Detecting Bugs at Scale with Semgrep

Parsia Hakimian
parsiya.net

Image credit: *Lost in Random*, Zoink/Electronic Arts

TL;DW

Too Long;
Didn't Watch!

Image credit: Myself - The Semgrep Logo belongs to r2c

Yours Truly

2013-2019: Cigital/Synopsys - consulting

Web -> Mobile -> Thickclients/Blockchain (lol)

DEF CON 26

Tineola Taking a Bite Out of Enterprise Blockchain

2019 - 2022: Electronic Arts - security “engineer”

Videogames -> Dynamic testing ->

Source code review -> Semgrep

DEF CON 28 Appsec Village

localghost: Escaping the Browser Sandbox without 0-Days

2022: Unemployed and a burden on society

Episode 1: Static Analysis

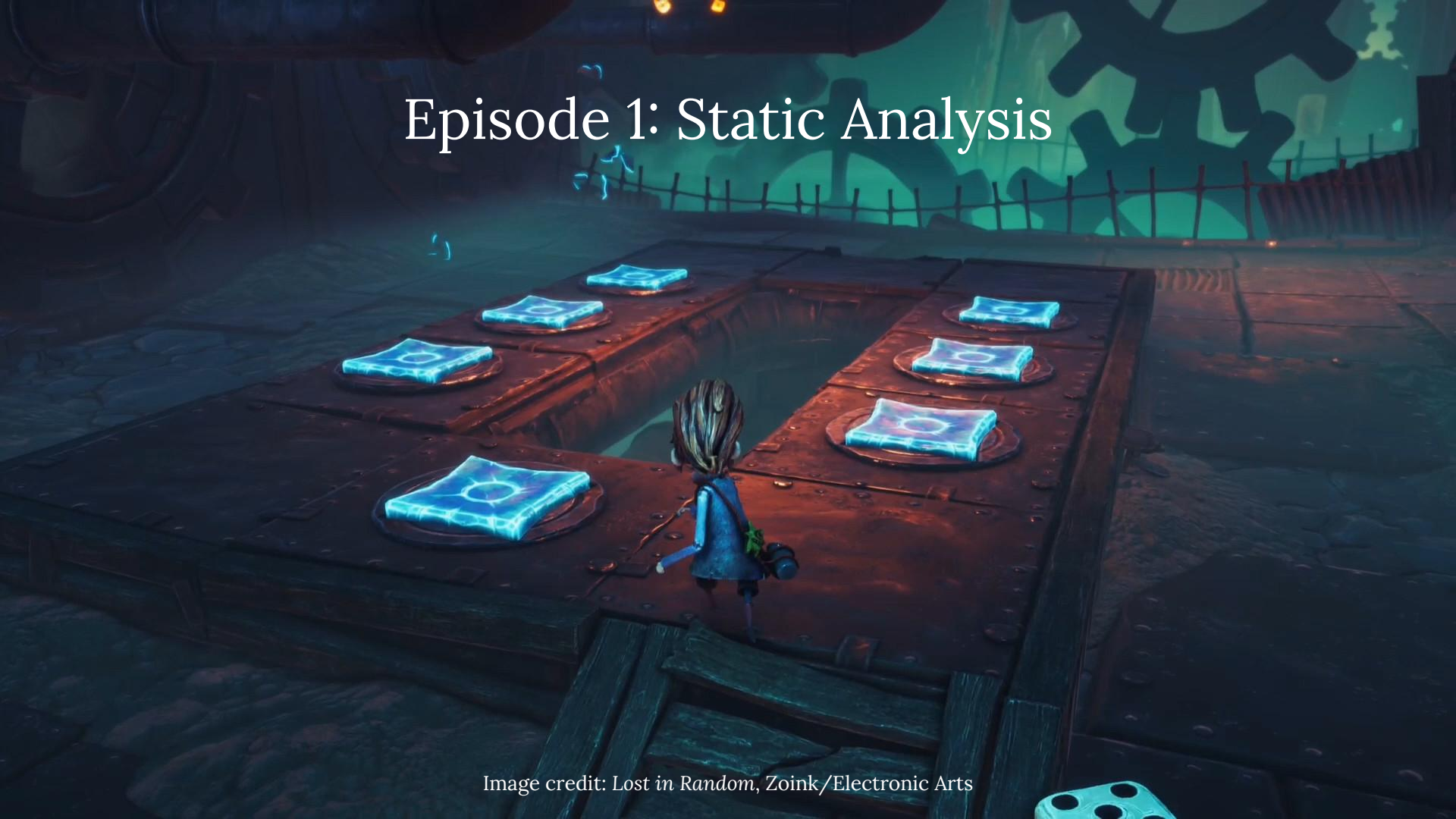


Image credit: *Lost in Random*, Zoink/Electronic Arts

Inside You There are Two Wolves

Security Engineer

Developer

Image credit: Victor Becker
Lost in Random, Zoink/Electronic Arts

Inside Your SAST There are Two Wolves

For Security Engineers	**For Developers**
Some noise is OK.	Any noise is bad!
Can be slow.	Must be quick*.
Manual usage is tolerated.	Should be part of CI/CD.
Alert on everything.	Never break the build.
Configuration/tuning is expected.	The tool should work out of the box.

