German OWASP Day 2025

YuraScanner

Leveraging LLMs for Task-driven Web App Scanning

Aleksei Stafeev, Tim Recktenwald, Gianluca De Stefano, Soheil Khodayari, Giancarlo Pellegrino

Coverage

More coverage
=
More opportunities to find vulnerabilities

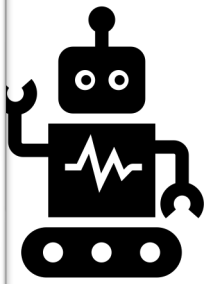
Web Security Scanners

Main objective

Automatically find
target website

Tools:

- **Click** *some* butt
- **Type** *some* lette
- *Beep* occasiona

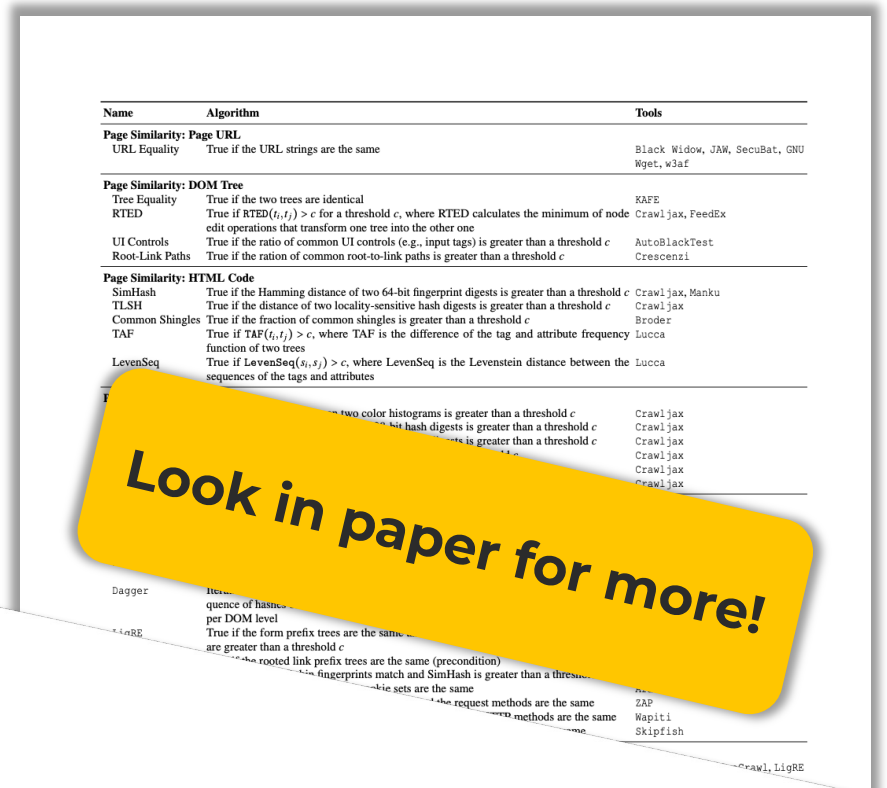


Research on Crawling

- **Page Similarity**
  - URL
  - DOM Tree
  - HTML (e.g., Hash-based)
  - Screenshots
  - Cust
  - ...sed: e.g., BFS or DFS
  - Randomized
  - Reinforcement Learning
  - Code coverage guided
  - Rule-based

Randomized won!

Look in paper for more!



Name	Algorithm	Tools
Page Similarity: Page URL		
URL Equality	True if the URL strings are the same	Black Widow, JAW, SecuBat, GNU Wget, w3af
Page Similarity: DOM Tree		
Tree Equality	True if the two trees are identical	KAFE
RTED	True if $\texttt{RTED}(t_i, t_j) > c$ for a threshold c, where RTED calculates the minimum of node edit operations that transform one tree into the other one	Crawljax, FeedEx
UI Controls	True if the ratio of common UI controls (e.g., input tags) is greater than a threshold c	AutoBlackTest
Root-Link Paths	True if the ration of common root-to-link paths is greater than a threshold c	Crescenzi
Page Similarity: HTML Code		
SimHash	True if the Hamming distance of two 64-bit fingerprint digests is greater than a threshold c	Crawljax, Manku
TLSH	True if the distance of two locality-sensitive hash digests is greater than a threshold c	Crawljax
Common Shingles	True if the fraction of common shingles is greater than a threshold c	Broder
TAF	True if $\texttt{TAF}(t_i, t_j) > c$, where TAF is the difference of the tag and attribute frequency function of two trees	Lucca
LevenSeq	True if $\texttt{LevenSeq}(s_i, s_j) > c$, where LevenSeq is the Levenstein distance between the sequences of the tags and attributes	Lucca
	...two color histograms is greater than a threshold c	Crawljax
	...bit hash digests is greater than a threshold c	Crawljax
	...ts is greater than a threshold c	Crawljax
		Crawljax
		Crawljax
		Crawljax
Dagger	...quence of hashes per DOM level	
LigRE	True if the form prefix trees are the same ... are greater than a threshold c	
	...the rooted link prefix trees are the same (precondition)	
	...fingerprints match and SimHash is greater than a thresh...	
	...kie sets are the same	
	...the request methods are the same	ZAP
	...TP methods are the same	Wapiti
		Skipfish
		...Crawl, LigRE

SoK: State of the Krawlers – Evaluating the Effectiveness of Crawling Algorithms for Web Security Measurements

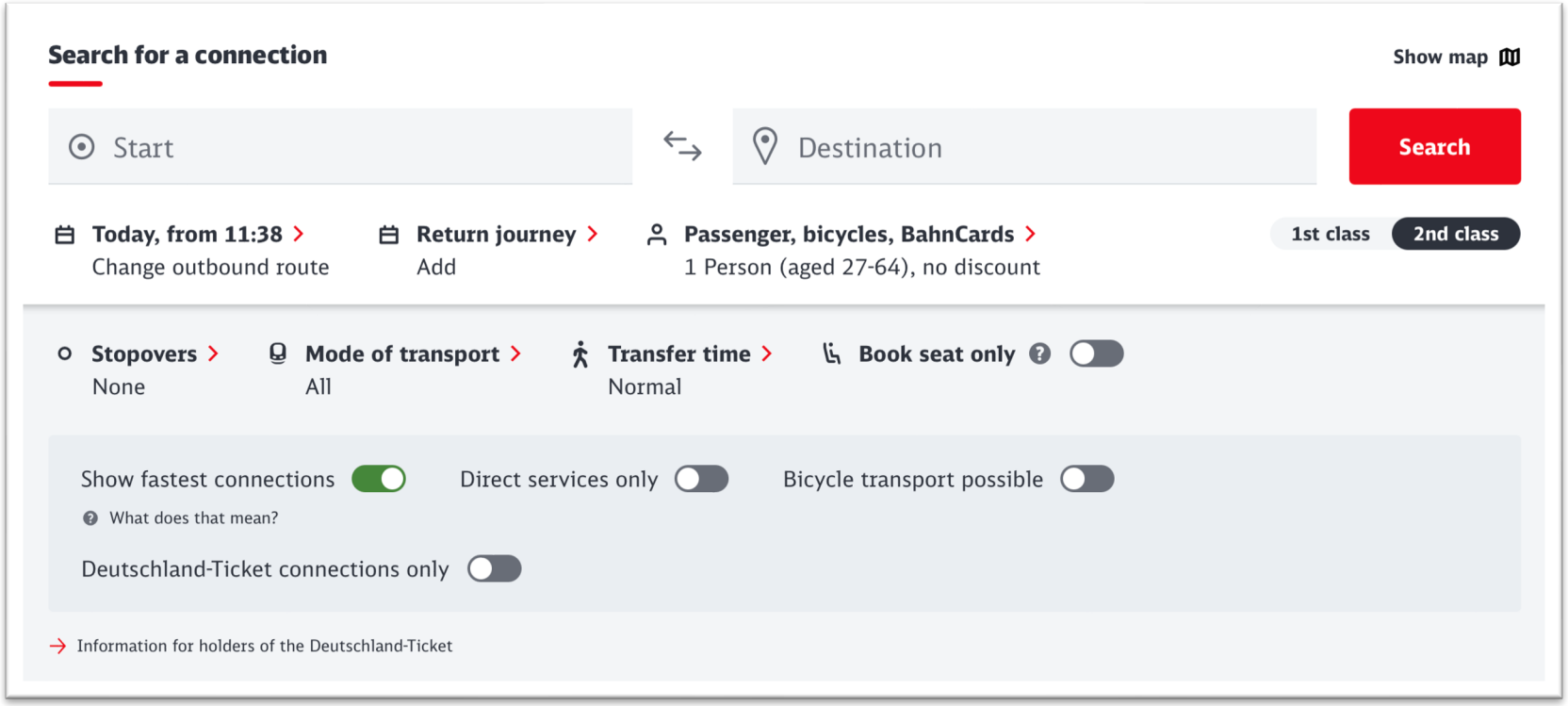
Aleksei Stafeev
CISPA Helmholtz Center for Information Security

