Scalable and Lightweight CTF Infrastructures Using Application Containers

Arvind S Raj, Bithin Alangot, Seshagiri Prabhu and Krishnashree Achuthan

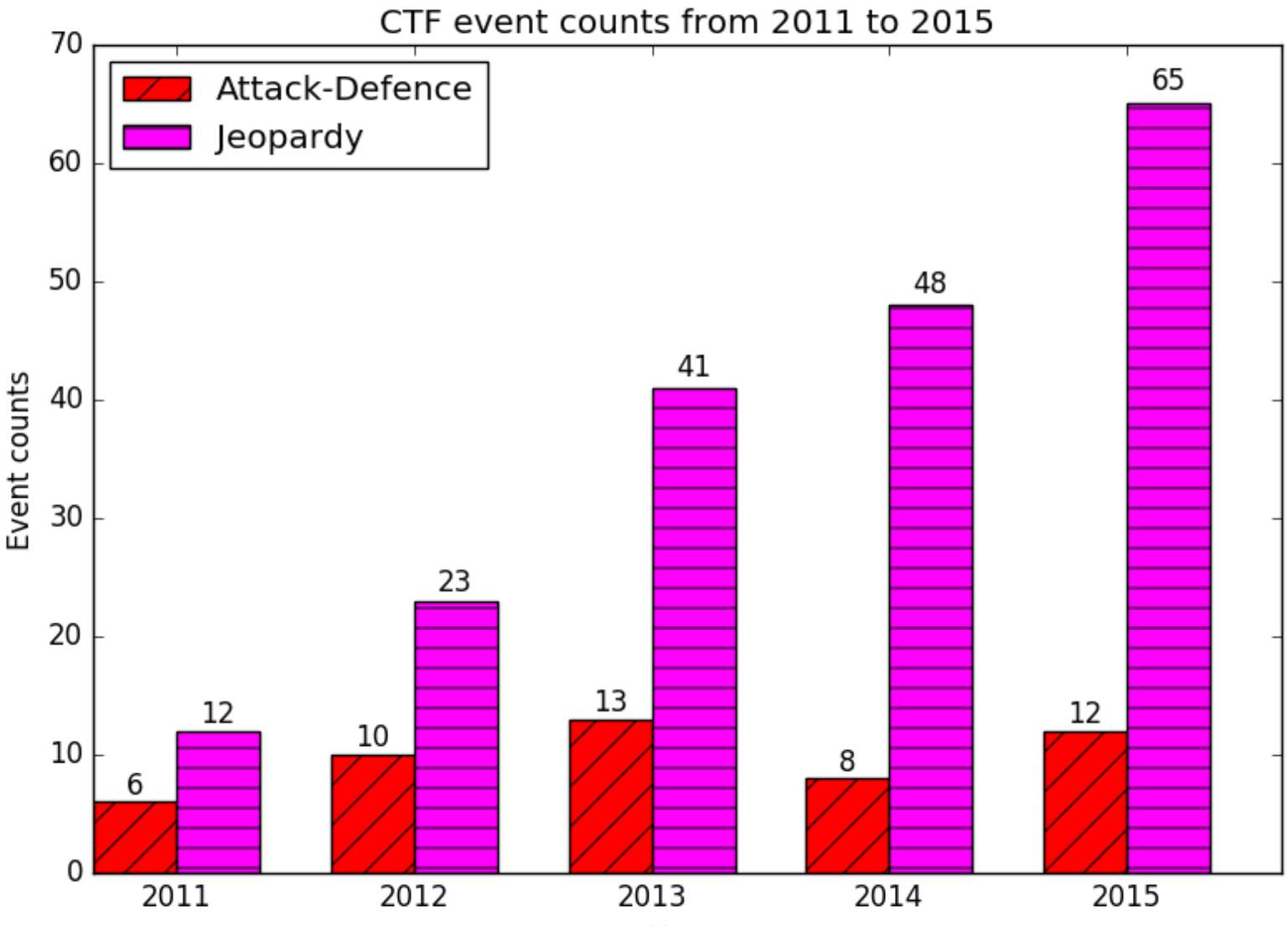
Amrita Center for Cybersecurity Systems and Networks
Amrita Vishwa Vidyapeetham, Kerala, India

