U Can’t Debug This: Detecting JavaScript Anti-Debugging Techniques in the Wild

Marius Musch and Martin Johns
TU Braunschweig
Motivation

- Static analysis of websites is hard
  - JavaScript is weakly typed
  - Code is usually minified, sometimes obfuscated
  - Dynamic creation of new code is common
Motivation

- Dynamic analysis only works with a real browser
  - Code expects DOM
  - Code expects non-standard objects
Dynamic analysis gives websites the power to mess with you and your results!

=> Anti-Debugging
Anti-Debugging in action

DEMO
Study — Crawl

- Crawl of 1 million sites
  - Including 3 subpages beyond the frontpage
  - Log indicators related to 6 anti-debugging techniques

- Result
  - Found breakpoints on 1128 sites (502 only on subpages)

- However, triggering a breakpoint exactly once is by no means Anti-Debugging already
Study — Severity

- Breakpoint frequency within sites

- Determine severity score
  - Triggering a breakpoint once vs five times per second
  - Triggering breakpoints vs less intrusive techniques
  - Using one technique vs multiple on the same page
Study — Replay

- Focus on 2000 most severe cases

- Record website visit once using WPR proxy
  - Replay multiple times while measuring code coverage
  - After a while, no new code gets executed

- Then, replay with DevTools open and look for new code
  - Allows us to detect 3 additional techniques
Study — Sidechannels

```javascript
> function measure() {
    const start = performance.now();
    debugger;
    const time = performance.now() - start;
    if (time > 100) {
        alert("DevTools are open!");
    }
}
setInterval(measure, 1000);
```

- **Result**
  - 229 sites executed different code when under analysis
9 Anti-Debugging techniques

- Impede
  - Preventing shortcuts
  - Triggering breakpoints
  - Clearing the console

- Alter
  - Modifiying built-in functions

- Detect
  - 2 basic techniques
  - 3 sophisticated techniques (based on timing side-channels)
More in the paper

- Includes a total of 9 techniques
- Systemization of techniques
- Many additional analyses
- Discussion of maliciousness
- ... and more!

https://js-antidebug.github.io
Thanks for listening :) 

Questions?

Contact
- Mail: m.musch@tu-bs.de
- Twitter: @m4riuz