Giancarlo Pellegrino
CISPA Helmholtz Center for Information Security

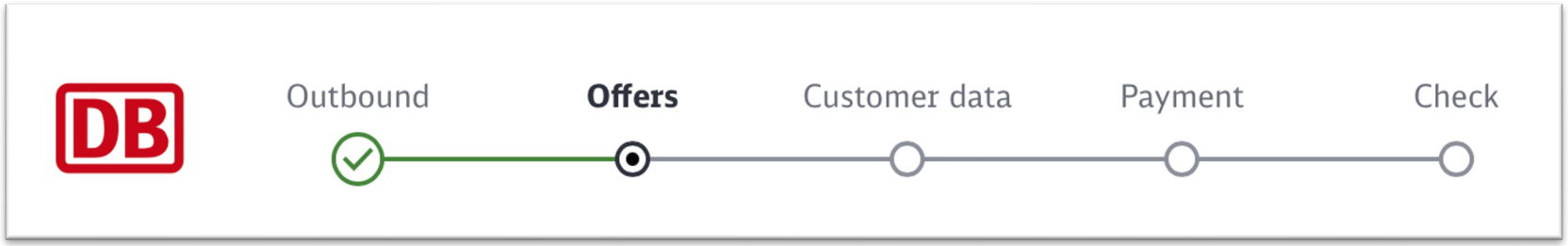
Abstract

Web crawlers are tools widely used in ...surements whose performa... studied so far. I...

Motivation


Search for a connection
Show map
Start
Destination
Search
Today, from 11:38
Change outbound route
Return journey
Add
Passenger, bicycles, BahnCards
1 Person (aged 27-64), no discount
1st class
2nd class
Stopovers
None
Mode of transport
All
Transfer time
Normal
Book seat only
Show fastest connections
Direct services only
Bicycle transport possible
What does that mean?
Deutschland-Ticket connections only
Information for holders of the Deutschland-Ticket

Motivation



Multi-step workflow

Motivation

- Traditional scanners struggle with exploring deeper states
- **Key limitation:** They lack awareness of multi-step workflows
- ML-based methods have been proposed to tackle this weakness
  - E.g., reinforcement learning on user-provided traces [1]
- Does not scale well!

[1] E. Z. Liu, K. Guu, P. Pasupat, T. Shi, and P. Liang, "Reinforcement Learning on Web Interfaces using Workflow-Guided Exploration," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Approach

- Instead of training a model, we opted to use large language models (LLMs)
- Non-academic approaches have proposed LLM-based browsing agents to assist users with tasks [2, 3]
  - E.g., “Book a hotel in Düsseldorf”
- Instead, we want to complete workflows and reach deeper states in web applications without user interaction

[2] N. Friedman. (2022) Natbot. https://github.com/nat/natbot.
[3] (2024) Skyvern. https://github.com/Skyvern-AI/skyvern.

You are a scanner...

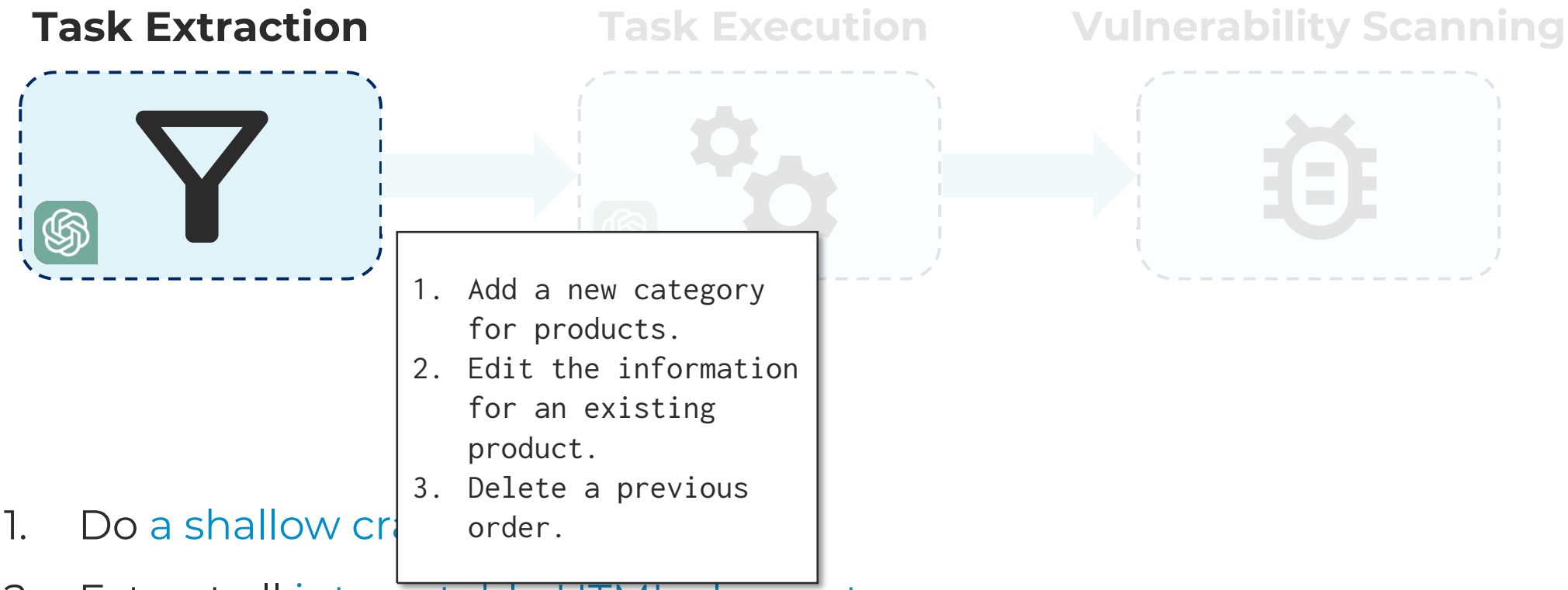
Yu r a scanner...