2016 USENIX Advances in Security Education Workshop

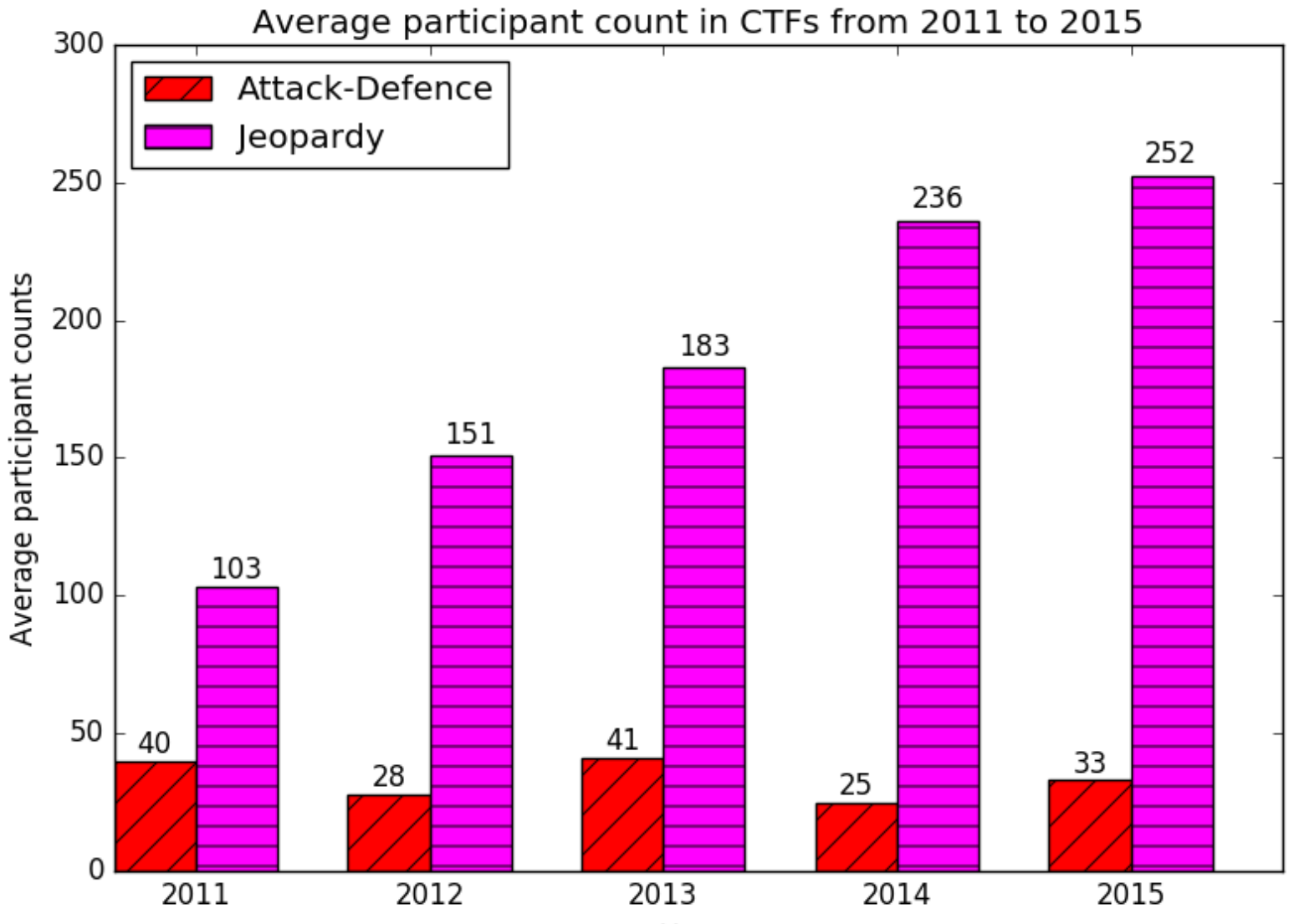
Introduction

- CTFs - an effective means to teach secure coding and computer security.
- Two popular formats: Jeopardy and Attack-defence.
- Jeopardy: Self-paced, offence only, non-interactive and more popular.
- Attack-defence: Real-time, offence and defence, interactive but less popular.

CTF event counts



Participation trends



Format challenges

- Both organizers and participants face challenges.
- Organizers: Complex infrastructure engineering and high resource requirements.
- Participants: Complex gameplay, infrastructure setup and IT policies.

Problem

Can we build less resource intensive and easily scalable contest infrastructures?

Solution

Replace virtual machines with application containers.

- Significant reduction in resource usage and engineering required.
- Eliminates several difficult to setup components.
- Improves gameplay experience for participants.

Outline of presentation

AMRITA VISHWA VIDYAPEETHAM
University Established u/s 3 of UGC Act 1956
Amritapuri Campus

1. Challenges in existing attack-defence CTF game format and infrastructures
2. Overview of Docker and associated technologies
3. Container-based attack-defence CTF game infrastructure
4. Performance evaluation
5. Future work
6. Conclusion