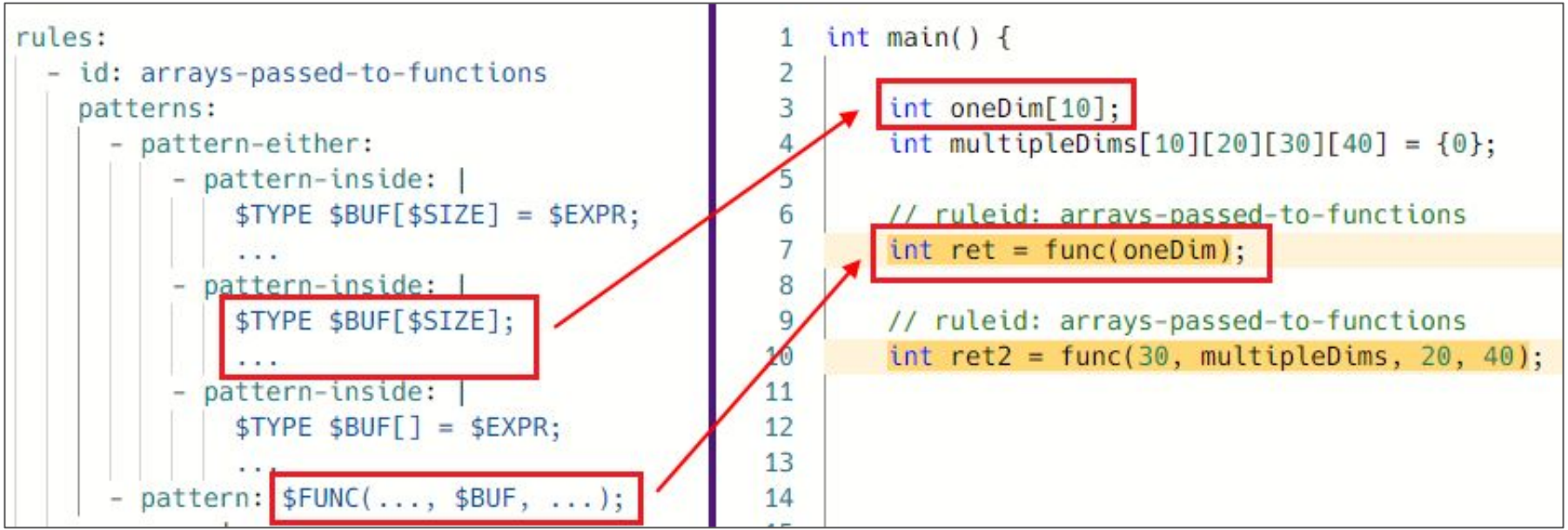
Episode 2: Semgrep

Image credit: *Lost in Random*, Zoink/Electronic Arts
The Semgrep logo belongs to r2c

Semgrep vs. CodeQL vs. Coverity vs. … !

Image credit: *Lost in Random*, Zoink/Electronic Arts

Rules are Easy!



Semgrep Limitations

Weak multi-file analysis - Limited taint tracking
Join Mode, DeepSemgrep (paid)

Limited set of available rules
We must create custom rules

Image credit: *Lost in Random*, Zoink/Electronic Arts

Episode 3: The Lay of the Land

Image credit: Leo Brynielsson
Lost in Random, Zoink/Electronic Arts

Code, Configs and more Code!

