Under the Hood of MARVEL

An Anti-Repackaging Solution Based on Android Virtualization

UNIVERSITÀ DEGLI STUDI DI GENOVA
Dibris
UNIVERSITÀ DEGLI STUDI DI PADOVA

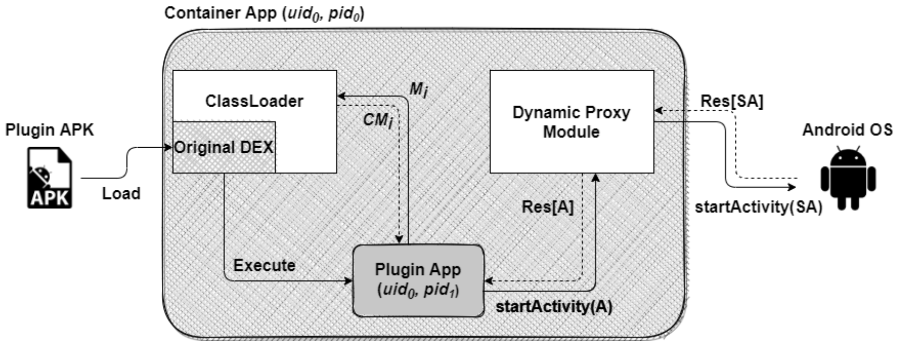
Alessio Merlo
Antonio Ruggia
Luca Verderame
name.surname@dibris.unige.it

Eleonora Losiouk
Mauro Conti
surname@math.unipd.it

Agenda

- Basic Concepts
- MARVEL
- MARVELoid
- Experimental Campaign
- Experimental Results
- Demo
- Conclusion & Future work

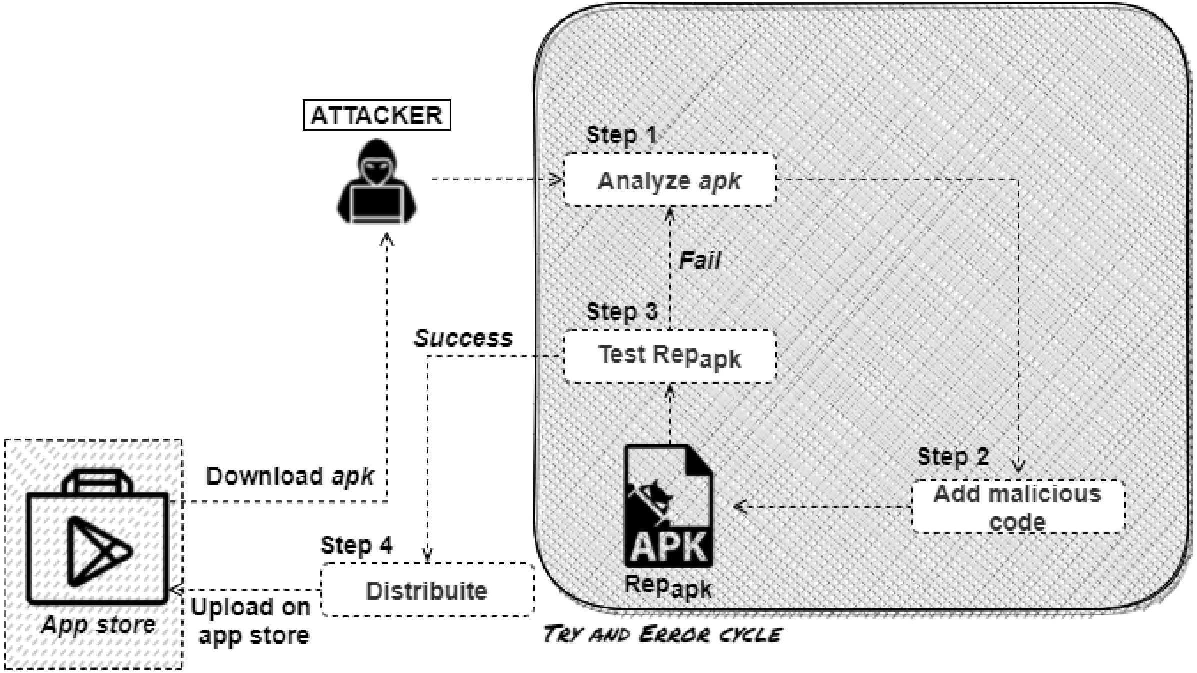
Android Virtualization (AV)



