



***“ Youth is happy because it has the capacity to see beauty. Anyone who keeps the ability to see beauty never grows old.” - Franz Kafka***



# Kid

#6 There's Two Types Of Kids On The First Day Of School



MadeCrabs Report

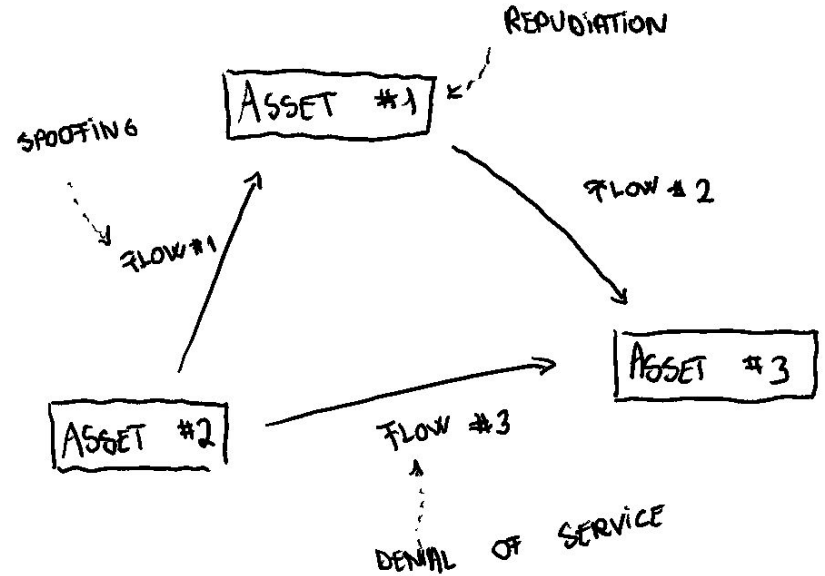
Final score: 194 points

Add a comment... POST

Bruno Alexandre 7 years ago  
The terror in her eyes, poor child.  
82 Reply  
View More Replies...

Pleasure maximization

# Vulnerability researcher



Threat modelling

# Kid

## #6 There's Two Types Of Kids On The First Day Of School



by MudoCrabs Report

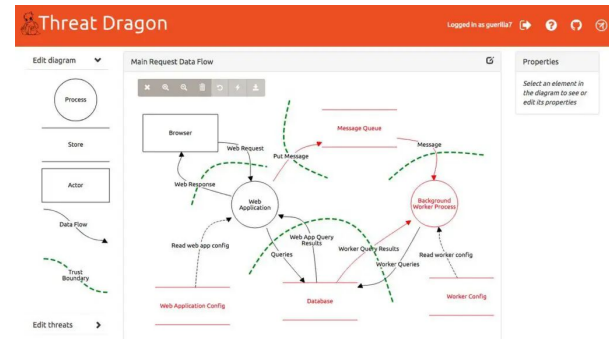
Final score: 194 points

Add a comment... POST

**Bruno Alexandre** 7 years ago  
The terror in her eyes, poor child.  
82 Reply  
View More Replies...

Pleasure maximization

# Vulnerability researcher



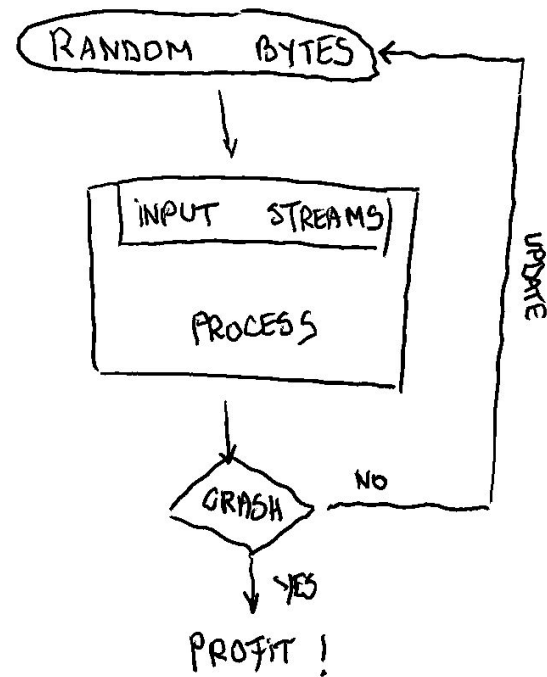
Threat modelling with Threat Dragon

## Kid



Curiosity

## Vulnerability researcher



Fuzzing

## Kid



Curiosity

## Vulnerability researcher



```
$ AFL_USE_ASAN=1 /AFLplusplus/afl-cc -g -o program.elf program.c  
$ afl-fuzz -i inputs/ -o outputs/ -- program.elf @@
```