Image credit: *Lost in Random*, Zoink/Electronic Arts

Dependency Hell

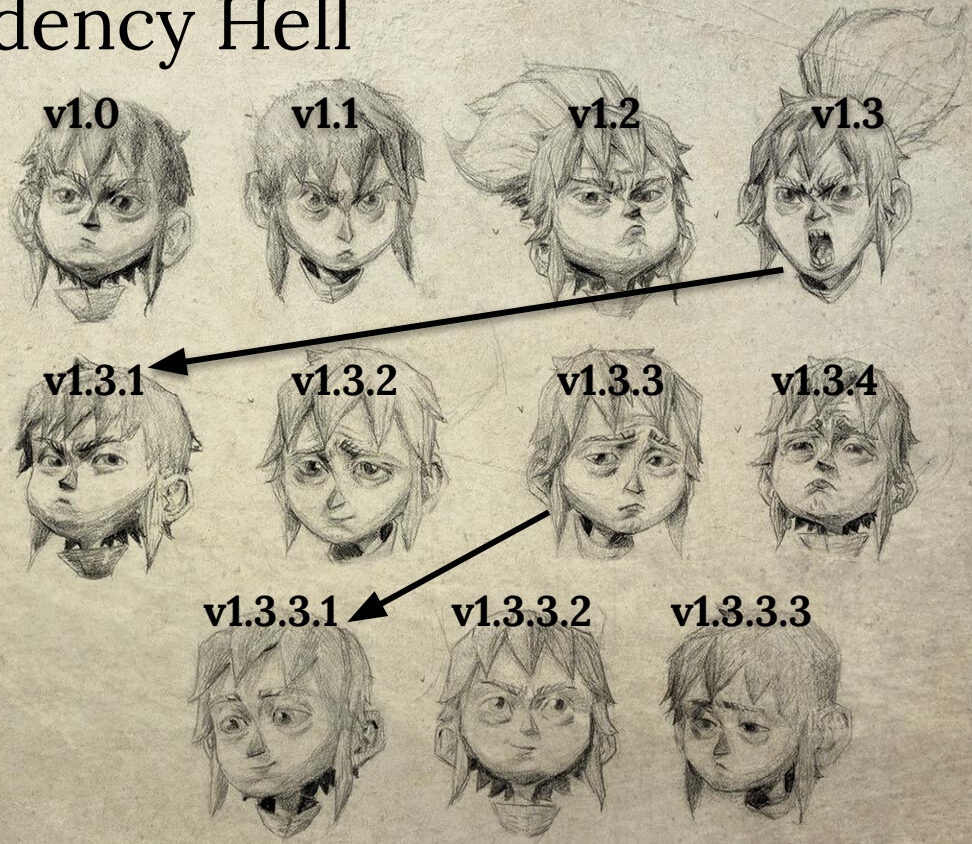


Image credit: Victor Becker
Lost in Random, Zoink/Electronic Arts

Source Code Management

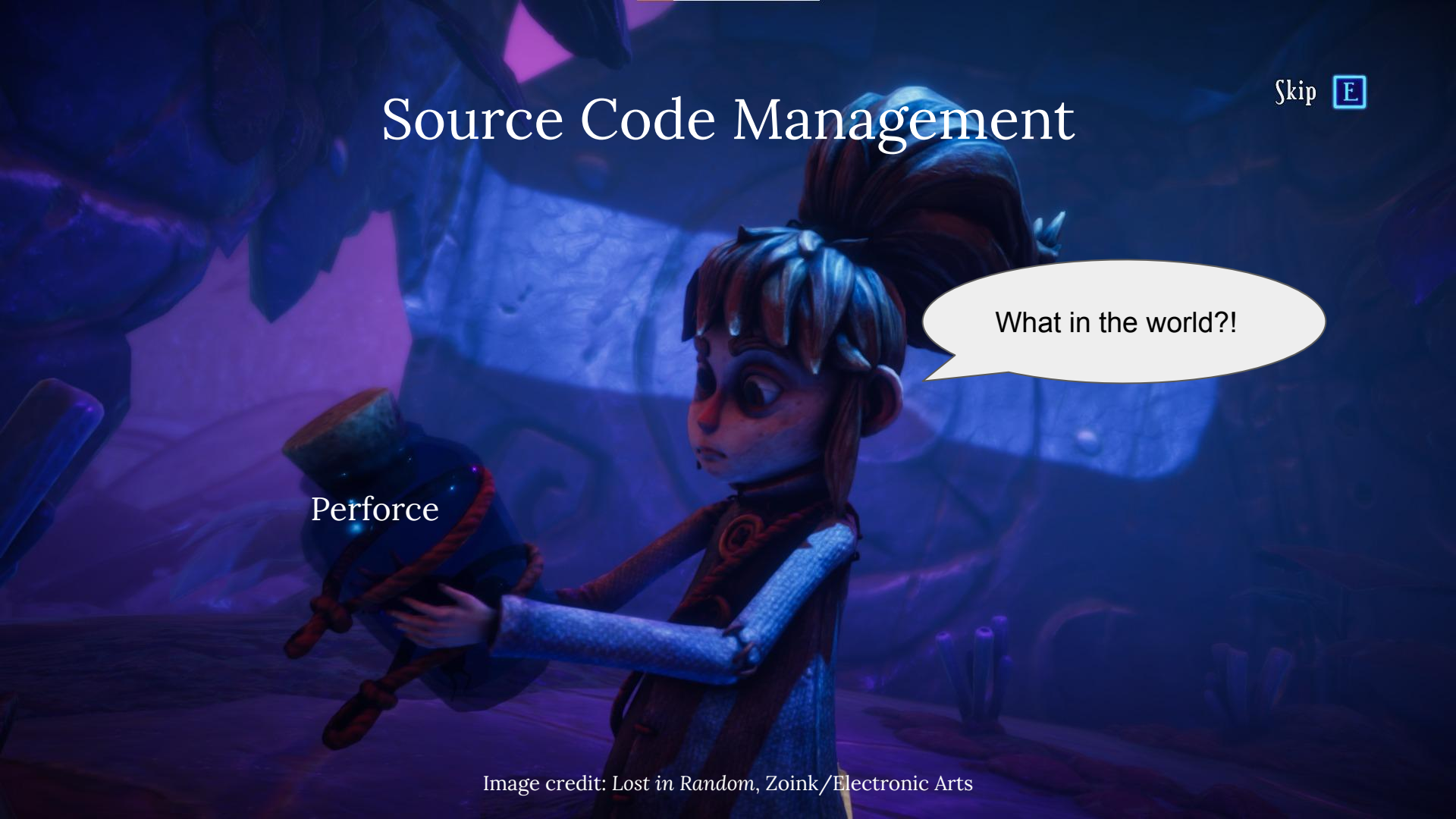


Image credit: *Lost in Random*, Zoink/Electronic Arts

~~Webview~~ Chromium Desktop Apps

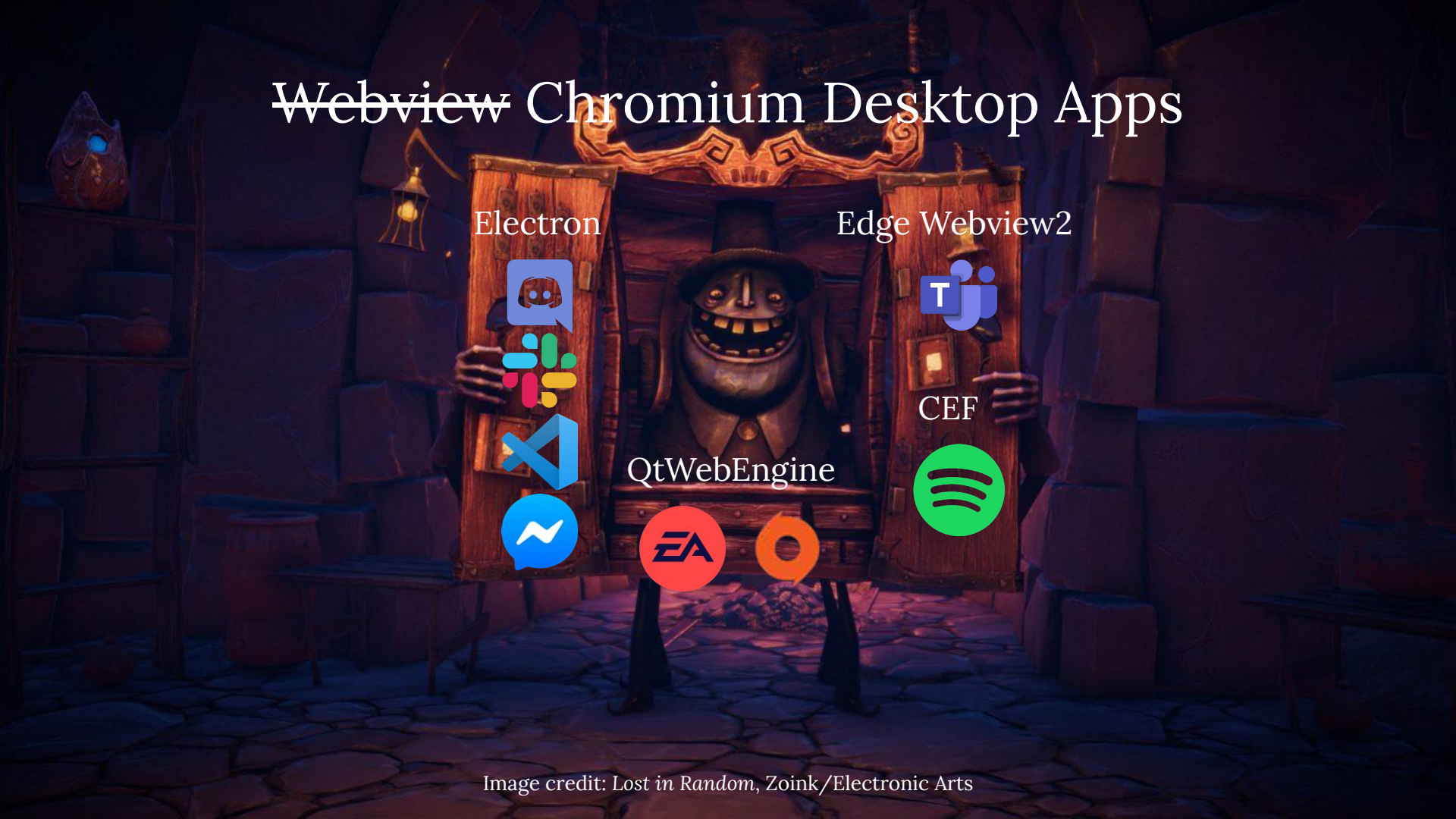
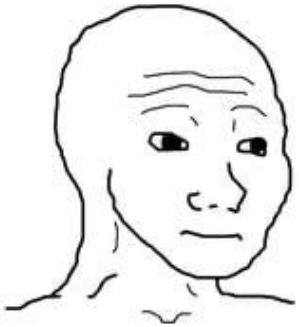
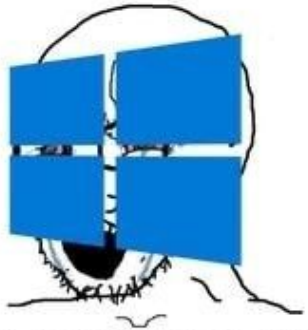


Image credit: *Lost in Random*, Zoink/Electronic Arts

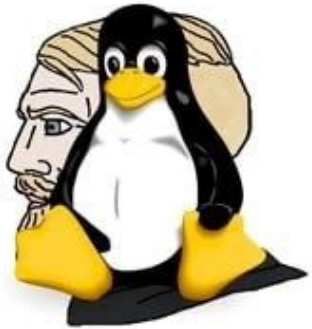


hey can i uninstall edge



NOOO!!! YOUR SYSTEM WILL BREAK

im going to uninstall the bootloader



go ahead lol

What to Look For?

Normal webapp client-side and server-side tests.

XSS -> RCE is a meme but mostly true.

As easy as

<img src=x onerror=document.location="file:///C:/windows/system32/calc.exe">

Special Webview App Things:

Protocol Handlers. E.g., slack://blah.

Local servers. E.g., RCE in PlayStation Now and RCE in VS Code WSL Extension.

The JavaScript Bridge or what can you call after an XSS?