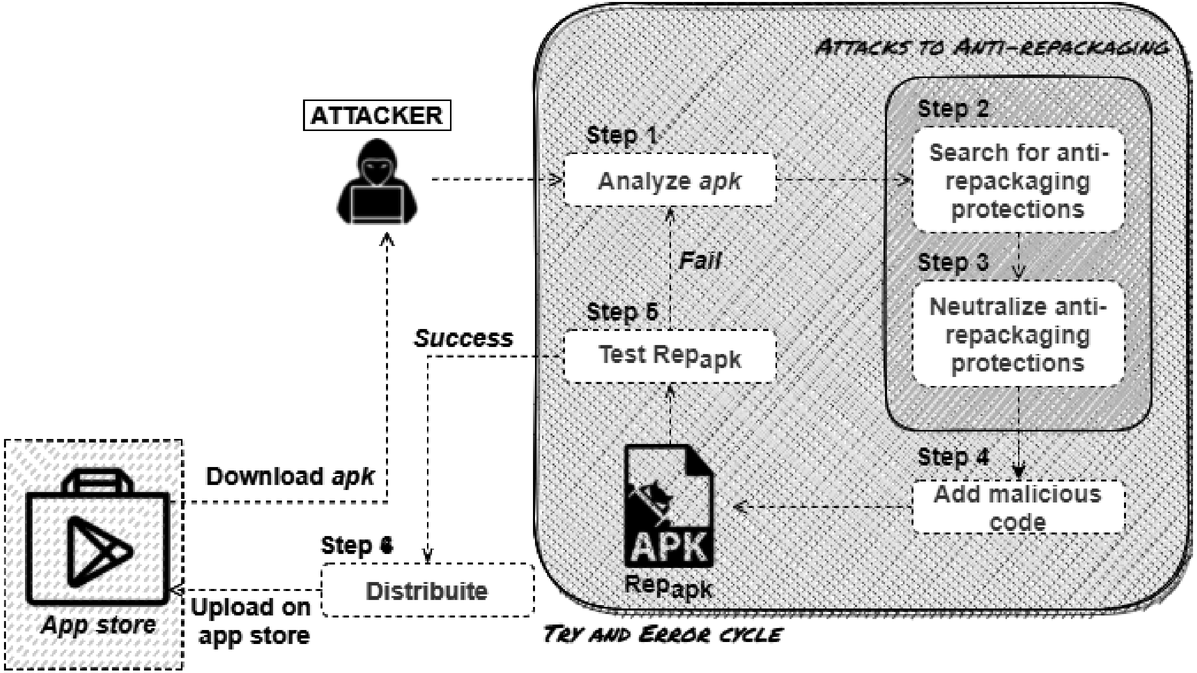
AV allows to execute an Android app (*plugin*) within the context of another app (*container*).

