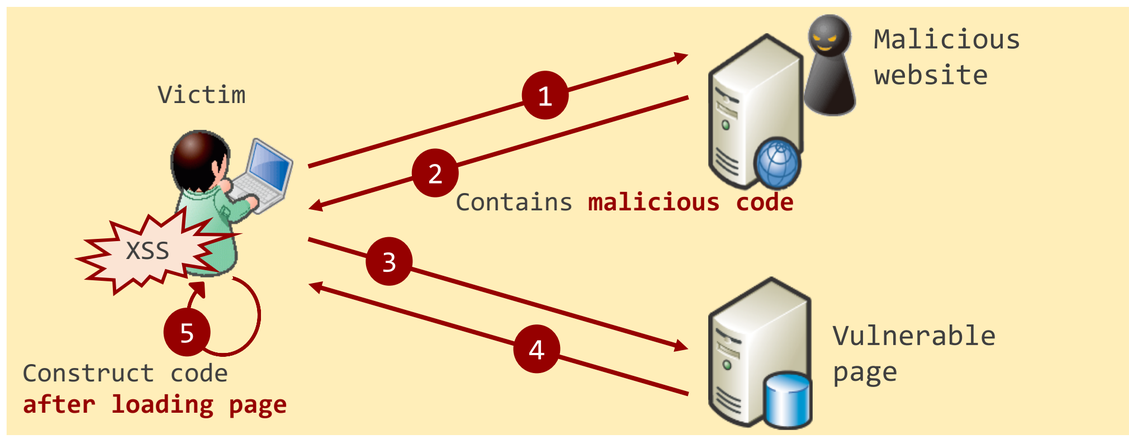
More Complete DOM XSS Detection with Dynamic Event Generation

Carnegie Mellon University CyLab Security and Privacy Institute

Authors: Jonathan Miao, Rintaro Fujita, William Melicher, Michael Stroucken, Lujo Bauer, and Limin Jia



DOM based Cross Site Scripting

- One of the most exploited web application vulnerabilities
- Malicious scripts are loaded into browsers
- Occurs client-side only, making it hard to detect
- Can steal passwords, authentication tokens, and login cookies

Our Approach

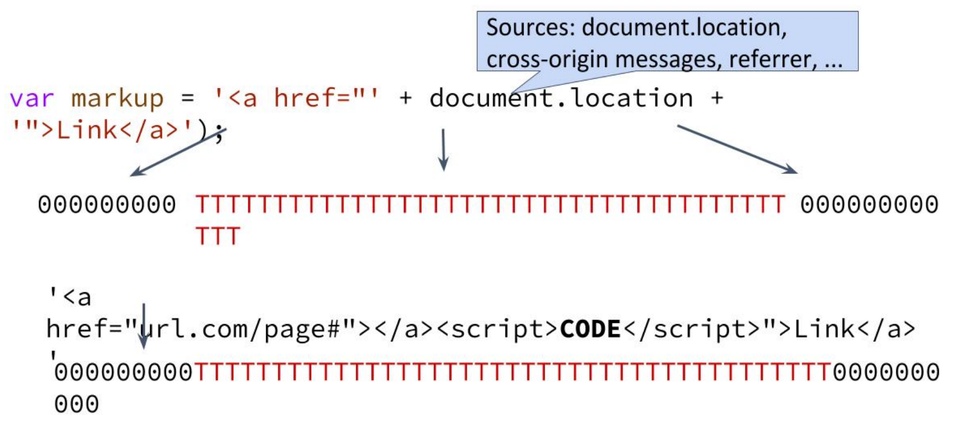
- Use Dynamic Analysis
  - Fast and works against complex programs
  - Taint Tracking detects potential vulnerabilities
  - Simulate an attack to confirm vulnerabilities
  - Previous approach did not interact with webpage
- Simulating User and Page Events
  - More code executed means broader coverage
  - Monkey Testing
  - Event Generation

Experiment Set-Up

- Taint Tracking
- Two Event Generating Extensions
- Chrome DevTools Code Coverage

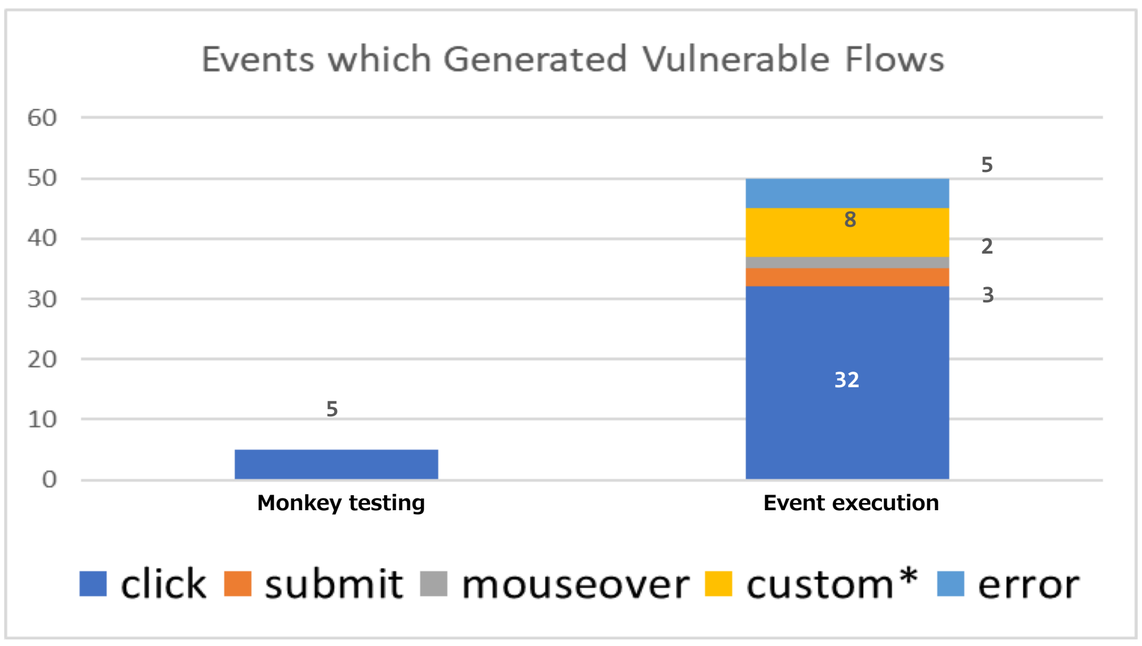
Taint Tracking

- Strings from external sources are marked with taint (T)
- Taint is propagated through operations such as concatenation
- Potential vulnerability found when taint reaches a vulnerable function



Results

Level	# of Crawled Pages	Average
Top Page	189	6.3%
1st Level Page	720	4.2%
2nd level Page	723	1.7%



- Monkey Testing: 0.82% increase in detections
- Event Execution: 8.70% Increase in detections
- Monkey Testing increased code coverage by roughly 16 percentage points
- Clicks made up 64% of new vulnerabilities

Future Work

- Pass in smarter parameters to events
- Sequencing the order in which events are triggered
- Crawling pages that require login credentials

Impact

- Improve web page security
- Discover common vulnerabilities