YuraScanner

YuraScanner

A fully automated, task-driven web application scanner

Architecture of YuraScanner

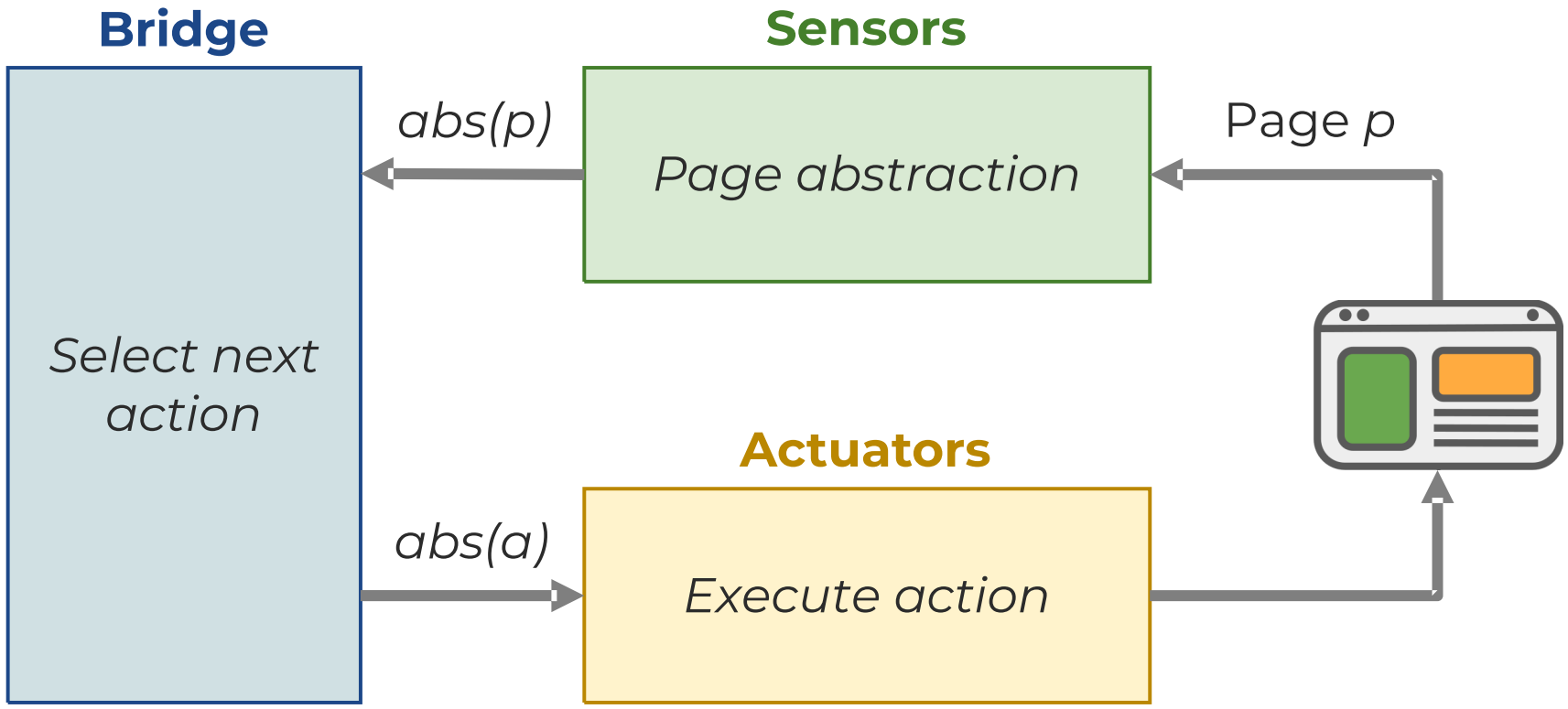


1. Do a shallow cr
2. Extract all interactable HTML elements
3. Provide a list of elements to a LLM and request a list of tasks

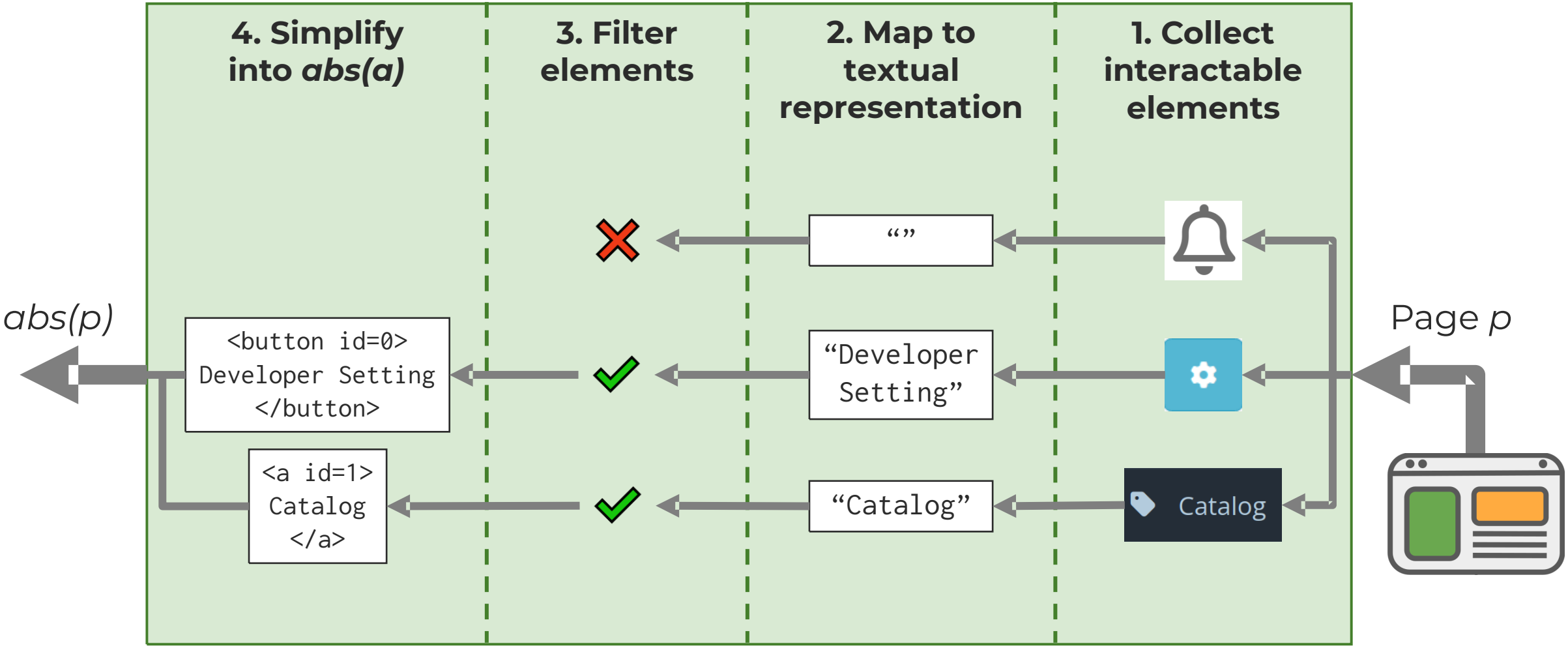
Architecture of YuraScanner

- Executes every task iteratively
- At each step of a task:
  - Generates a simplified *textual* page representation
  - Queries LLM for the next command (e.g., *click button #2*)
  - Executes the command