Client-side Storage

Episode 4: Automation

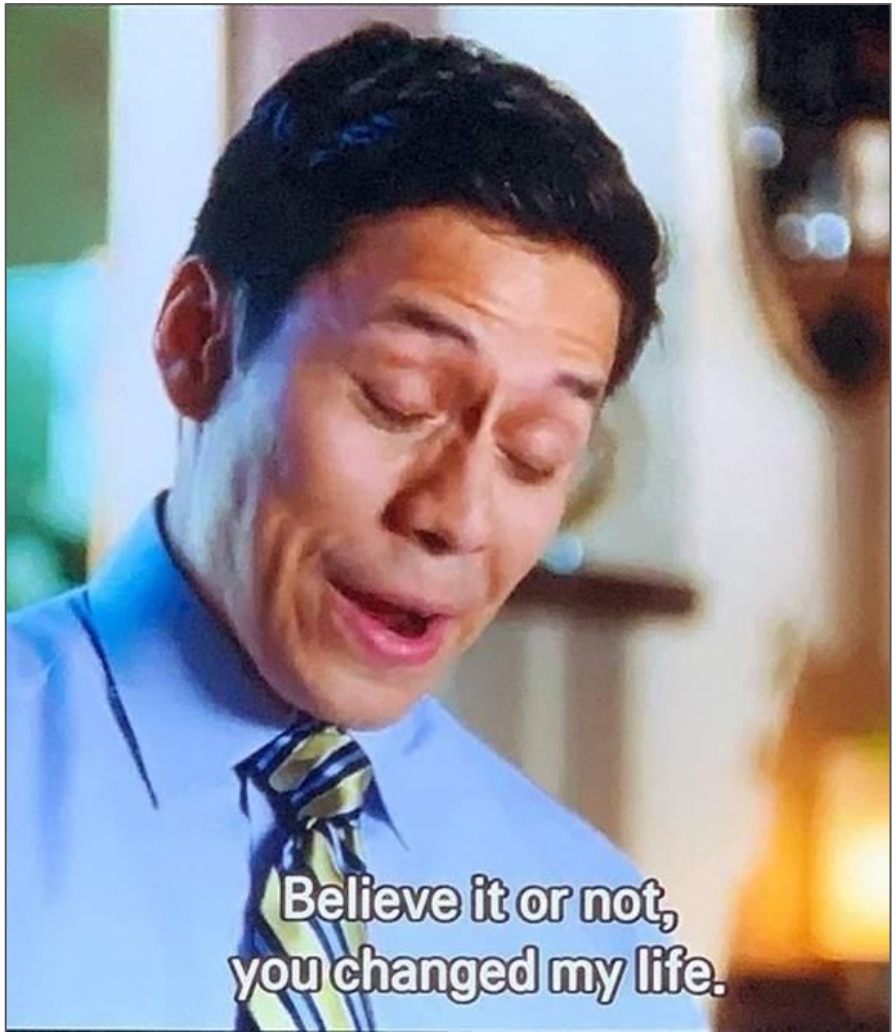
Image credit: *Lost in Random*, Zoink/Electronic Arts

Manual Work is a Bug

A.B.A: always be automating

1. Documenting the steps.
2. Create automation equivalents.
3. Create automation.
4. Self-service and autonomous systems.

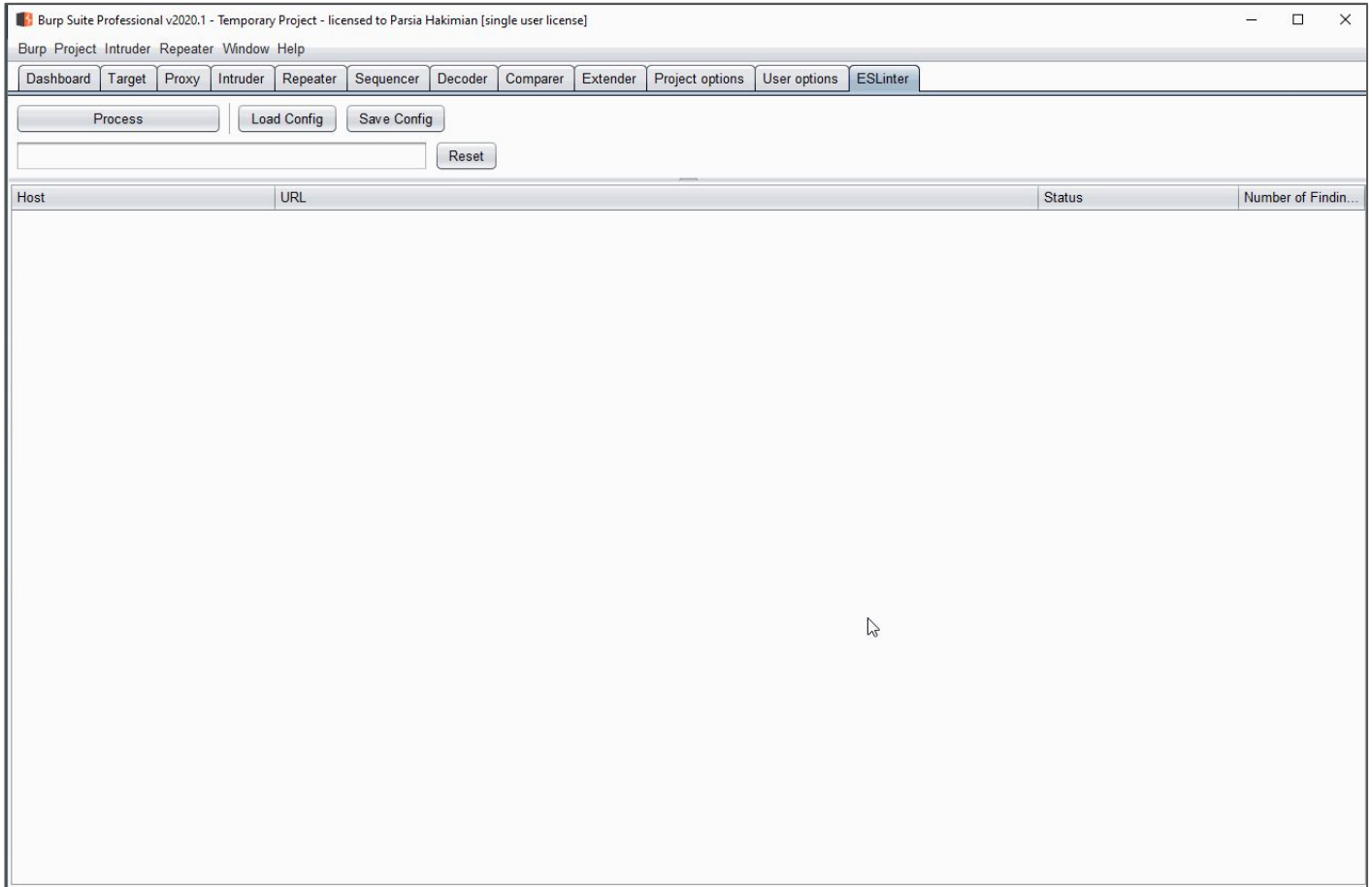
https://queue.acm.org/detail.cfm?id=3197520