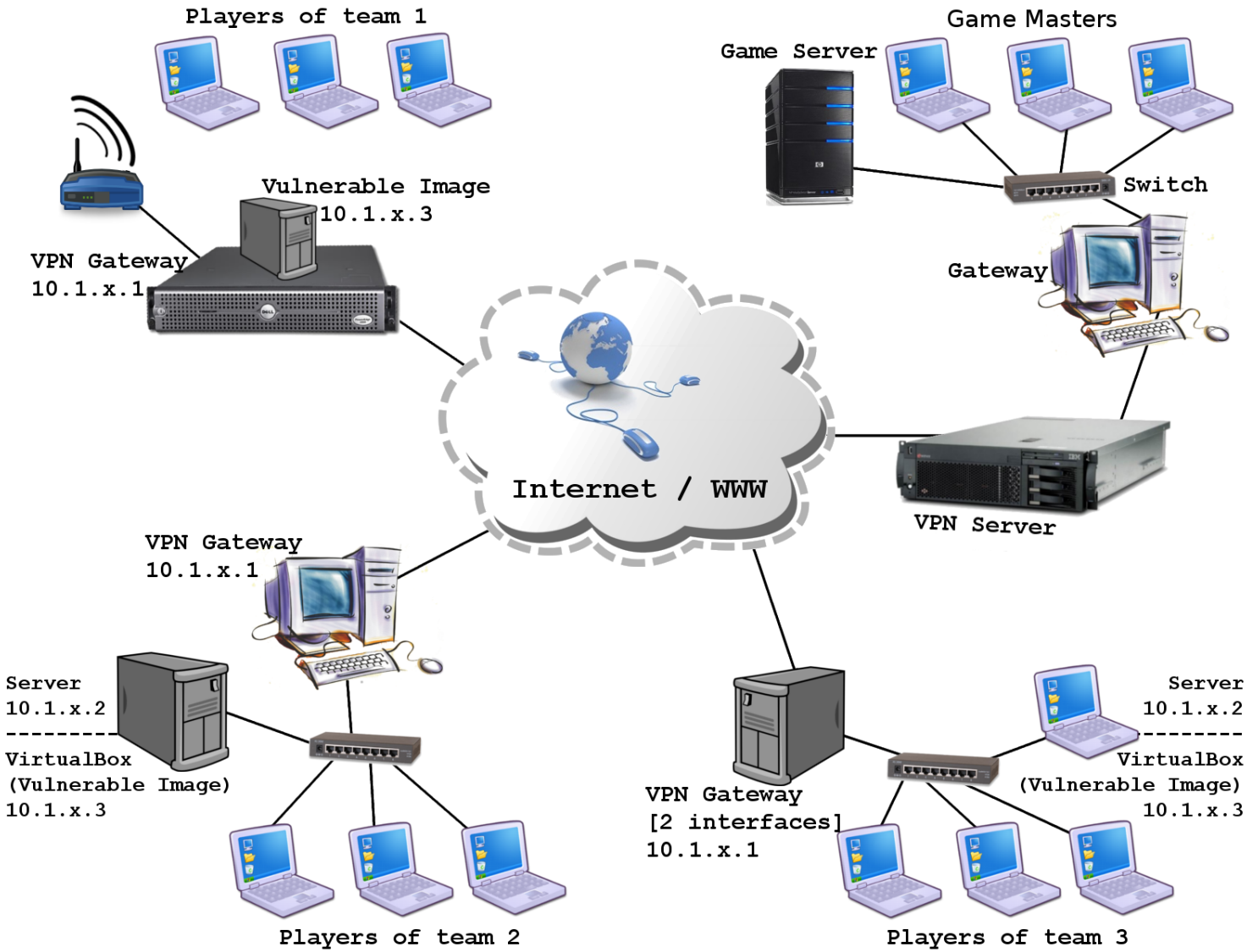
Outline

1. Challenges in existing attack-defence CTF game format and infrastructures
2. Overview of Docker and associated technologies
3. Container-based attack-defence CTF game infrastructure
4. Performance evaluation
5. Future work
6. Conclusion

Challenges

- 2 sources: gameplay and game infrastructure.
- Gameplay affects participants: requires doing too many tasks.
- Distracts them from primary objective.
- Infrastructure affects organizers and participants.
- 2 infrastructure types: distributed and centralized.