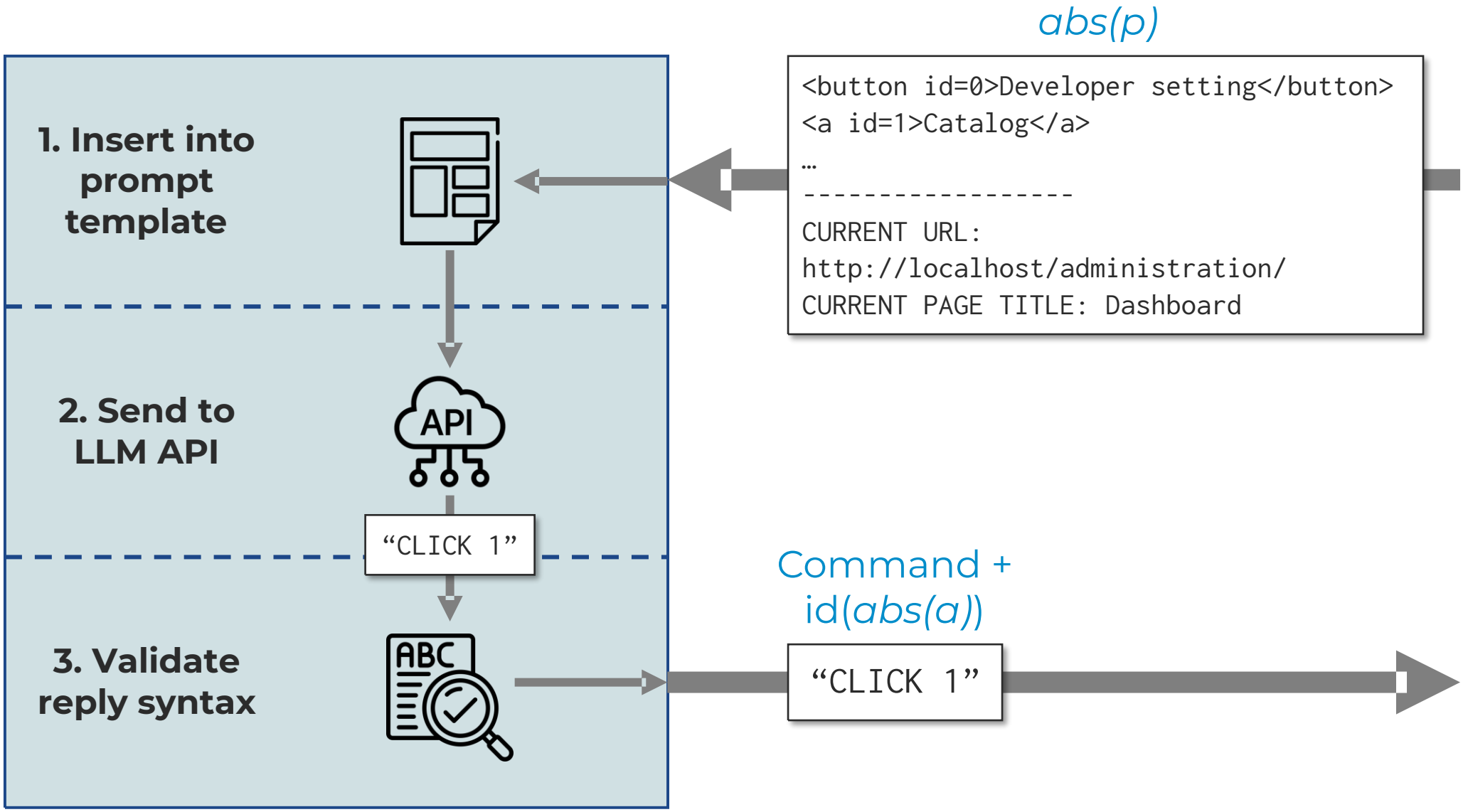
Task-driven Crawling



Sensors

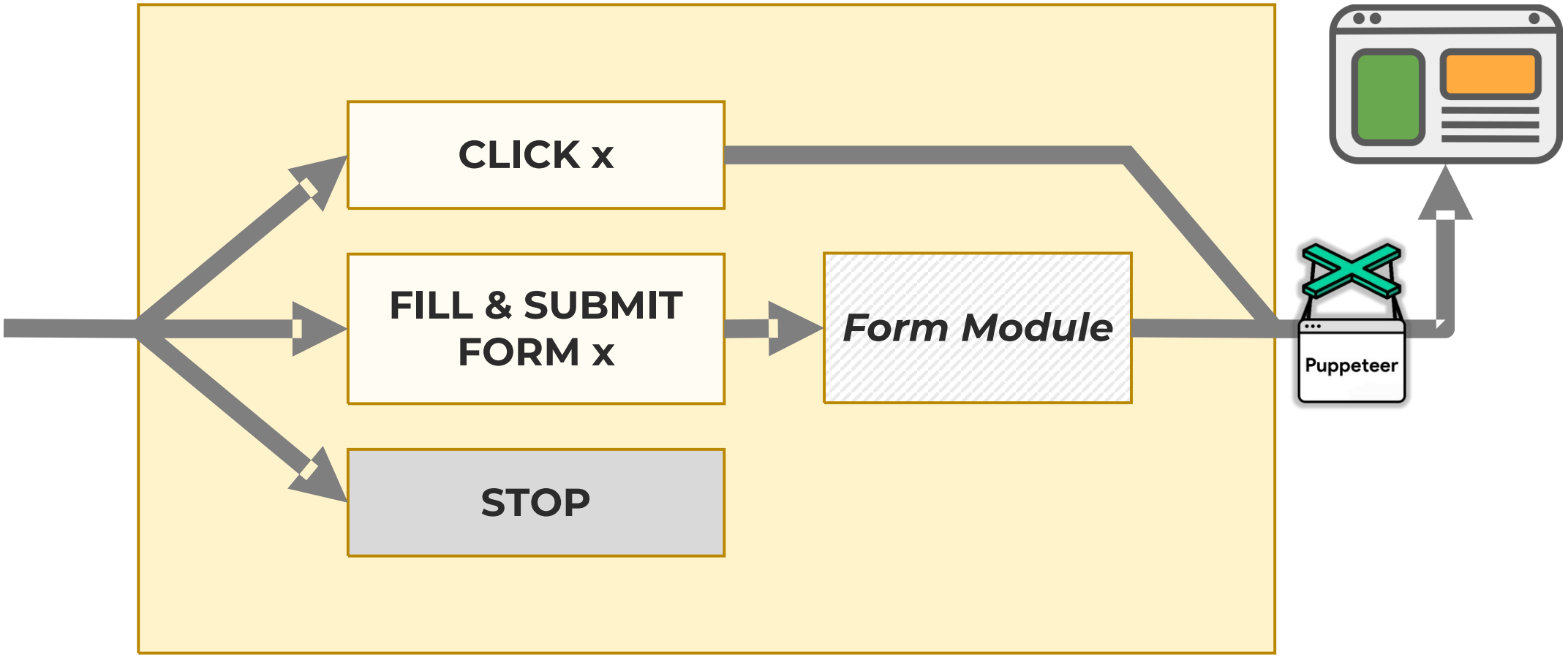

4. Simplify into abs(a)
3. Filter elements
2. Map to textual representation
1. Collect interactable elements
abs(p)
Page p
<button id=0> Developer Setting </button>
 Catalog
“”
“Developer Setting”
“Catalog”
Catalog

Bridge


abs(p)
<button id=0>Developer setting</button>
Catalog
…

CURRENT URL:
http://localhost/administration/
CURRENT PAGE TITLE: Dashboard
1. Insert into prompt template
2. Send to LLM API
API
“CLICK 1”
3. Validate reply syntax
ABC
Command + id(abs(a))
“CLICK 1”

Actuators



Architecture of YuraScanner

1. Visits every form collected during task execution
2. Uses the XSS detection engine of Black Widow [4]

[4] Benjamin Eriksson, Giancarlo Pellegrino, and Andrei Sabelfeld: "BlackWidow: Blackbox data-driven web scanning," in *2021 IEEE Symposium on Security and Privacy (SP)*

Evaluation

- We evaluated **YuraScanner** on 20 popular, modern web applications
- We divided our testbed into two sets:

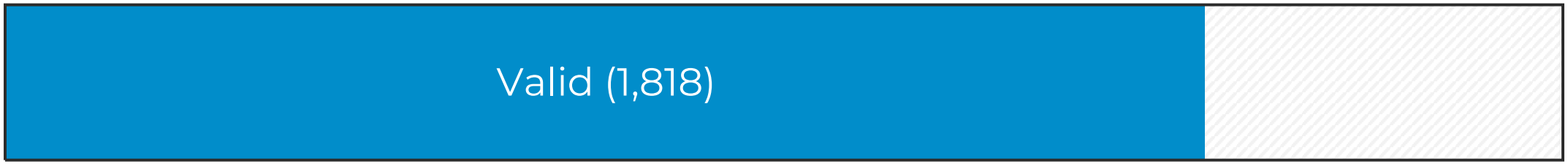
1. **Task Extraction and Execution**
   - Random subset of 10 web apps
   - **Manually label** valid tasks and their success rate during task execution
2. **Vulnerability Detection**
   - All 20 web apps
   - Inspect vulnerabilities found by the attack module

Evaluation Results: Task Extraction

2,361 tasks

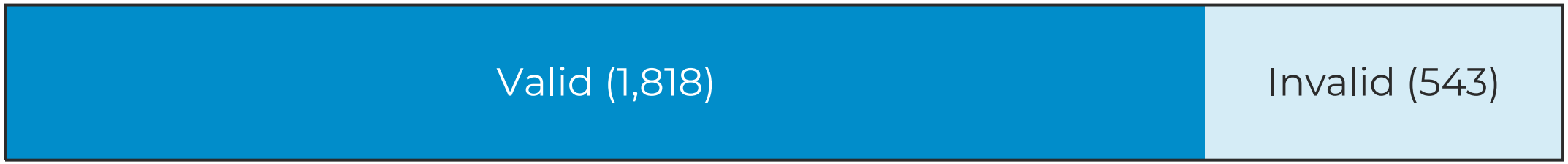
- 2,361 tasks were generated in total across 10 web applications

Evaluation Results: Task Extraction



- 2,361 tasks were generated in total across 10 web applications
- 77% of the tasks were valid (1,818 tasks)

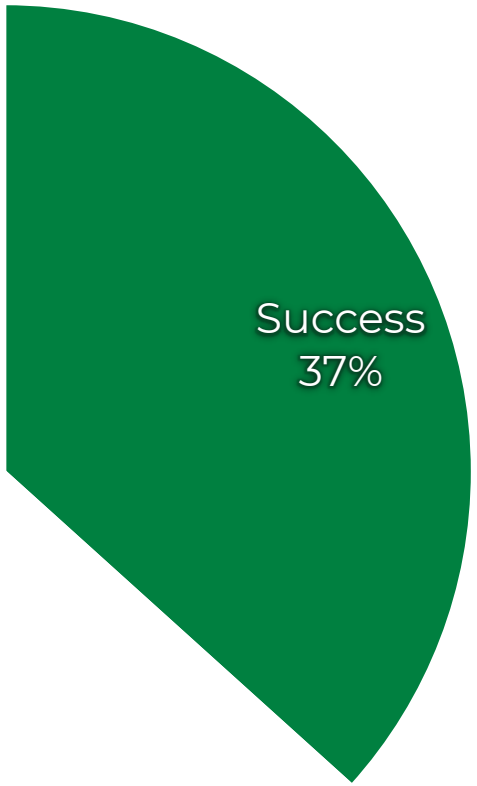
Evaluation Results: Task Extraction



- 2,361 tasks were generated in total across 10 web applications
- 77% of the tasks were valid (1,818 tasks)
- "Invalid" = Functionality does not exist in the web application
- Invalid task generation mainly occurred on pages with insufficient context
  - E.g., login page with only one button