Literature Review

Academic Papers Industry Netflix (lol)

Some Early Automation: ESLinter



Semgrep Usage

What rules are used?

Where does the output appear?

How is Semgrep executed?

Image credit: *Lost in Random*, Zoink/Electronic Arts

For Security Engineers - How?

How do you run it?

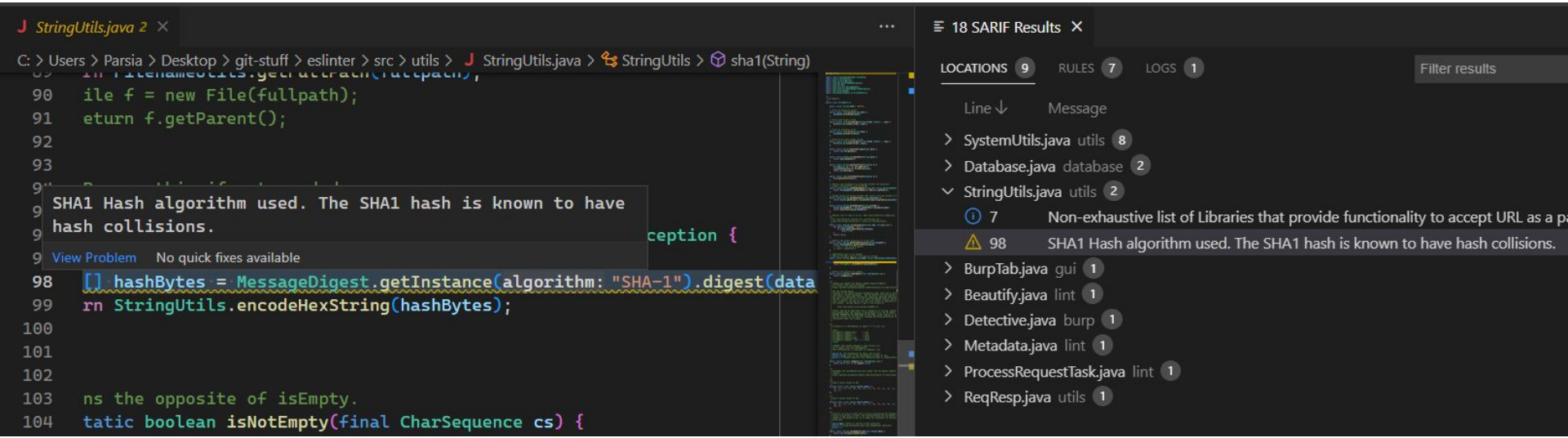
It can be manual.

$ semgrep --config /myrules/ --sarif --output myresults.sarif /path/to/code

For Security Engineers - Output

Where is the output stored?

Use a SARIF viewer.



For Security Engineers - Rules

Image credit: *Lost in Random*, Zoink/Electronic Arts

Detecting XXE in Java

Let's find every Java XML parser!

parsiya: blog-2022-03-java-xxe

SEMGREP RULE

Simple Advanced

```yaml
rules:
- id: blog-2022-03-java-xxe
  patterns:
    - pattern-either:
      - pattern: import DocumentBuilderFactory;
      - pattern: import XMLInputFactory;
      - pattern: import DOMParser;
      - pattern: import TransformerFactory;
      - pattern: import Validator;
      - pattern: import SchemaFactory;
```

TEST CODE

```
1  import javax.xml.parsers.DocumentBuilderFactory;
2  import javax.xml.stream.XMLInputFactory;
3  import oracle.xml.parser.v2.DOMParser;
4  import javax.xml.transform.TransformerFactory;
5  import javax.xml.validation.Validator;
6  import javax.xml.validation.SchemaFactory;
7  import javax.xml.transform.sax.SAXTransformerFactory;
8  import org.xml.sax.XMLReader;
9  import org.dom4j.io.SAXReader;
10 import org.jdom2.input.SAXBuilder;
11 import javax.xml.bind.Unmarshaller;
12 import javax.xml.xpath.XPathExpression;
13 import java.beans.XMLDecoder;
14
15 import javax.whatever.Whatever;
```