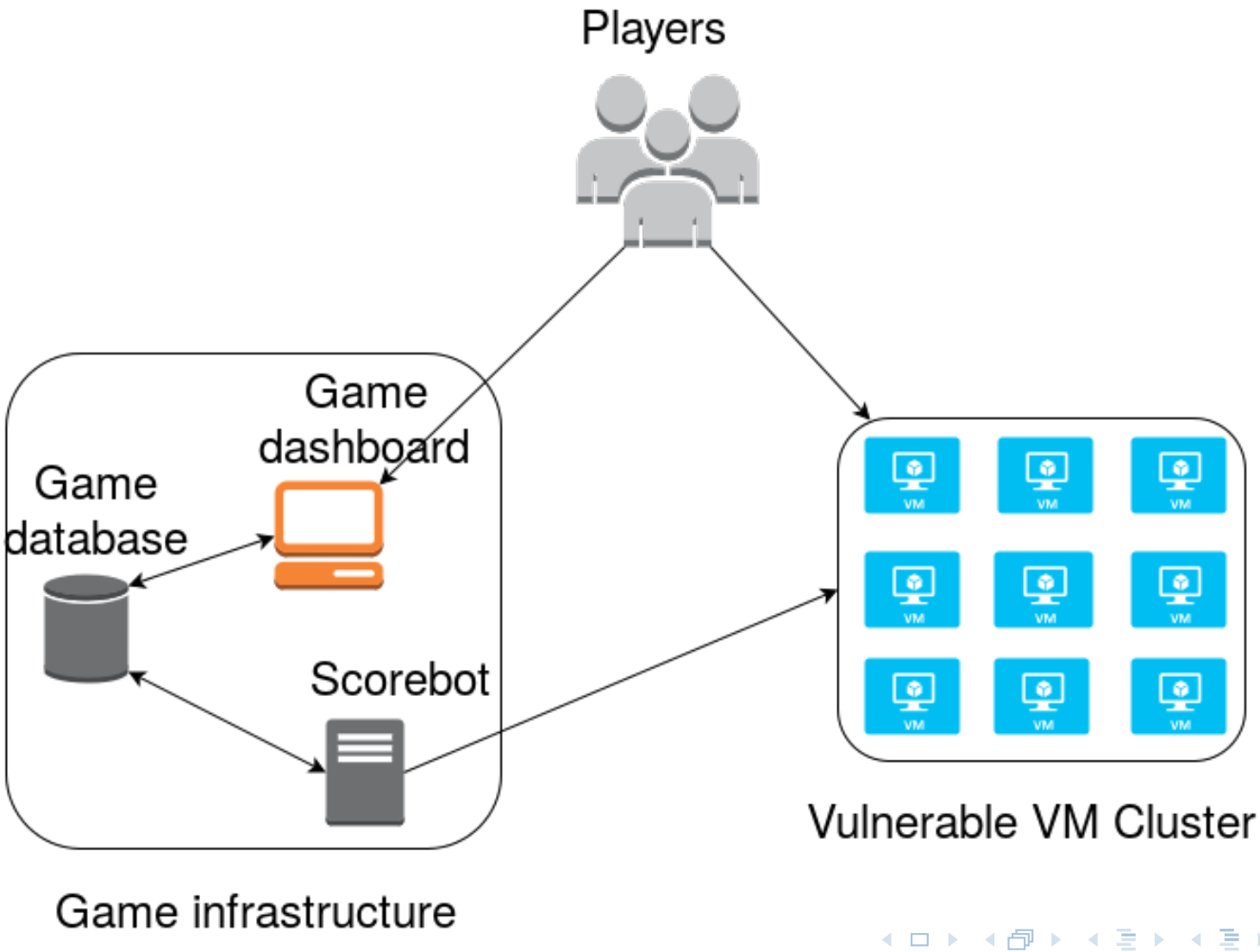
Distributed infrastructure



Challenges

- Organizers
  - Infrastructure needs lot of resources, engineering and monitoring.
  - eg: rwthCTF 2012's VPN server: 16GB RAM, 8 core i7 processor and 8 OpenVPN daemon processes.
- Participants
  - Difficult to obtain hardware such as computers and network switches/routers.
  - University IT policies prevent connecting to UDP based VPNs.

Centralized infrastructure



Challenges

- Organizers
  - Exponential increase in computing resources required.
  - Setting up exploit sandboxes, installing libraries and executing exploits.
- Participants
  - Network latency when accessing services.
  - Recreating services locally for analysis and testing is not straightforward.
  - Locked in to a standard exploit environment.

Outline

1. Challenges in existing attack-defence CTF game format and infrastructures
2. Overview of Docker and associated technologies
3. Container-based attack-defence CTF game infrastructure
4. Performance evaluation
5. Future work
6. Conclusion

Docker vs Virtual machines

Figure : Virtual Machines

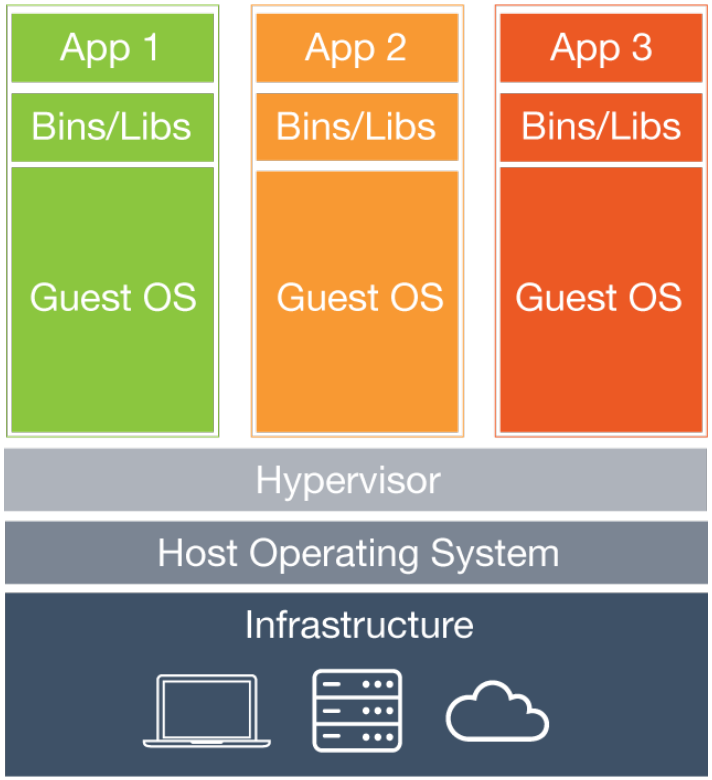
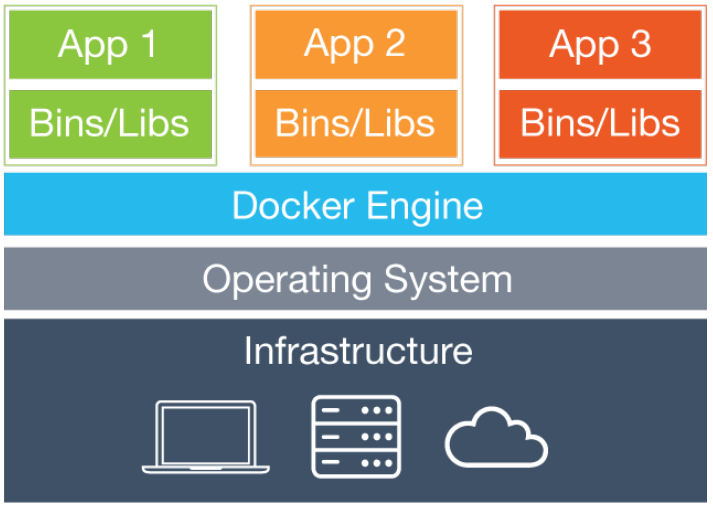


Figure : Docker containers



Images courtesy www.docker.com

Why Docker?