MediaWiki *Get detailed information about membership or subscription plans*

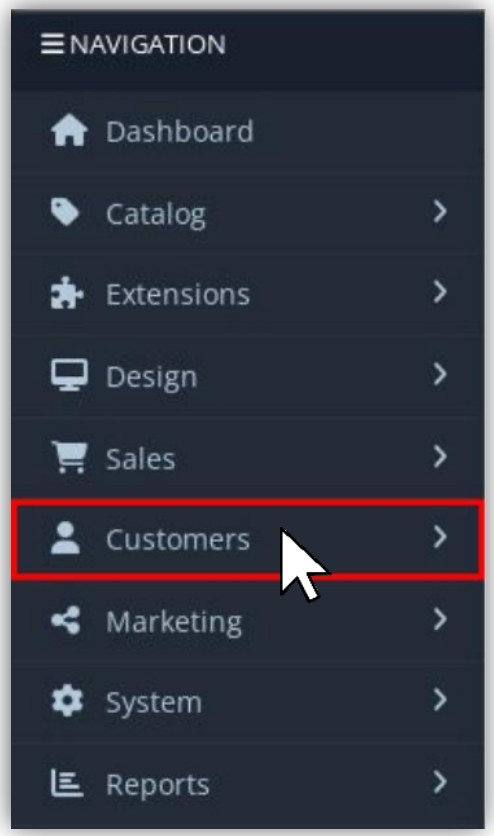
Evaluation Results: Task Execution



Task Execution Classification
(1,818 valid tasks)

Evaluation Results: Task Execution

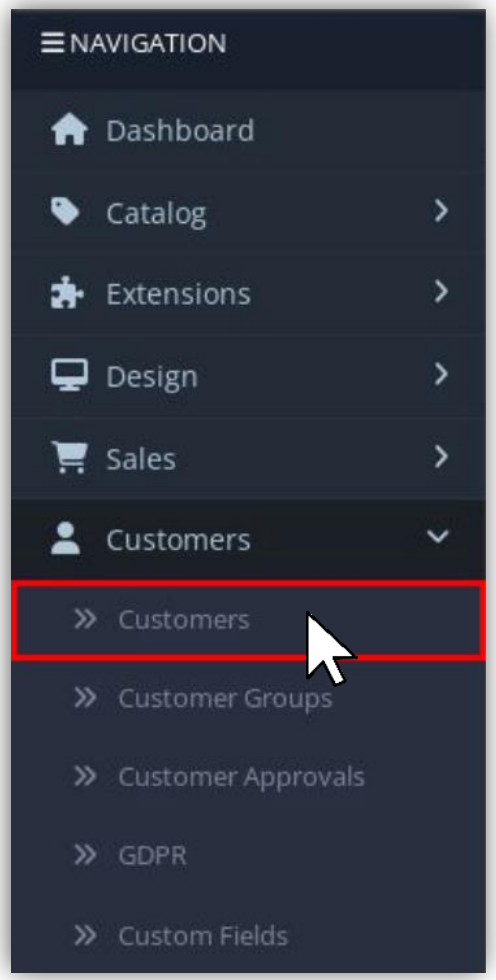
Task Execution Classification
(1,818 valid tasks)



Task: Add new “Customers” to the database

Evaluation Results: Task Execution

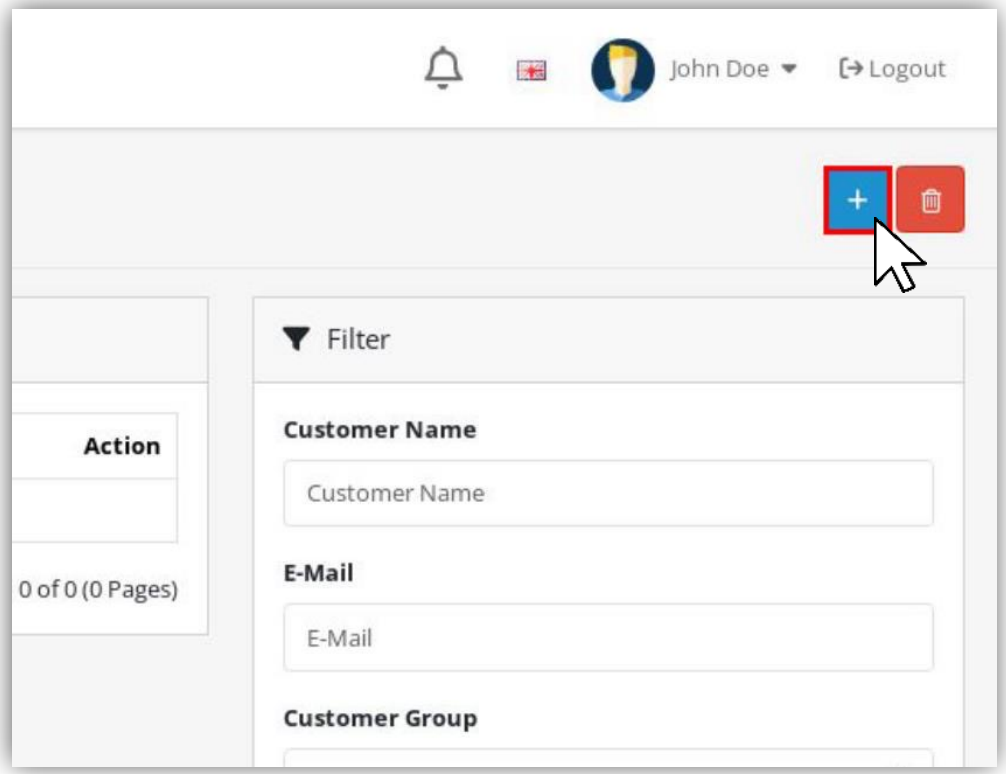
Task Execution Classification
(1,818 valid tasks)



Task: Add new “Customers” to the database

Evaluation Results: Task Execution

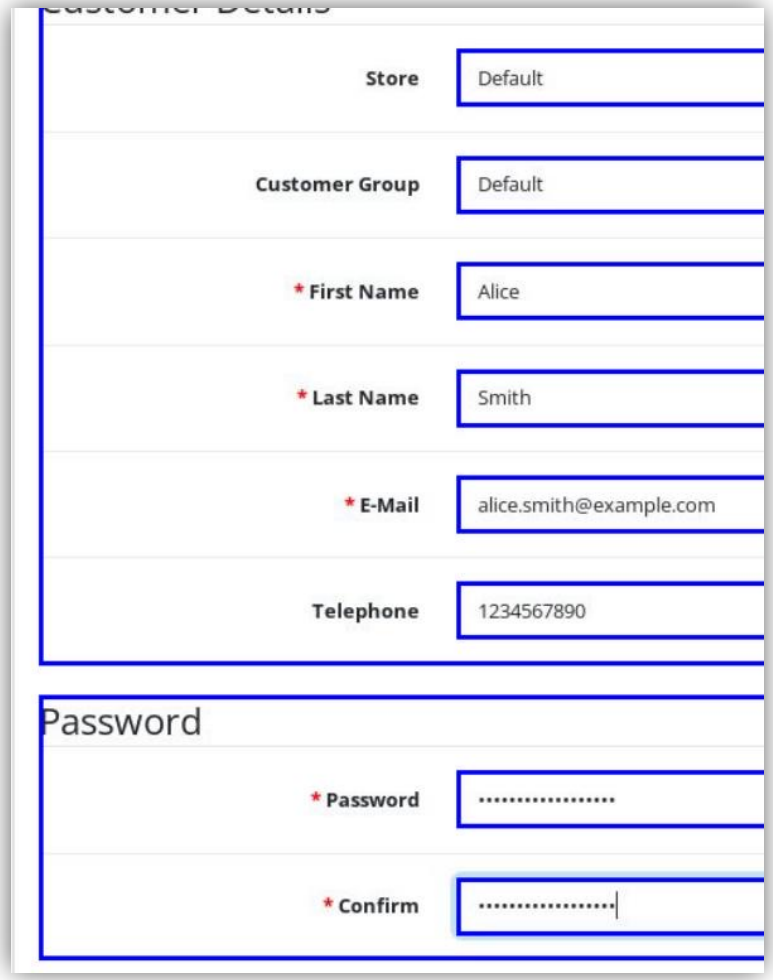
Task Execution Classification
(1,818 valid tasks)