Keywords in Function Names - Encode/Decode

encode-decode-in-function-name

simple mode | advanced mode

```yaml
rules:
  - id: encode-decode-in-function-name
    patterns:
      - pattern: |
          $FUNC(...);
      - metavariable-regex:
          metavariable: $FUNC
          regex: .*(encode|decode).*
    message: Encoding/decoding function, worth a review.
    languages:
      - cpp
    severity: WARNING

```

test code | metadata | docs

```cpp
int main() {

    // ok:encode-decode-in-function-name
    randomfunc();

    // ruleid:encode-decode-in-function-name
    encodeme();

    int param = 1;
    // ruleid:encode-decode-in-function-name
    int res = decodeme(param);

    // ruleid:encode-decode-in-function-name
    thisencodefunctionisbad();

    // ok:encode-decode-in-function-name
    int res2 = safefunction();

    return 0;
}
```

And there's more!

Other tickets - incoming vulnerability reports

Find the root cause in source code and write rules.

Crypto(graphic) primitives - MD5

MD5

Memory-safe language - Unsafe

Hardcoded secrets

C++ - https://github.com/parsiya/semgrep-hotspots

sizeof(pointer*)

Snprintf

For Developers

Image credit: *Lost in Random*, Zoink/Electronic Arts

For Developers - Output

Parse the SARIF results and do whatever.

- https://github.com/microsoft/sarif-tools
- https://github.com/parsiya/sarif-info

We explored some options like

- Automatic tickets in Jira or … .
- Slack notifications.
- Pull/Merge request comments.

For Developers - Rules

Ask them!

- What keep you up at night?
- Which section of the codebase is vulnerable?
- What kind of security issues are important to you?

Show them!

- How rules are written.
- How Semgrep works.

They will start writing their own rules.

For Developers - Rules

Don’t have noise! You will lose their trust!

Don’t have noise! You will lose their trust!

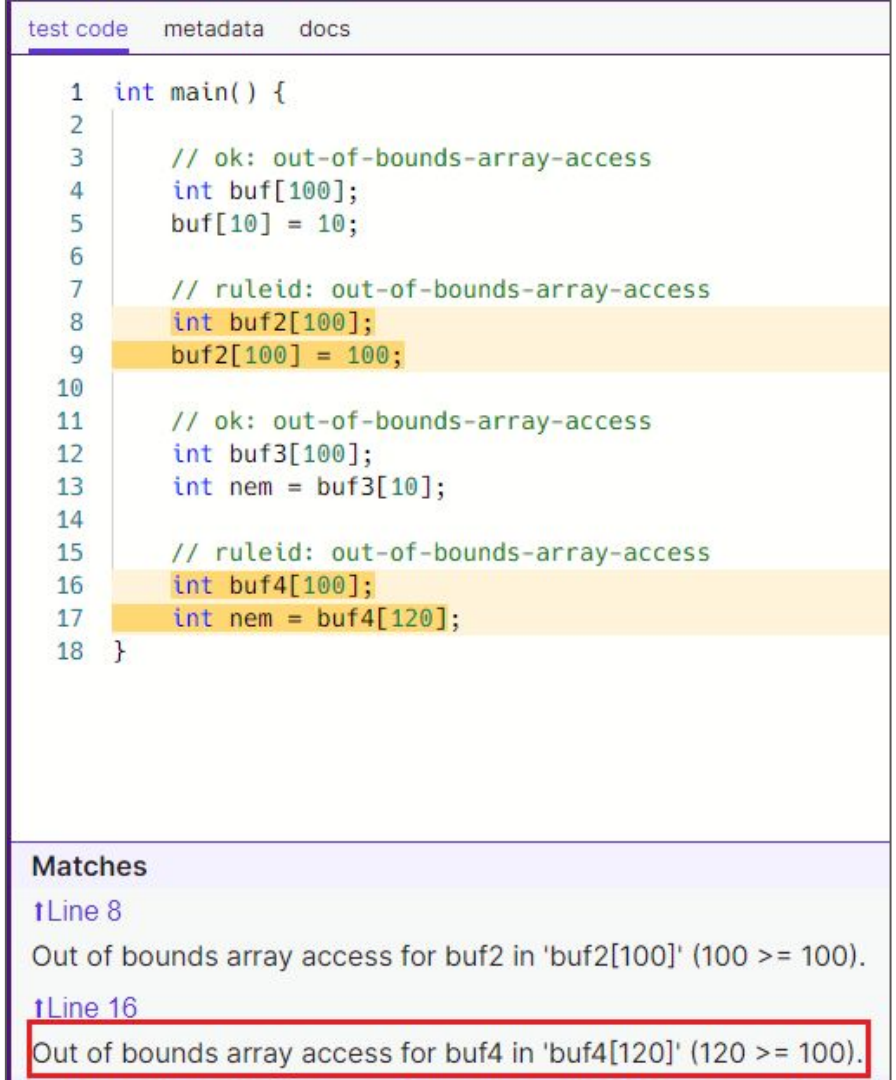
Don’t have noise! You will lose their trust!

Don’t have noise! You will lose their trust!

Don’t have noise! You will lose their trust!

Don’t have noise! You will lose their trust!

Don’t have noise! You will lose their trust!



Don’t Break the Build!

