challenge Services*. Verkregen november 28, 2025, van https://docs.ctfd.io/tutorials/challenges/deploying-challenges

Innovirtuoso. (z.d.). *Capture The Flag (CTF) Explained: How Hacking Competitions Turn You Into a Cyber Pro*. Verkregen december 11, 2025, van https://innovirtuoso.com/cybersecurity/capture-the-flag-ctf-explained-how-hacking-competitions-turn-you-into-a-cyber-pro/

OWASP. (2024). *Docker Security Cheat Sheet*. Verkregen november 28, 2025, van https://cheatsheetseries.owasp.org/cheatsheets/Docker_Security_Cheat_Sheet.html

Singier, L. (2025). Ideeendocument BP.