- Built-in container image reuse and extend capabilities.
- Remote API and programming language bindings aid in automation.
- Easy to share and distribute container images.
- Third party tools for container and image management.

Distribution and PORTUS

- Docker Inc's Distribution: Tool to manage container images - similar to a Git server.
- SUSE's PORTUS: Role-based access control of Distribution's images.
- Allows creating namespaces for teams and assigning different access levels to them.
- Alternatives: GitLab, Dockerhub, Amazon EC2 container service, Google Container Registry and more.

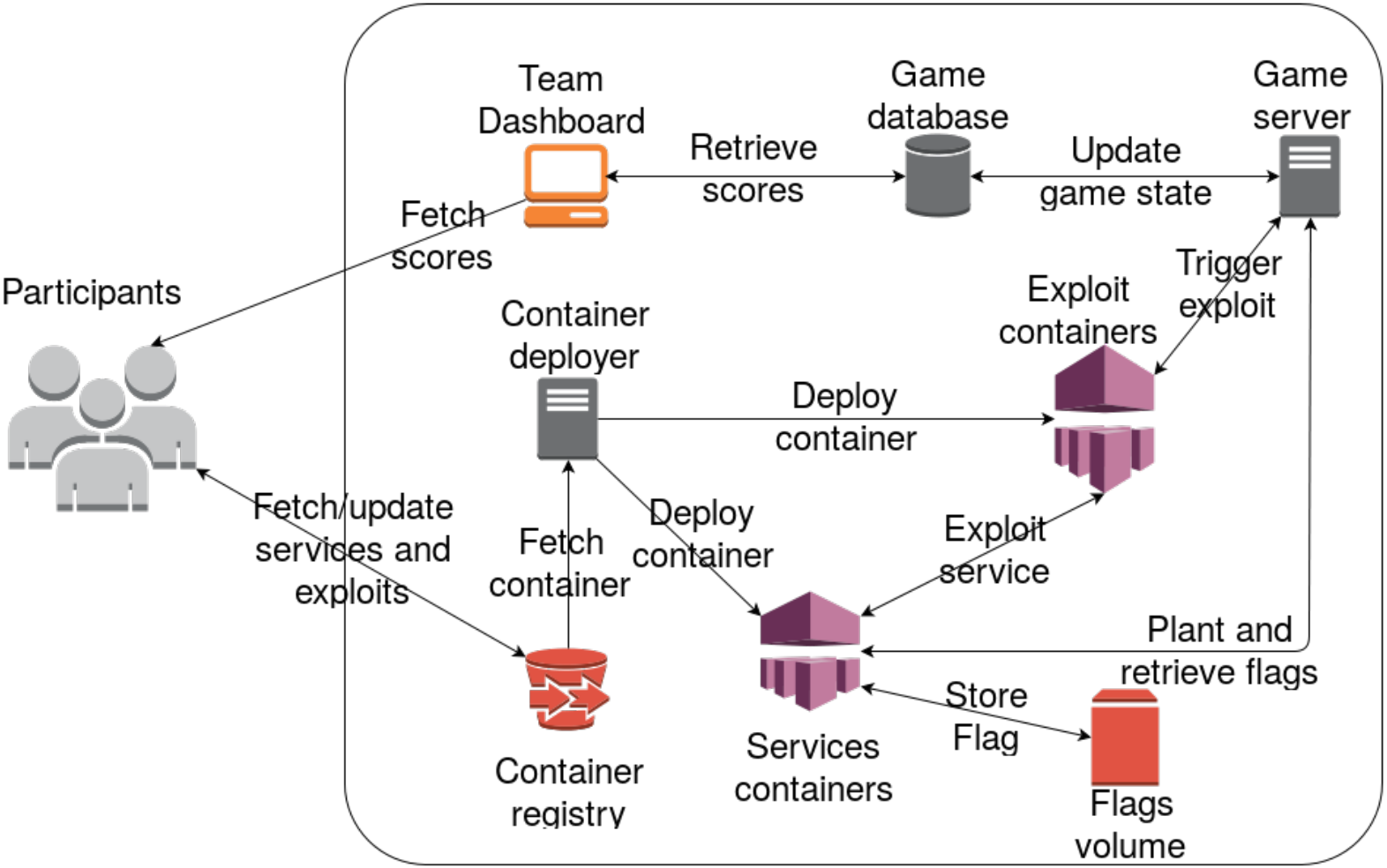
Outline

1. Challenges in existing attack-defence CTF game format and infrastructures
2. Overview of Docker and associated technologies
3. Container-based attack-defence CTF game infrastructure
4. Performance evaluation
5. Future work
6. Conclusion

Components

- **Container registry**: Git server like service for container images.
- **Container hosts**: Servers which run all the containers.
- **Service related containers**: Docker containers which either run a service or an exploit for a service.
- **Flag volume**: Docker volumes for persistent storage of flags.
- Modified versions of components of the iCTF centralized framework.