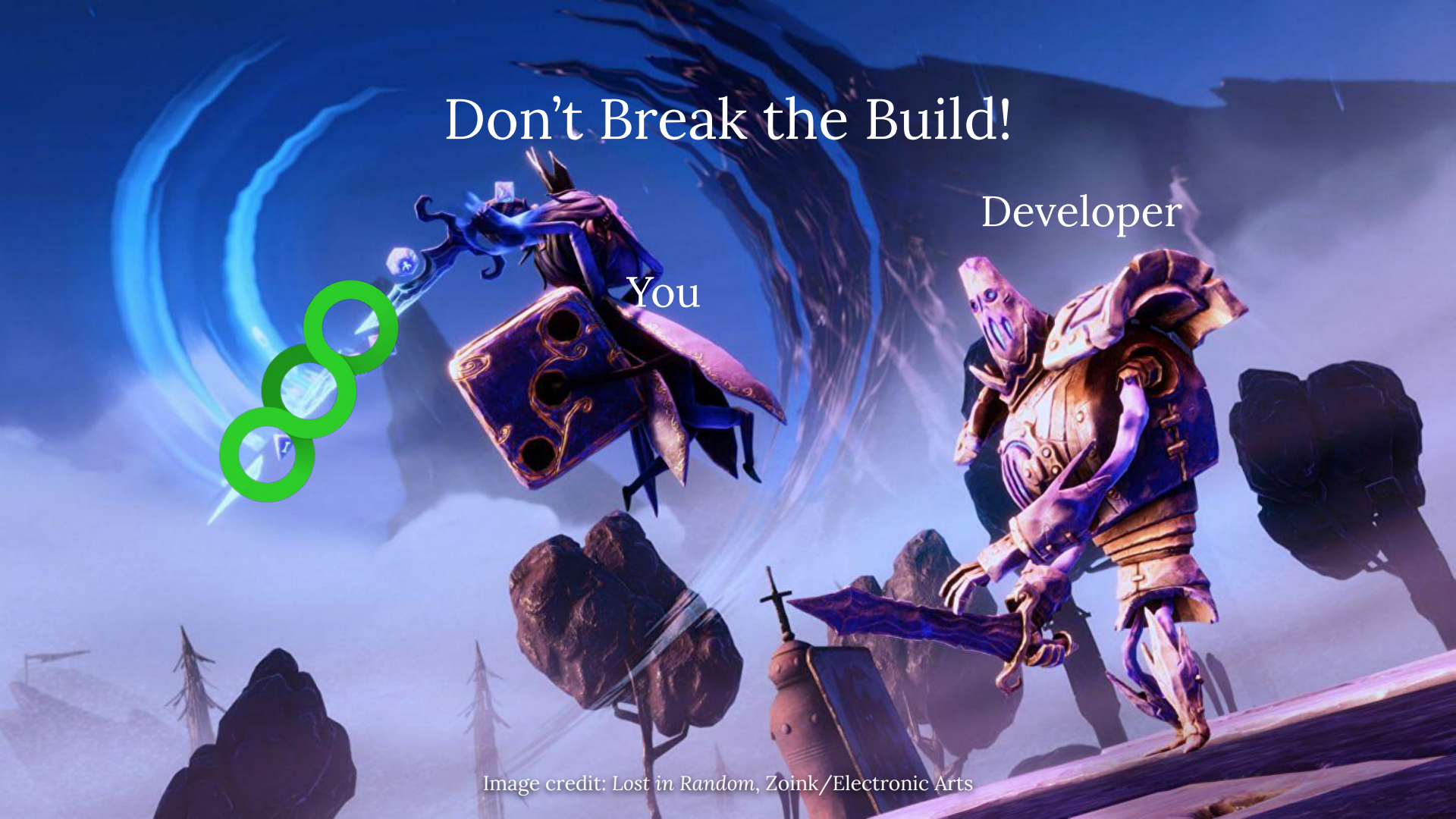


Image credit: *Lost in Random*, Zoink/Electronic Arts

Moral of the Story

Image credit: *Lost in Random*, Zoink/Electronic Arts

What Did We Learn Here Today?

Semgrep is free and fun. Please use it!

Write your own rules.

- Keep the noisy rules for yourself.
- Send accurate rules to the developers.

Try to make the process as seamless as possible for them.

Encourage the developers to experiment with Semgrep and write rules.

Bonus: Where Do I start?

Static Program Analysis: https://cs.au.dk/~amoeller/spa/

Learn Semgrep: https://semgrep.dev/learn

Protocol Handlers:

- https://parsiya.net/blog/2021-03-17-attack-surface-analysis-part-2-custom-protocol-handlers/

Local Servers:

- localghost: Escaping the Browser Sandbox without 0-Days
- RCE in PlayStation Now
- RCE in VS Code WSL Extension

The JavaScript Bridge in Modern Desktop Applications

Bonus: Where Do I start? Continued

Manual Work is a Bug

https://queue.acm.org/detail.cfm?id=3197520

ESLinter: Manual JavaScript Linting is a Bug

https://github.com/parsiya/eslinter

Semgrep in the CI/CD Pipeline

GitHub Action
GitLab SAST
OWASP secureCodeBox

Semgrep Hot Spots

https://parsiya.net/blog/semgrep-hotspot/
https://github.com/parsiya/semgrep-hotspots

Bonus: Where Do I start? Continued

Semgrep Deployment

- Building a SAST program at Razorpay's scale
- A Guide On Implementing An Effective SAST Workflow

Questions

twitter.com/CryptoGangsta

parsiya.net

github.com/parsiya

Image credit: *Lost in Random*, Zoink/Electronic Arts