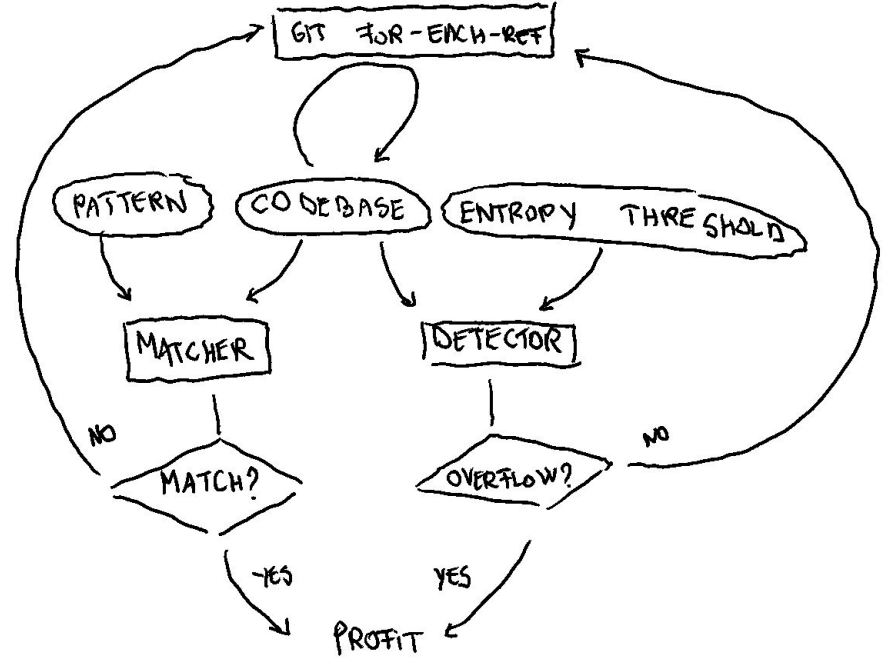
Fuzzing with AFL++

## Kid



Sugar addiction

## Vulnerability researcher



Secret scanning



## Kid



Sugar addiction

## Vulnerability researcher



```
$ gitleaks detect --report-path report.json
```

Secret scanning with Gitleaks

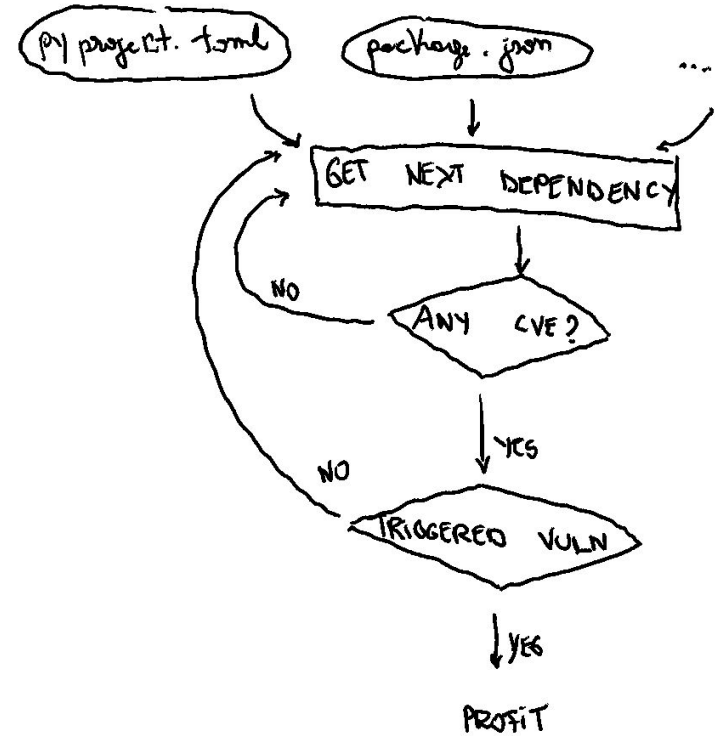
## Kid

My grandma when she sees me every year:



Persuasion

## Vulnerability researcher



Dependency scanning

## Kid

My grandma when she sees me every year:



Persuasion

## Vulnerability researcher




```
$ osv-scanner --lockfile codebase/portrait/poetry.lock
```

Dependency scanning with OSV-Scanner

### AFLplusplus

The fuzzer afl++ is afl with community patches, qemu 5.1 upgrade, collision-free coverage, enhanced laf-intel & redqueer AFLfast++ power schedules, MOpt mutators, unicorn\_mode, and a lot more!

 C ★ 4.1k  829

### gitleaks

Protect and discover secrets using Gitleaks

 Go ★ 14.1k  1.3k

### osv-scanner

Vulnerability scanner written in Go which uses the data provided by <https://osv.dev>

 Go ★ 5.4k  292

### threat-dragon

An open source threat modeling tool from OWASP

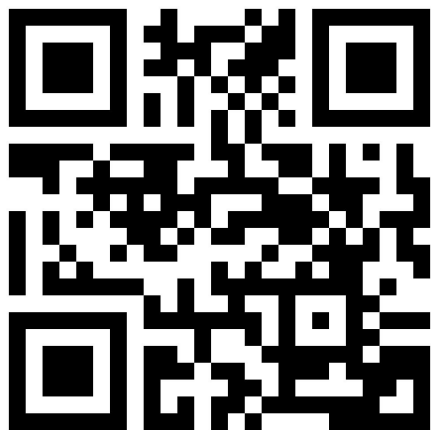
 JavaScript ★ 656  182



 [oss\\_fortress](#)

Workshop about finding vulnerabilities in code using open source tools

 HTML  3  1



# Ubuntu Summit 2023

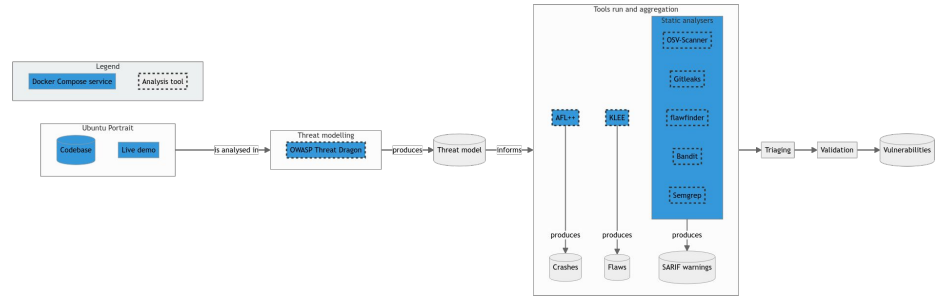
Nov 3–5, 2023  
Riga, Latvia  
Europe/Riga timezone



Industrial-grade  
Ubuntu Landcape



Vulnerable-by-design  
Ubuntu Portrait







## Writing rules

### 📄 INFO

For live testing of your rules, you can also use [the Playground](#).

1. The vulnerabilities listed below were not detected by any technique that we've seen so far. Inspect [the Semgrep documentation](#) and write rules to catch them in the `/root/analysis/semgrep-rules` folder. The rules should have as many metadata fields filled as possible.

Vulnerability ID	Hints on how to catch it with Semgrep
<code>UBUSEC-SECRET-LOG</code>	Calls to functions from <code>logging</code> where the parameters have sensitive names
<code>UBUSEC-UID-IDOR</code>	<code>execute_string_command</code> calls with dynamic arguments
<code>UBUSEC-ARCHIVE-WRITE</code>	<code>os.path.join</code> calls where the arguments came from the parameters of the function
<code>UBUSEC-HASH-LEN</code>	<code>sha256_update</code> calls where the parameter is created by concatenation

2. Modify the command from the first section to use them and to save the resulting SARIF file in `/root/analysis/semgrep.custom.sarif`.

### 🔧 SOLUTION

To display the solution of this task, enter the text `i-surrender-to-the-code-security-gods` in the field below.