Task: Add new “Customers” to the database

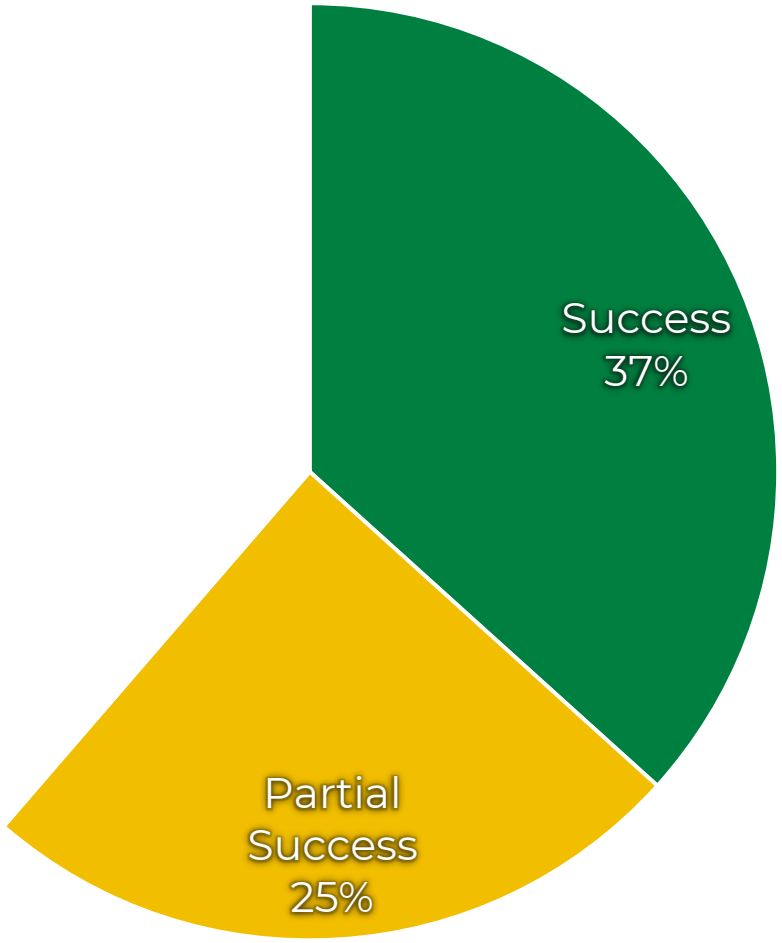
Evaluation Results: Task Execution

Task Execution Classification
(1,818 valid tasks)



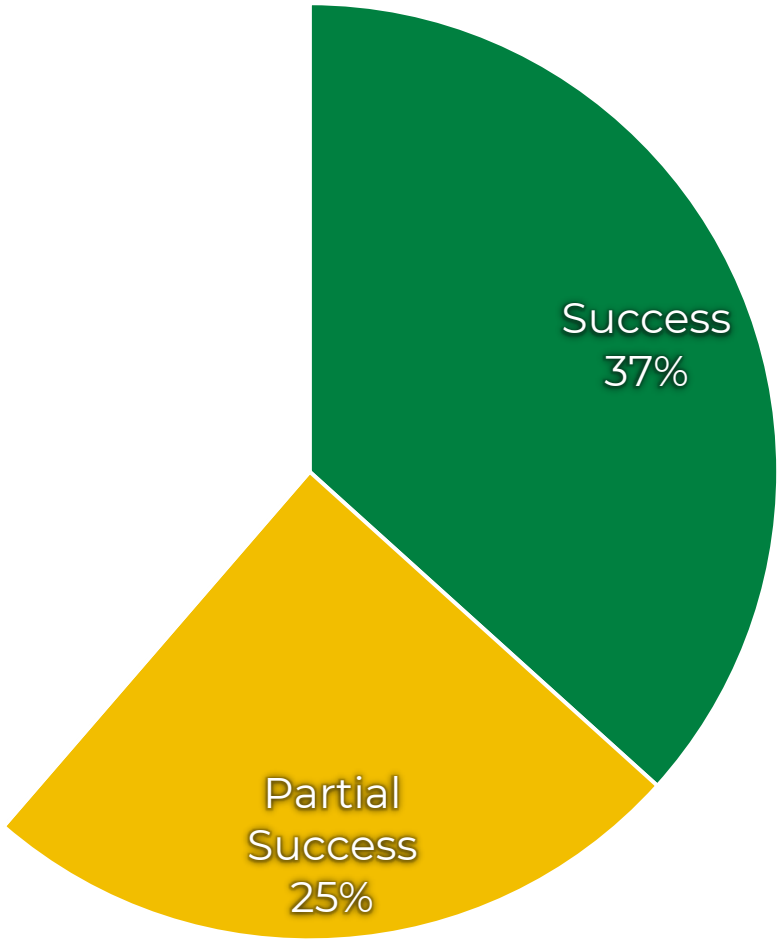
Task: Add new “Customers” to the database

Evaluation Results: Task Execution

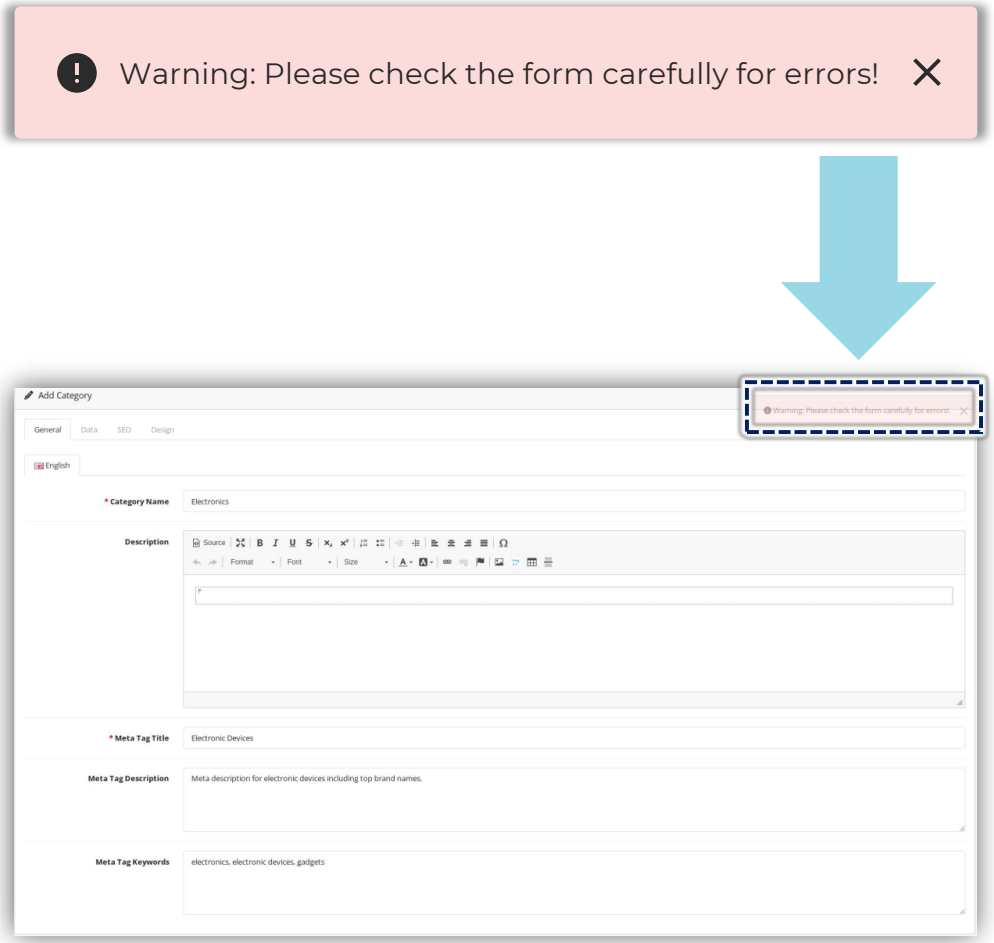


Task Execution Classification
(1,818 valid tasks)

