KERSTIN SCHMID (FAU),

FELIX FREILING (FAU),

KONSTANTIN BAYREUTHER (DHBW MANNHEIM)

IMF 2025, 17.09.2025

LIMITS TO THE FORENSIC ANALYSIS OF CONTAINER APPLICATIONS IN CLOUD ENVIRONMENTS

MOTIVATION

- Software applications in form of container is an popular deployment method
- Orchestration layers such as Kubernetes is utilized for automated and efficient management
- Cloud service providers (CSP) offer container solutions in different cloud models
- Impact to DFIR: Containers are ephemeral; IR-teams has to act within short period of time

RESEARCH QUESTIONS / GOALS

- Investigate the relation between container access level (deployment model) and the ability to gather sufficient evidence in case of an incident
  - Infrastructure as a Service (IaaS): high amount of significant artifacts
  - Platform as a Service (PaaS) / Software as a Service (Saas): decreasing amount of artifacts
- Prove a tradeoff between these access level and provability
- Discuss the implications to the DFIR process

RELATED WORK

- Cloud forensics with the main focus of host forensics Grobauer and Schreck [2010], Ruan et al. [2011], and Farina et al. [2015]
- Recoverability of data from docker process memory Clausing [2016] and and Gharaibeh et al. [2024]
- All mentioned works either assume non-cloud environments or full system access
- Limited amount of scientific literature that discusses about limits of different cloud access models to the ability to collect forensic evidence

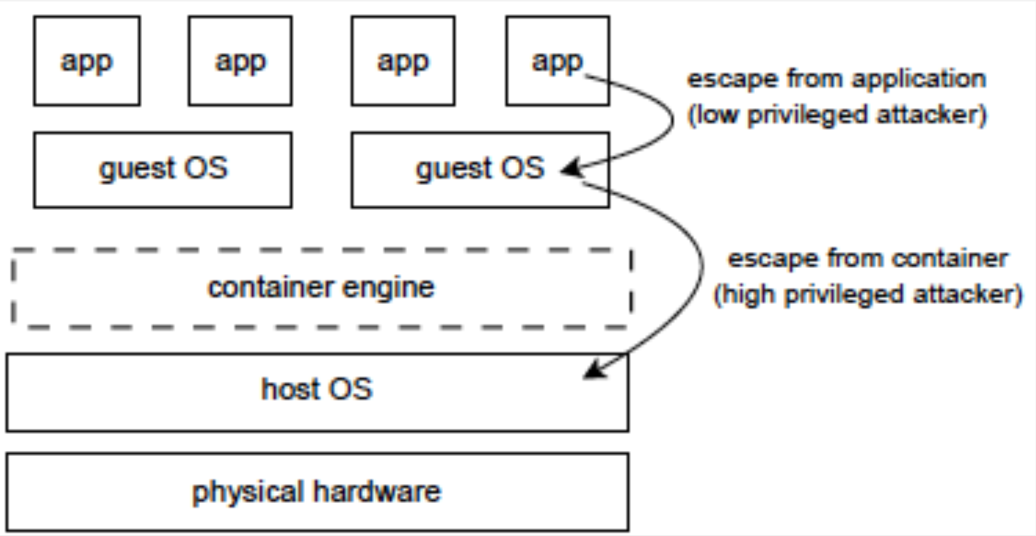
METHODOLOGY:
EXPERIMENTAL SETUP & ACQUISITION METHODS

Acquisition method	Forensic artifacts	IaaS: EKS[1]	PaaS: EKS Fargate[2]	SaaS: ECS[3]
Live analysis host	Memory dump, docker metadata, logs	✓	✗	✗
Snapshot container host	Container file system	✓	✗	✗
Analysis Kubernetes cluster via kubectl	Metadata, logs	✓	✓	✗
Live analysis container	Application content, files, runtime information	✓	✓	✓

[1] Amazon Elastic Kubernetes Service
[2] Amazon Elastic Kubernetes Service Fargate
[3] Amazon Elastic Container Service

METHODOLOGY: ATTACK SCENARIOS

- Low-privileged attacker:
  - Exploitation of a vulnerable web application
  - Deployment of attacker owned container
- High-privileged attacker:
  - Linux privilege escalation attack
  - Escape to host



RESULTS: LOW-PRIVILEGED ATTACKER

Attack vectors	**IaaS: EKS**	**PaaS: EKS Fargate**	**SaaS: ECS**
Initial Access: Exploit Public-Facing Application	✓	✓	✗
Execution: Command and Script Interpreter	✓	✗	✗
Credential Access: Unsecure Credential	✓	✗	n/a
Lateral Movement: Use Alternate Authentication Material	✓	✗	n/a
Execution: Deploy Container	✓	✓	n/a
Impact: Resource Hijacking	✓	✓	n/a

RESULTS: HIGH-PRIVILEGED ATTACKER

Attack vectors	**IaaS: EKS**	**PaaS: EKS Fargate**	**SaaS: ECS**
Initial Access: Compromise Software Supply Chain	✓	✗	n/a
Privilege Escalation: Exploitation for Privilege Escalation	✓	✓	n/a
Privilege Escalation: Deploy (Privileged) Container	✓	n/a	n/a
Lateral Movement: Escape to Host	✓	n/a	n/a

DISCUSSION & CONCLUSION

Attack vectors	IaaS: EKS	PaaS: EKS Fargate	SaaS: ECS
Low-privileged container	6/6	3/6	0/2
High-privileged container	4/4	1/2	0/0

- Amount of collectible evidences highly depends on the level of access
- Impact to the DFIR process
  - Container solution is better secured by default the more operational responsibility is transferred to the CSP
  - EKS solution: more significant artifacts; higher probability of finding answers to questions like „What happened?“

LIMITATIONS & FUTURE WORK

- Limitations:
  - Extraction time of artifacts immediatley after scenario was executed
  - Only one container application was executed simultaenously
- Future work:
  - CSPs must be required to provide interfaces and appropriate logging for an effective DFIR process
  - Investigation of further container technologies and additional access options to the containers

QUESTIONS?

Thanks for your attention!