System design



Gameplay

Organizers

- Configure a CTF contest as desired.
- Build the service container images.
- Configure the container registry and upload service container images to it.
- Setup the game database and configure all game scripts.
- Optionally distribute encrypted copies of service container images to all teams.

Gameplay (cont.)

Participants

- Import the service container images from registry or organizer distributed copies.
- Analyze services for vulnerabilities, fix them and commit and upload changes to container registry.
- Create exploit containers for discovered vulnerabilities in accordance with the requirements, test them locally and upload them.

Game round overview

A game consists of several rounds with following phases

- **Synchronize**: All updated container images are synchronized with their live containers or images.
- **Store flags**: Flags are stored in all services of all teams and services' status is updated.
- **Run exploits**: All exploit containers are run against all services of all teams except exploit author.
- **Retrieve flags**: Flags stored earlier are retrieved, service status is updated and points are deducted if not retrieved successfully.

Benefits for organizers

- Lightweight game infrastructure.
- No need for engineering and monitoring VPN network.
- No need for configuring exploit environments.
- Tools like Docker swarm and Docker cloud further ease managing infrastructure.

Benefits for participants

- No additional hardware, dealing with IT policies or setting up VPN.
- No dealing with network latency: setup services locally.
- Infrastructure maintains service backups, simplifying gameplay.
- Fully customizable exploit environments.

Outline

1. Challenges in existing attack-defence CTF game format and infrastructures
2. Overview of Docker and associated technologies
3. Container-based attack-defence CTF game infrastructure
4. Performance evaluation
5. Future work
6. Conclusion

Experiments performed

- Two kinds of experiments
  - 3 services, 5 to 40 teams.
  - 30 teams, 1 to 8 services.
- Measure CPU utilization and memory usage for a 10 minute game round.
- Worst case: All teams write exploits for all services.
- Compare with estimated usage in VM based infrastructure.

Estimating VM resource usage

- Simulating requires high amounts of resources.
- Estimate based on requirements for InCTF's attack-defence round.
- 1GB RAM for 3 services found sufficient in past 5 editions.
- 200MB RAM per service and rest for the OS.

Observations

- Container server: 16GB RAM and 8 core Intel Core i5 2600 processor.
- Highest memory usage: 3.4GB and 4.4GB. Exploits included.
- Estimated usage for VMs: 40GB and 60GB. Exploits not included.
- Highest CPU usage observed 13% and 20%.
- Can easily handle loads comparable to most attack-defence CTFs today.

Outline

1. Challenges in existing attack-defence CTF game format and infrastructures
2. Overview of Docker and associated technologies
3. Container-based attack-defence CTF game infrastructure
4. Performance evaluation
5. Future work
6. Conclusion

Future work

- Develop techniques and identify tuning parameters to prevent overloading of Docker daemon with several simultaneous requests.
- Provide teams access to network traffic captures for reverse engineering exploits.
- Identify parameters to determine utility of CTF game infrastructures.
- Perform usability study of container-based infrastructure.

Outline

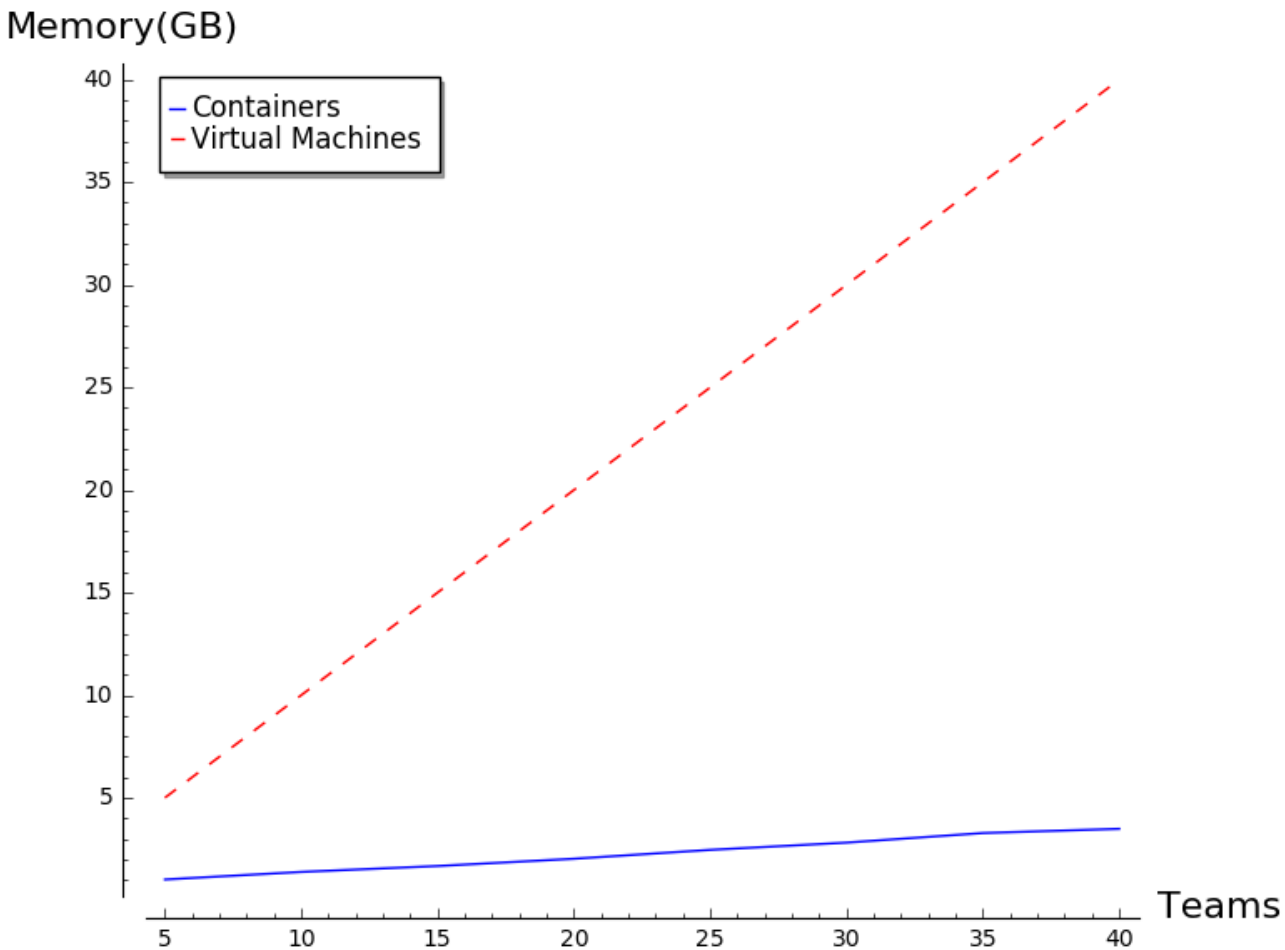
1. Challenges in existing attack-defence CTF game format and infrastructures
2. Overview of Docker and associated technologies
3. Container-based attack-defence CTF game infrastructure
4. Performance evaluation
5. Future work
6. Conclusion