- **Dynamic Code Loading** allows the Java code that is not known about before a program starts
- **Java Reflection** allows a Java program to examine or "introspect" upon itself
- **Java Dynamic Proxy** creates a Proxy object to serve/handle multiple method calls

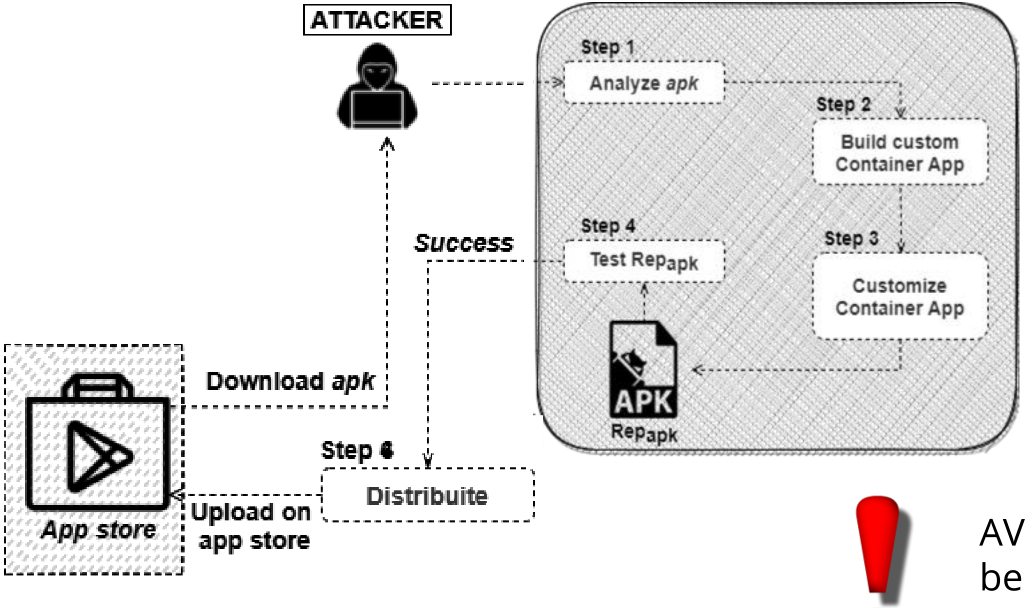
Android App-Repackaging



Android App-Repackaging



Android App-Repackaging



AV allows to modify the behavior of the app without repackaging it

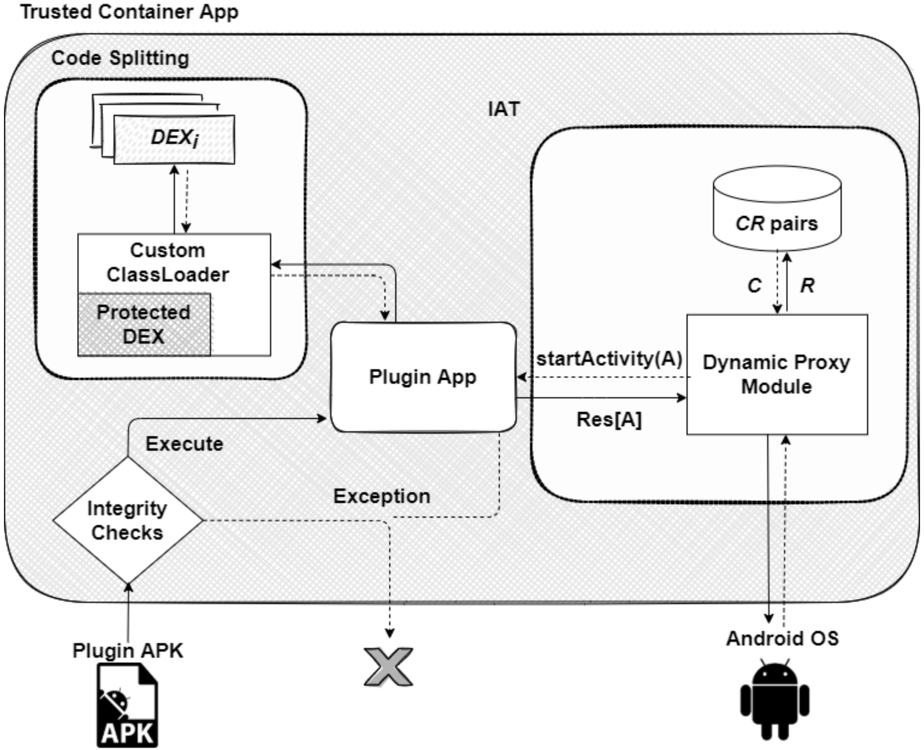
MARVEL ∘ Goal

Mobile-app Anti-Repackaging for Virtual Environments Locking

- (G1) Preventing the attacker from being able to statically analyze an app
- (G2) Preventing an app from being executed in a malicious container
- (G3) Detecting an intermediate malicious container executes a plugin