Evaluation Results: Task Execution



Task Execution Classification
(1,818 valid tasks)



Evaluation Results: Task Execution

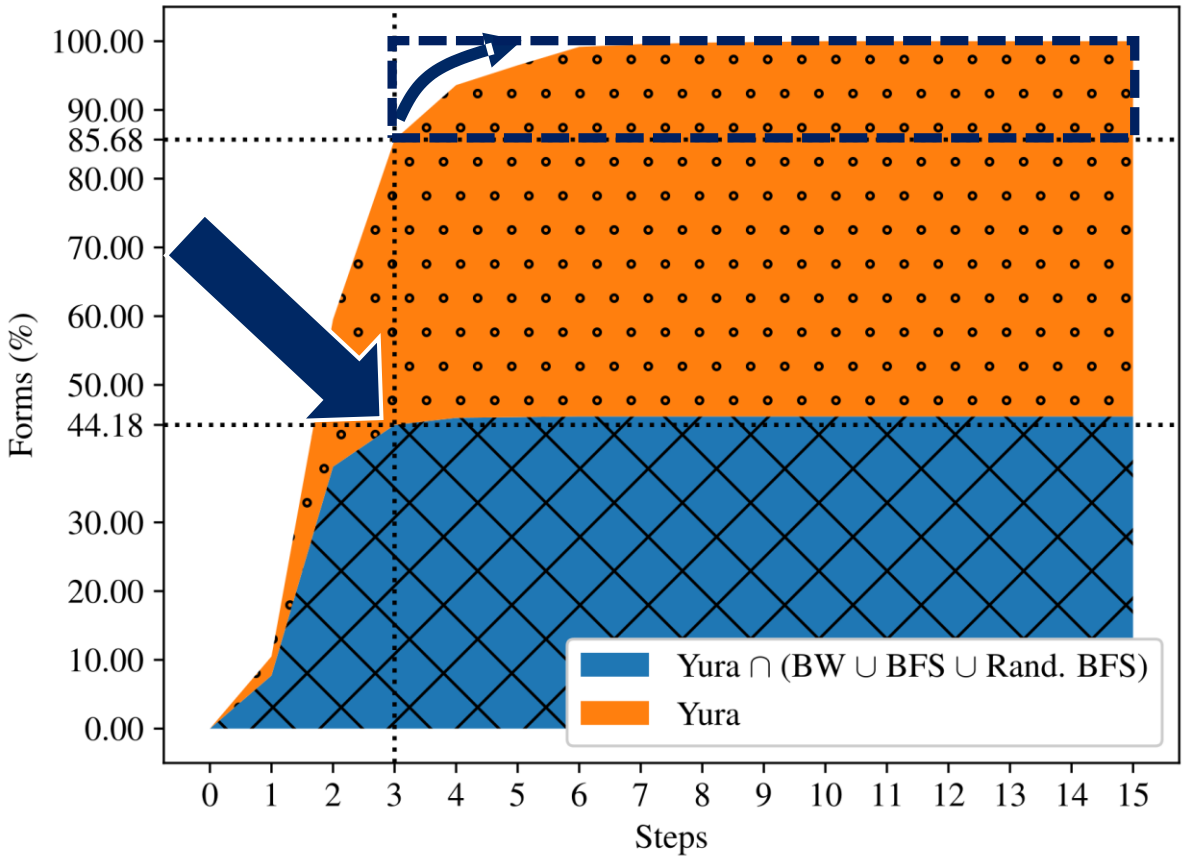
Task Execution Classification
(1,818 valid tasks)

- Missing State
  - E.g., tried to edit an object before creating one
- Deviated
  - Clicked on a wrong button
- Gave Up Immediately
  - Issued a STOP command

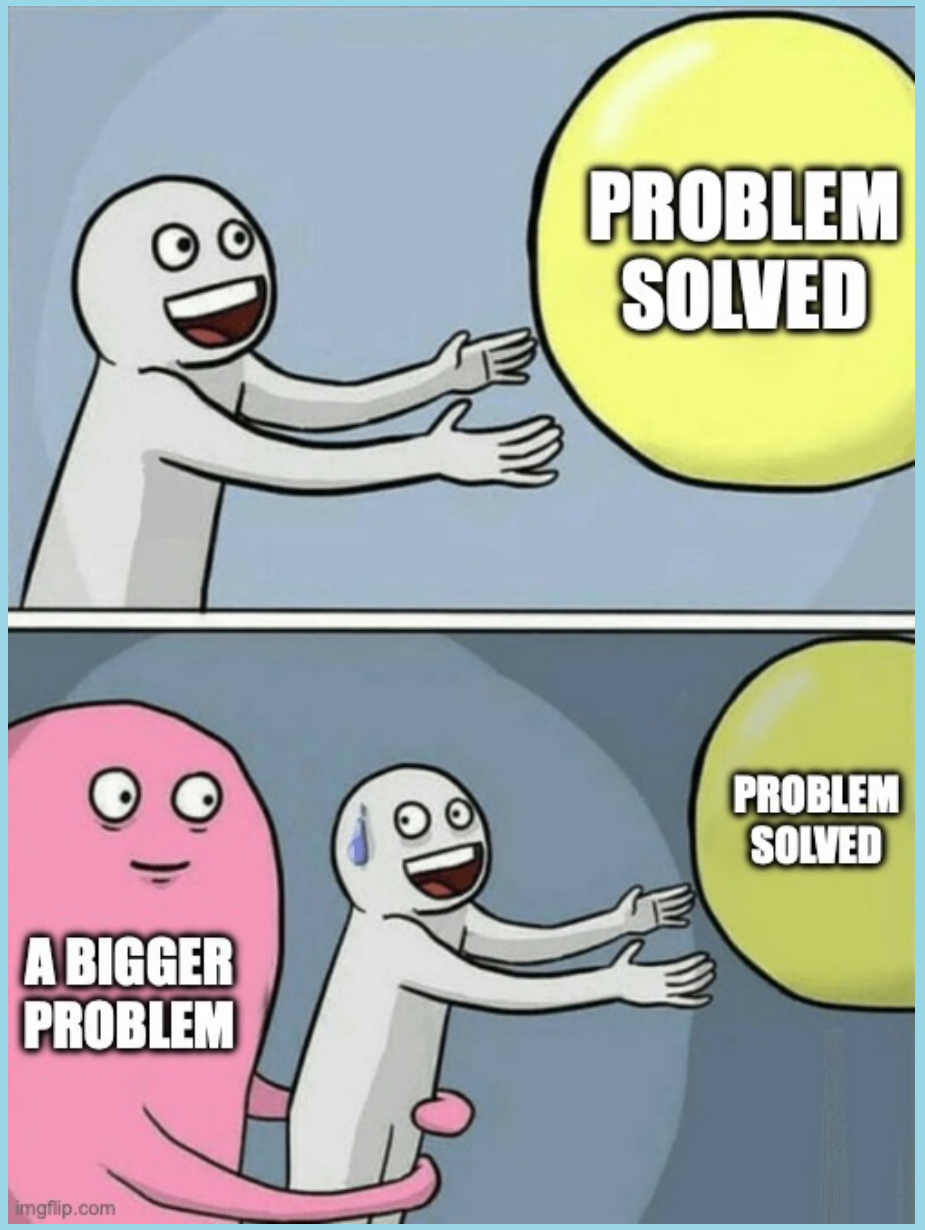
Characterization of the New Attack Surface

Is the new attack surface found by YuraScanner "*deeper*"?

- Comparison with BlackWidow, BFS, Random BFS
- 44% percent of the forms were discovered by all scanners
- The remaining forms were found **exclusively** by YuraScanner
  - Notably, 14.3% were found at **depth > 3**
  - **Out of reach** for traditional tools!