Conclusion

- Existing attack-defence CTF game infrastructures are complex to setup and require several computing resources.
- Using application containers instead of virtual machines reduces resource requirement and engineering effort needed.
- Additional tools can improve gameplay experience for participants and further simplify infrastructure management.
- https://github.com/inctf/inctf-framework.

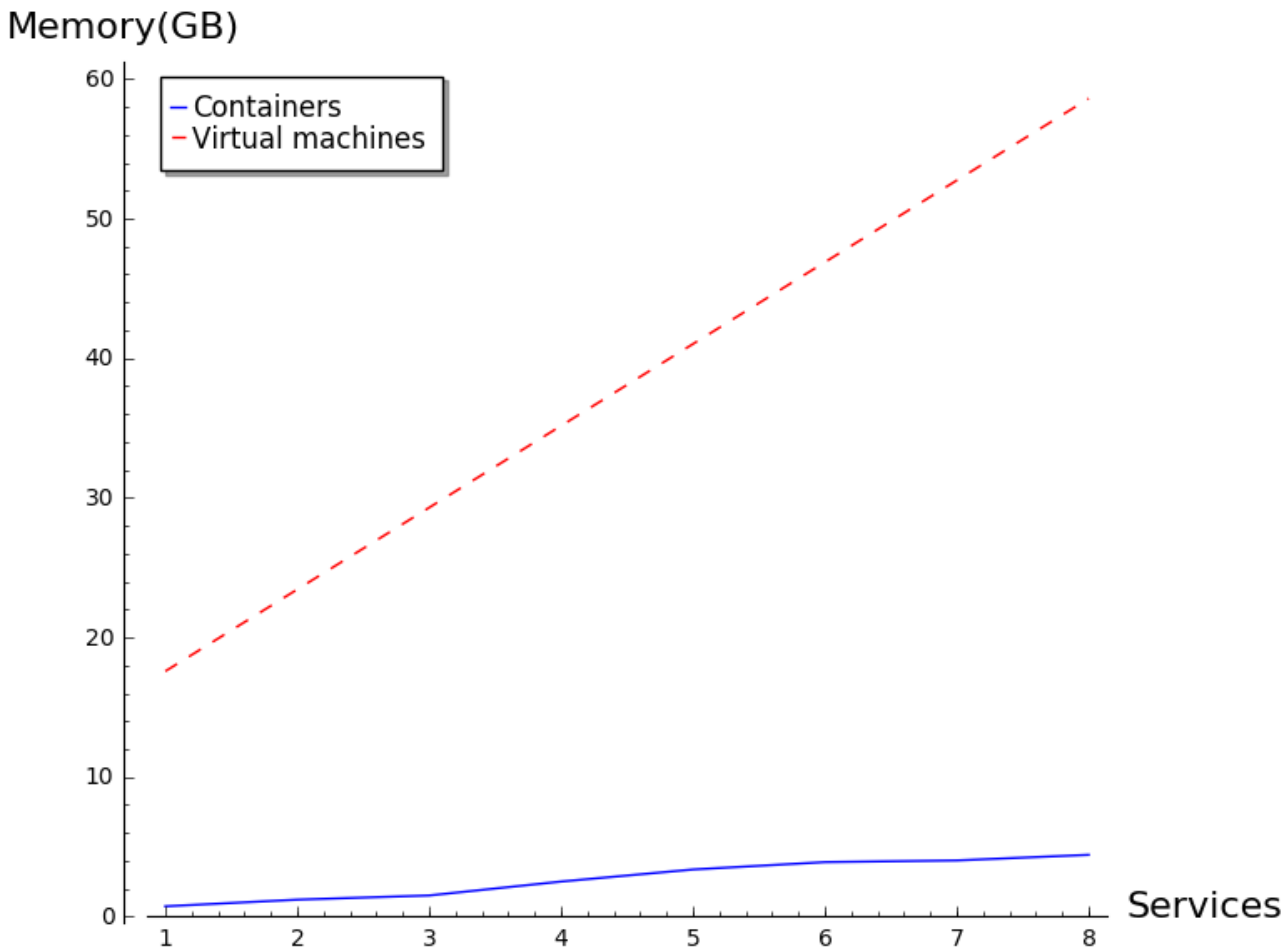
Observations

Figure : Average memory usage: 3 services, multiple teams



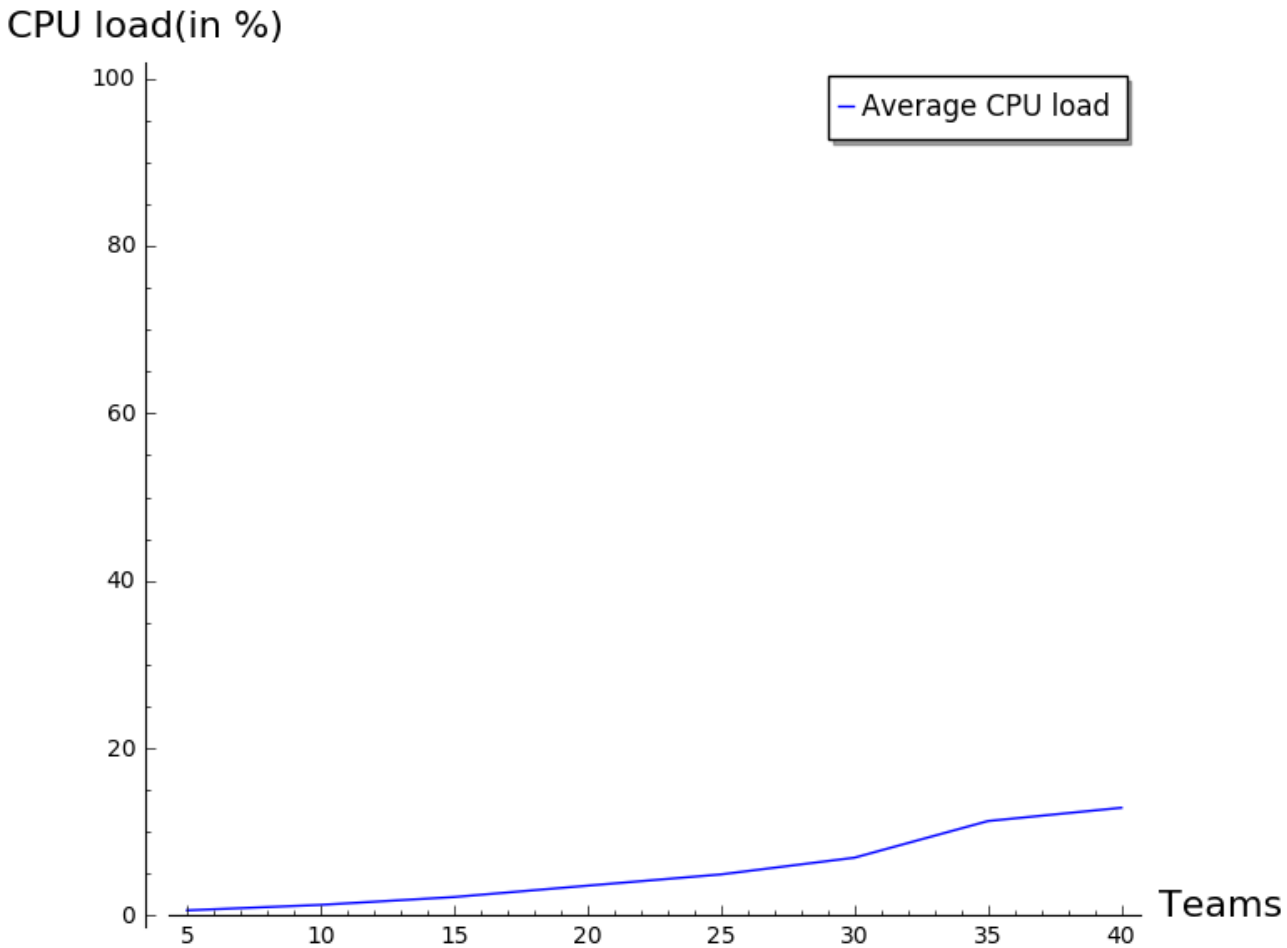
Observations(cont.)

Figure : Average memory usage: 30 teams, multiple services



Observations

Figure : Average CPU usage: 3 services, multiple teams



Observations(cont.)

Figure : Average CPU usage: 30 teams, multiple services