MARVEL ∘ Overview

- An app can be executed only by the Trusted Container (TC)
- Mutual verification between plugin and TC app
- Code splitting between plugin and TC app



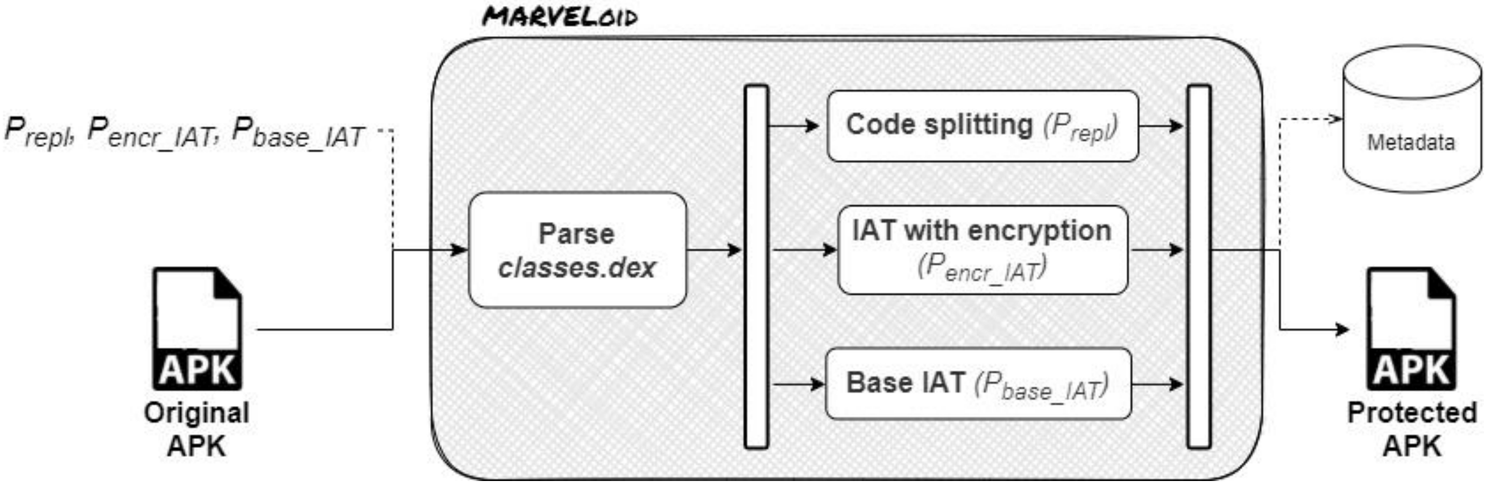
MARVEL ∘ Implementation

Artifacts Evaluated acm Reusable V1.1

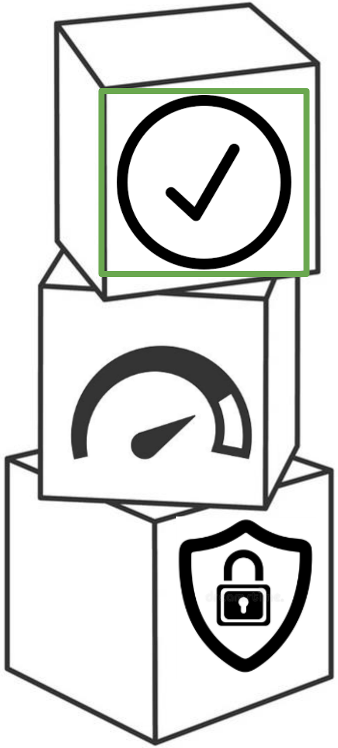
- MARVELoid
  - A Java tool to protect Android apps
  - Handles the code splitting and injections of Interconnected Anti-Tampering Controls (IAT).
- Trusted Container
  - A virtualization app that is built on top of the official *VirtualApp* framework
  - Responsible for the enforcement of the MARVEL runtime protection.

