Web Browser Exploitation Training

Samuel Groß

Bio

Samuel Groß is an independent security researcher and, in his spare time, a Master's student at Karlsruhe Institute of Technology. He has been researching browser security for some years now and has published multiple articles on the subject, including a Phrack paper about JavaScript engine exploitation techniques at the example of JavaScriptCore, the JavaScript engine inside WebKit/Safari. He successfully participated in the yearly Pwn2Own contest in 2017 and 2018, both times demonstrating a remote exploit against Safari which also gained root or kernel-mode code execution on the underlying macOS system.

Abstract

Modern web browsers pose a challenging and attractive target for security researchers. However, with ever growing codebases and increasing code complexity, the barrier to entry for security research in this area has been rising as well. This training is designed to prepare students for a successful entry into this field. Students will learn to identify, analyze, and exploit vulnerabilities in the context of a web browser renderer process. Through various hands-on exercises, students get practical experience and gain a good understanding of the respective code bases. Exercises will be designed for Chrome and Firefox, although many of them can also be completed on Edge and/or Safari.

The training will roughly be divided into two parts: the first part provides an in-depth introduction to web browser internals, mainly the DOM and JavaScript engines, with a focus on security relevant aspects. The second part of the training will then focus on identifying and exploiting vulnerabilities in the renderer process, and where to go from there.

While no previous experience with browser internals is required, students should be moderately familiar with memory corruption exploitation, low-level process internals, common debuggers, and C++. For students that do not wish to install compiler toolchains etc. directly on their laptops, Linux-based virtual machine images for Firefox and Chromium will be provided.

Objectives

* Understanding of browser internals
* Working exploit development environment
* Familiarity with browser code base(s)
* Knowledge of typical vulnerabilities
* Ability to identify and reliably exploit vulnerabilities in the renderer process
* Techniques for post (renderer) exploitation

Agenda

Day 1:
- High-level browser overview
- Scripting basics
- DOM Tree
- DOM Events
- Memory (mis)management

Day 2:
- JavaScript engine introduction
- JavaScript VMs and the stdlib
- "Usermode" callbacks in JS
- Garbage Collection

Day 3:
- JIT Compilers

Day 4:
- Vulnerability analysis and discovery
- Building exploit primitives
- Getting native code execution
- The Same-origin Policy and how to bypass it

Prerequisites

Students should

* Be familiar with memory corruption exploitation
* Be comfortable with C++ and know some JavaScript basics
* Bring a laptop running Windows, macOS or Linux, with at least 75GB disk space and 8GB RAM
* Have VMWare installed if they want to use the provided VM images