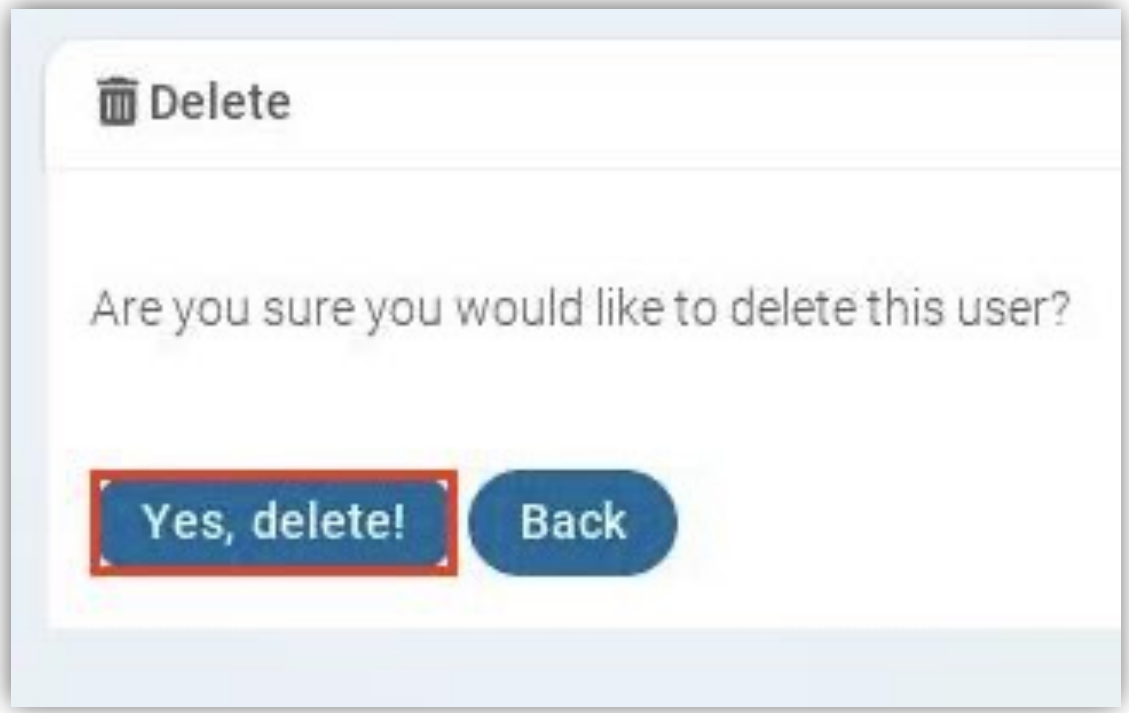
Suffering from Success



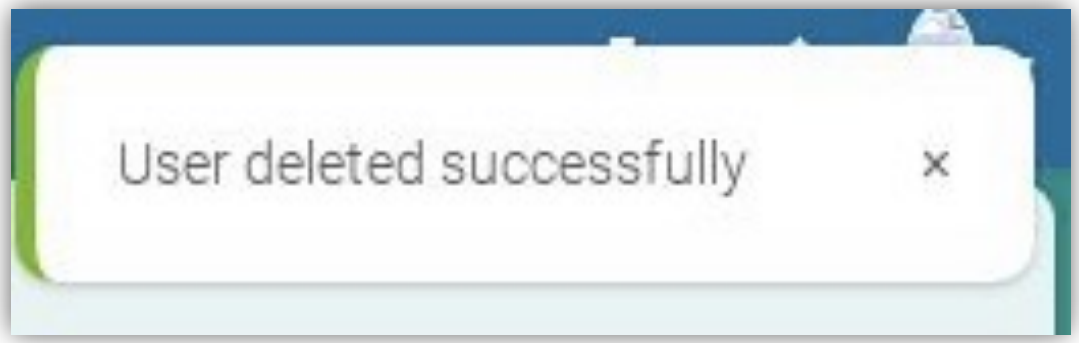
Suffering from Success

Task: “Delete a user from the ‘User Management’ section.”

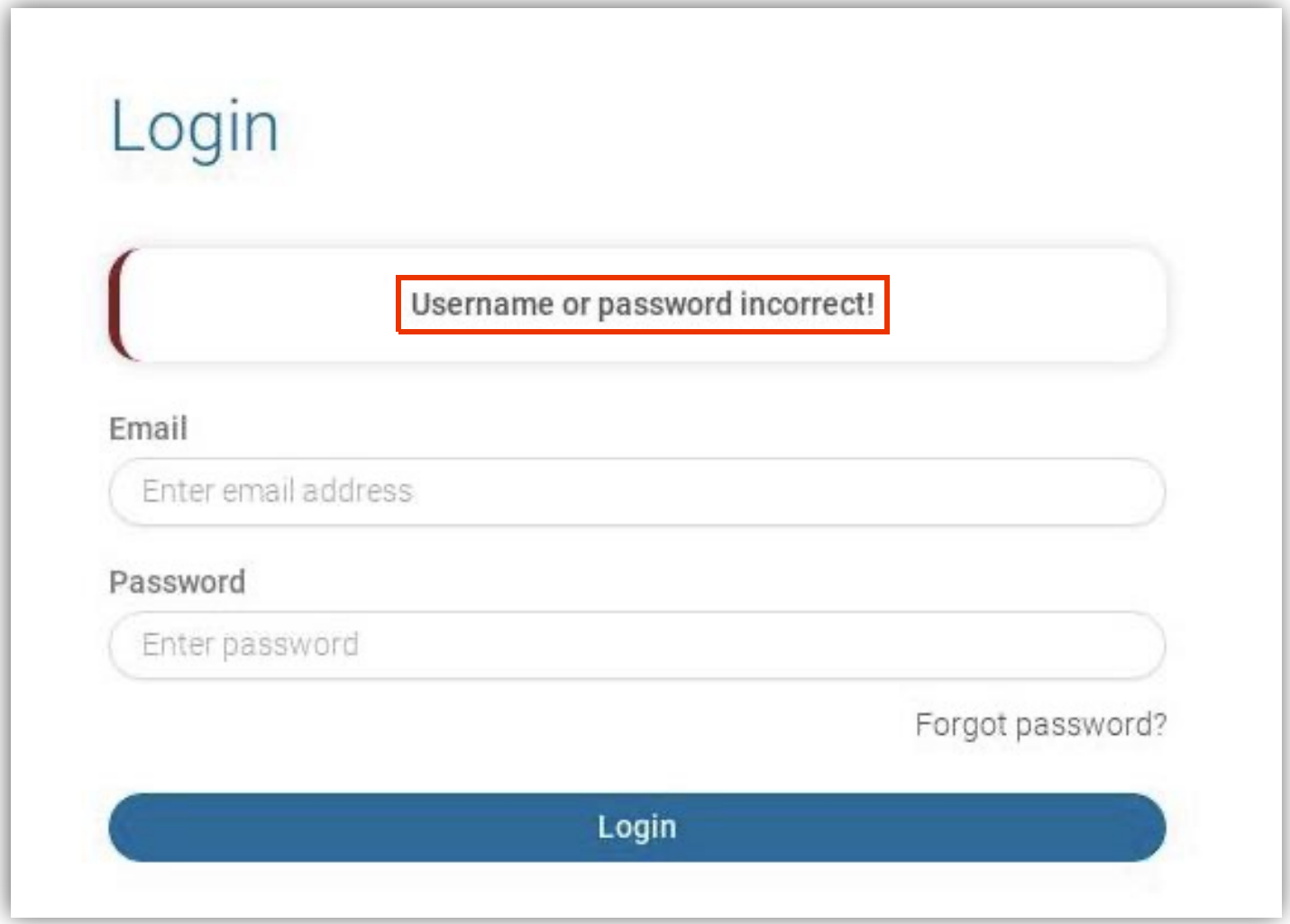
Suffering from Success



Suffering from Success



Suffering from Success



Directed by
YURASCANNER

Vulnerability Detection

App	Total	Unique	YuraScanner		Black Widow	
			Stored XSS	*Reflected XSS*	*Stored XSS*	*Reflected XSS*
Redacted	**12**	**11**	**4**	**7**	-	**1**
Moodle	**2**	**1**	**1**	-	**1**	-
Leantime	**1**	**1**	-	-	**1**	-

- 13 unique zero-day vulnerabilities discovered
- 12 of them found by **YuraScanner**
- Located between two to four clicks away from the front page

Takeaways

- Coverage is vital for the vulnerability detection
- Traditional scanners struggle with multi-step workflows
  - 👍 LLMs is a promising solution with 62% workflows executed fully or partially
- Task-driven crawling reaches **deeper functionalities** compared to traditional scanners
- LLMs may put the web application in an unrecoverable state
  - ⚠️ Need more safeguarding