The source code is available at: https://github.com/totoR13/MARVEL

MARVELoid


MARVELoid
P_{repl}, P_{encr_IAT}, P_{base_IAT}
Original APK
APK
Parse *classes.dex*
Code splitting *(P_{repl})*
IAT with encryption *(P_{encr_IAT})*
Base IAT *(P_{base_IAT})*
Metadata
Protected APK
APK

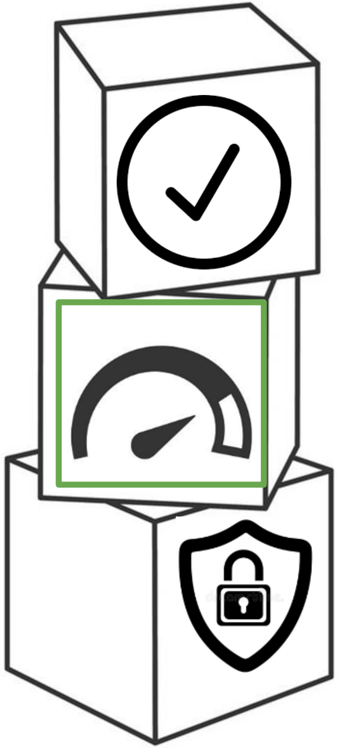
Experimental Campaign ∘ Goals



Correctness:

- Fault of MARVELoid
- Fault at runtime

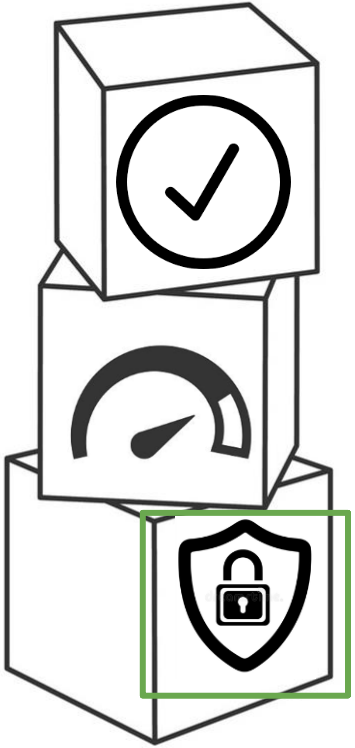
Experimental Campaign ∘ Goals



Performance:

- Protection time
- Space overhead
- Runtime resource overhead

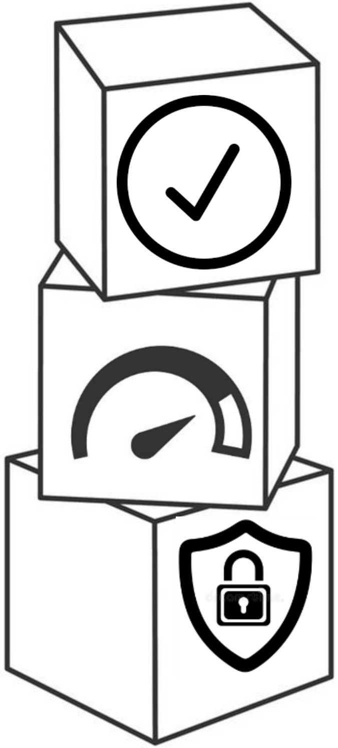
Experimental Campaign ∘ Goals



Security:

