Gaetano Perrone

Tutor: Simon Pietro Romano

XXXIV Cycle - II year presentation

SECSI: SECurity Solutions for Innovation

Context

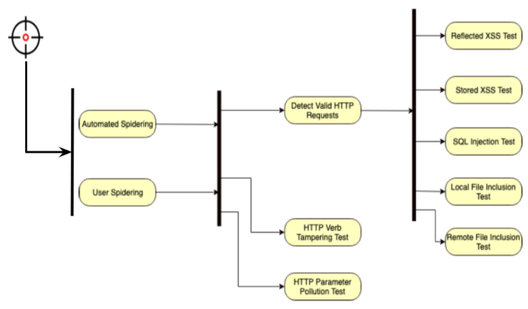
- Hacking Goals: a goal-centric attack classification framework
- Capturing flags in a dynamically deployed microservices-based heterogeneous environment

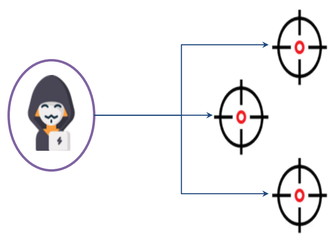
Hacking Goals

Attack classifications represent a crucial activity in security.

However, these classifications have been designed from the point of view of those defending a system.

We introduce a "goal-centric" methodology to classify attacks in terms of Hacking Goals



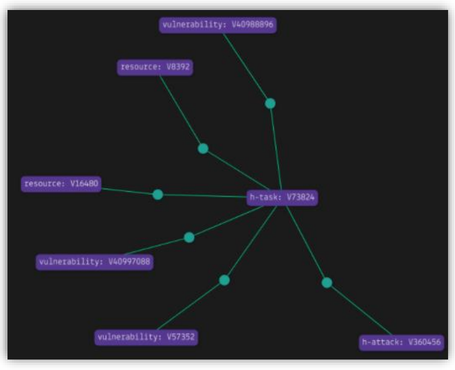


A **Hacking Goal** is a specific macro task that the attacker is going to achieve.

Hacking Tasks fullfil a Hacking Goal, generating a Hacking Tasks Tree

Hacking Actions are executed while performing a specific Hacking Task

The attacker behavior during a Penetration Test can be modelled through **Knowledge Graphs.**
KB represents a collection of interlinked description of entities – real-world objects and events.



It is possible to query Knowledge Base and find related tasks according to current goal.
- Hacking Goal is a Knowledge Graph query
- Hacking Tasks and Hacking Actions are entities in the Hacking Dependencies database
- Hacking Tasks Tree is built by using Knowledge Graph relations

```
match $t isa h-task;
$v isa vulnerability;
$s isa scope;
$s has name "Integrity";
$r1(exploits: $v, is-exploited: $s) isa vulnerability-to-scope;
$r2(exploits: $t, is-exploited: $v) isa vuln-to-task;
get $t; offset 0; limit 30;
```

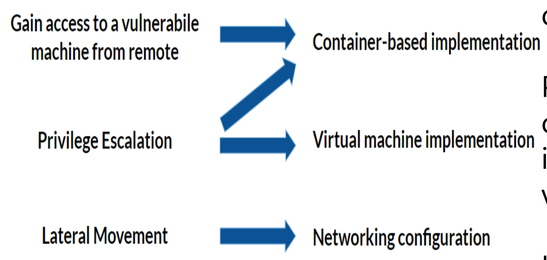
Capturing Flags

Capture the flag environment are training scenarios to learn cybersecurity.

The use of microservices in CTF envs can improves *scalability, decoupling and provisioning*.
However, container-based technology cannot reproduce every type of vulnerability

ID	Vulnerability Type	Can use Docker?
WAV	Web Application Vulnerabilities	Y
LAV	Linux-based Application Vulnerabilities	Y
SPEV	Privilege Escalation through services running with high privileges	Y
LUPEV	Linux-based User space privilege escalation	Y
MPEV	Privilege Escalation through Misconfiguration	Y
LMV	Linux Misconfiguration Vulnerabilities	Y
SV	Service Vulnerabilities	Y
NLAV	Non Linux-based Application Vulnerabilities	N
NLRV	Non Linux-based Remote Vulnerabilities	N
LKV	Linux Kernel-level Vulnerabilities	N
NLPEV	Non Linux-based Privilege Escalation Vulnerabilities	N

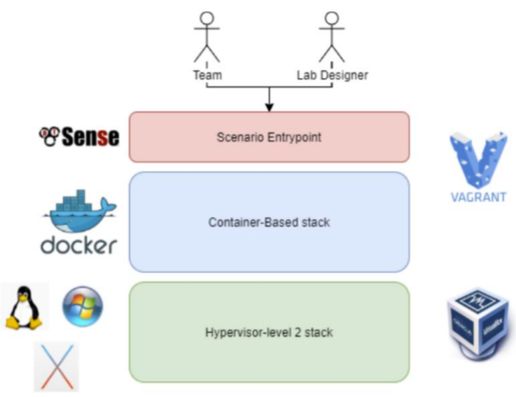
Mapping between CTF Requirements and Solution design choices



First Remote Access usually exploits user-space vulnerabilities => container virtualization

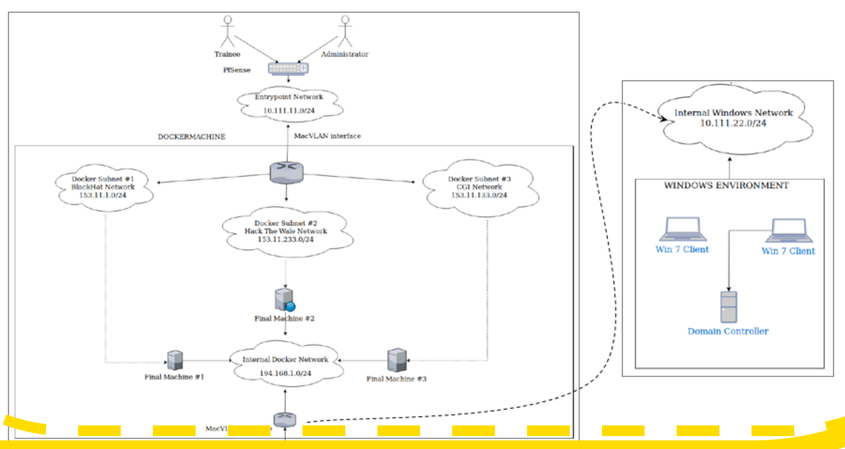
Privilege Escalation vulnerabilities can happen both in user-space and in kernel-space => Hybrid virtualization

Lateral Movement => network configurations.



Virtual Scenario layers
- ***Scenario Entry-point***: to provide each team with remote access to the emulated scenario
- ***Container-Based Stack***: this stack is composed by several Docker hosts that use OS-level virtualization
- ***Hypervisor-level 2 Stack***: this stack is composed by Virtual machines that do not use OS-level virtualization

Virtual Scenario Implementation used for a CTF event



Research Group

Member of the SecSI research Team

Collaborations

Founder of SecSI innovative cybersecurity startup. We collaborate with several companies.

ACCA SOFTWARE

NTT DATA

Future Developments

- Extend the behavioral approach to other domains (Infrastructural, Mobile)
- Development of a support decision system for penetration testing based on Knowledge Graphs
- Development of a generic testing methodology to detect injection vulnerabilities in Web Applications
- Define a formal vulnerability declarative description language for Cyber Range Scenarios