Detecting Client Side Prototype Pollution
Short talk

Samuel Kajava
Chalmers University of Technology and University of Gothenburg

Abstract

Prototype pollution is a vulnerability that exploits the internals of the inheritance mechanism in JavaScript (JS). Given the widespread use of the language, detecting and mitigating the threats introduced by the vulnerability is critical. Prior work has made significant strides on server-side code, leaving a gap in client-side JS.

Our work aims to bridge this gap by addressing issues with existing approaches such as reliance on outdated software, limited code coverage, and costly analysis. We tackle these challenges by using state of the art tools for static analysis, high coverage source code retrieval, and readily available means to verify identified vulnerabilities.

Our approach is realized by creating a multistage framework for detecting prototype pollution vulnerabilities utilizing webcrawling, static analysis with CodeQL, static payload generation, and dynamic gadget detection.

In this talk, I will cover our approach on a high-level and present some findings from the wild.

This work is a joint collaboration between Samuel Kajava, SiKai Lu, Eric Cornelissen, Benjamin Lundblad, Musard Balliu, and Andrei Sabelfeld.