- Injected protection mechanisms
- Attacker process

Experimental Campaign ∘ Definitions



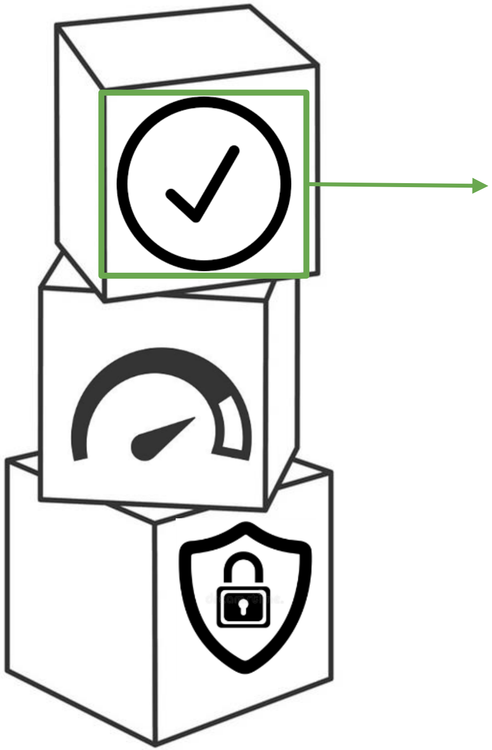
Static Analysis:

- Evaluate the MARVELoid tool

Dynamic Analysis:

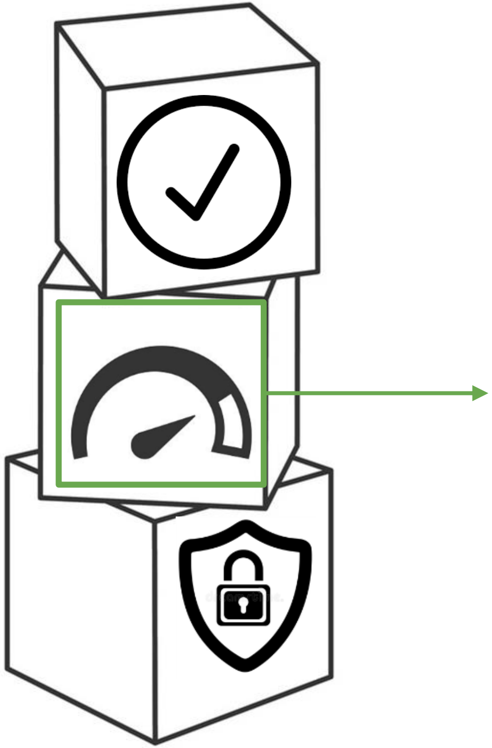
- Evaluate the resources overheads
- Evaluate the Trusted Container

Experimental Campaign ∘ Implementation



- Static Analysis: **Automatic**
- Dynamic Analysis: **Automatic**

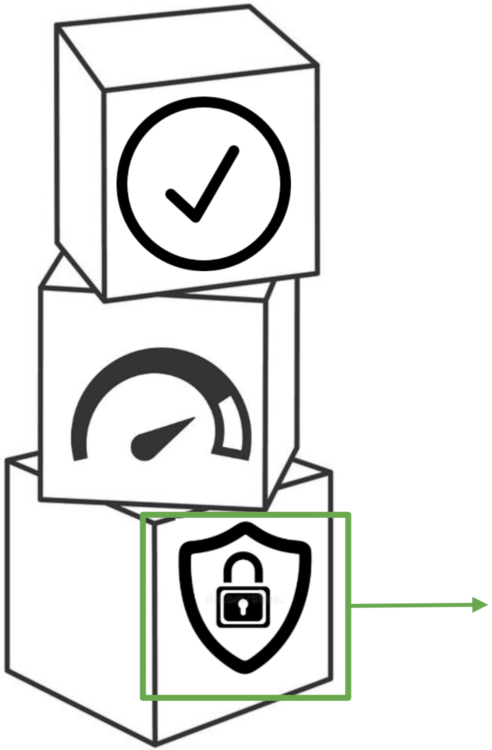
Experimental Campaign ∘ Implementation



- Static Analysis: **Automatic**
- Dynamic Analysis: **Automatic**

- Static Analysis: **Automatic**
- Dynamic Analysis: **Automatic**

Experimental Campaign ∘ Implementation



- Static Analysis: **Automatic**
- Dynamic Analysis: **Automatic**

👍

- Static Analysis: **Automatic**
- Dynamic Analysis: **Automatic**

👍

- Static Analysis: **Automatic**
- Dynamic Analysis: **Manual**

👎

Dynamic Analysis ∘ ARES

Black-box tool that uses Deep Reinforcement Learning to test Android apps

- Install and launch Android apps
- Generate a sequence of input depending on the view items

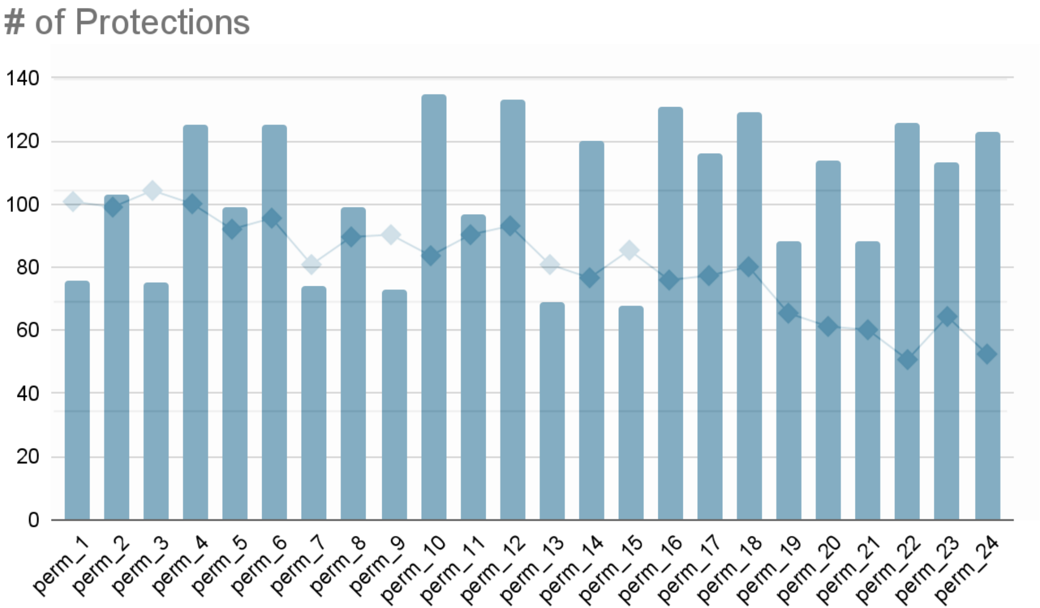
The source code is available at: https://github.com/H2SO4T/ARES

Dynamic Analysis ○ ARES++

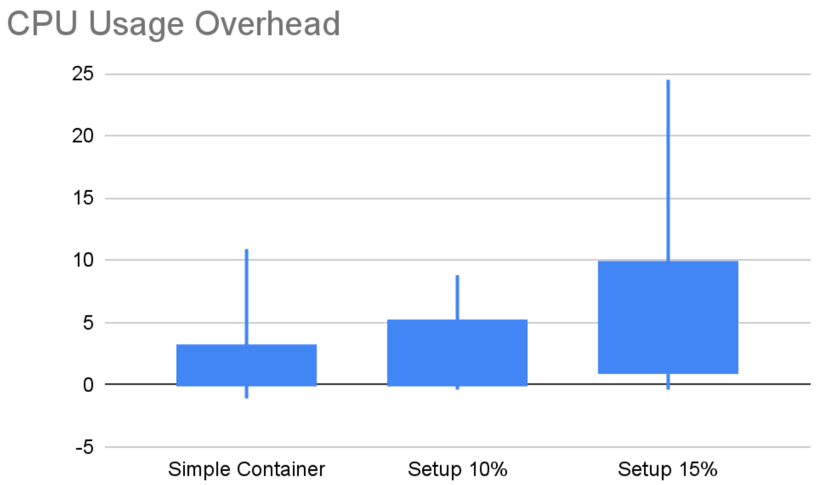
We extended ARES to:

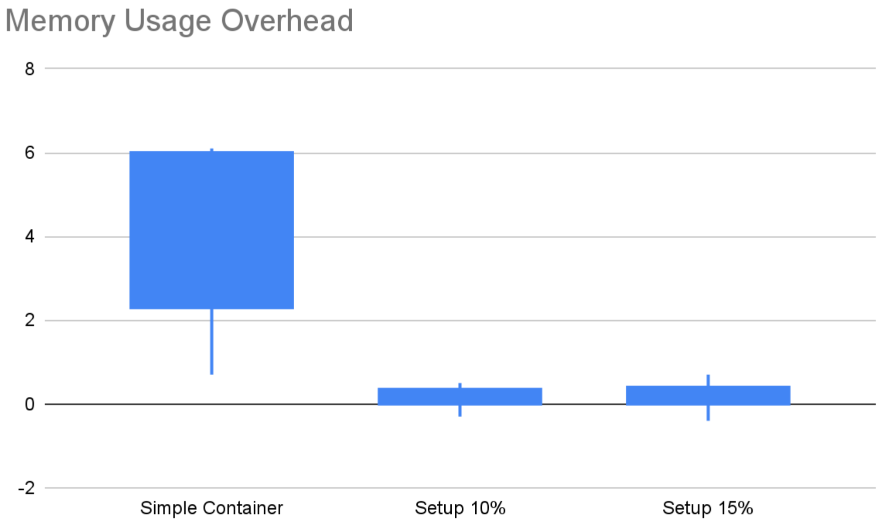
- Execute several plugin app in a container app
- Retrieve memory and CPU usage
- Dump the extracted values into a database

Experimental Campaign ∘ Static Results



Experimental Campaign ∘ Dynamic Results





Demo Time …

… it’s over!

Test limitation & Conclusions

Limitation

- Manual inspection for runtime security evaluation

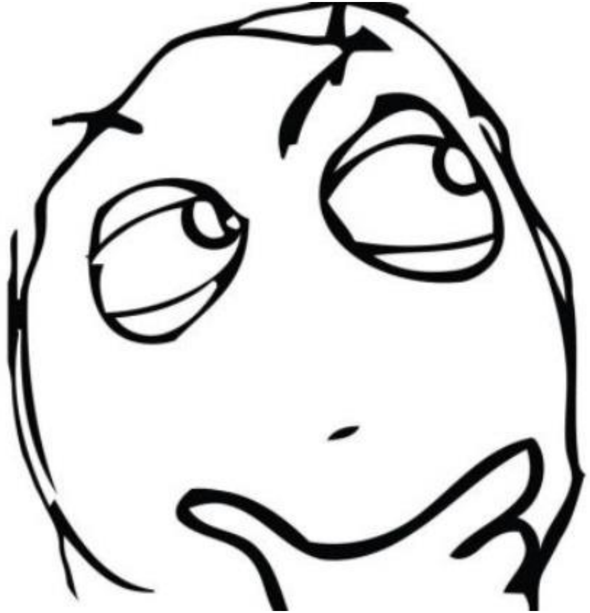
Future Improvements

- Extends the testing pipeline to add more features (exception analyzer)

Good practices

- Tools and experimental evaluation available to the community

Question & Answer



Thank you !!!