

Abstract of my PhD Dissertation

”Security Analysis of Web Browsers and PDF Viewers by Comparing DOM Graphs”.

Dominik Trevor Noß

09.09.2025

1 Abstract

Web browsers and PDF viewers execute JavaScript code that interacts with the document through the Document Object Model (DOM), thereby providing access to sensitive user data, making its security critical for user privacy. Security analysis of the DOM is hindered by its complexity and the unavailability of methods for exhaustively surveying the attack surface.

DOM graphs are mathematical representations that capture the structure of JavaScript environments by modeling objects as nodes and their relationships as labeled edges in a directed graph. This approach enables systematic analysis of complex DOM structures.

DOM graph crawling is the automated process of traversing all reachable JavaScript properties and objects within a browser or PDF viewer environment, systematically discovering and documenting the complete attack surface. A specialized crawler traverses from the global object through all accessible properties, collecting detailed information about each encountered element and storing the results in a structured graph format.

Systematic DOM graph comparison applies this methodology at scale through automated mass-evaluation frameworks that generate and process extensive test suites. I developed tools that create and evaluate over 150 000 test cases across multiple browsers, automatically extracting DOM graphs under different conditions and comparing them using a novel graph comparison algorithm that identifies differences by removing common isomorphic subgraph components.

Cross-Site Leaks (XS-Leaks) are vulnerabilities that allow attackers to infer unauthorized state information from other websites by exploiting observable differences in browser behavior. By applying systematic DOM graph comparison to state-dependent resources under different conditions, this methodology automatically discovers all leak techniques. This approach identified several new, now responsibly disclosed, classes of XS-Leaks, leading to improved browser security.

PDF viewers present additional challenges due to inconsistent, diverse implementations, and undocumented features that make security analysis challenging. I extended the DOM graph methodology to PDF JavaScript environments, developing novel extraction techniques that work despite limited export capabilities. This enables statistical and intersectional analysis of JavaScript implementations across PDF viewers, quantifying implementation disparities and revealing the fragmented state of PDF JavaScript support.

My research contributes a reproducible, systematic methodology for comparative security analysis of JavaScript environments, open-source tooling and empirical studies that enhance both browser and PDF viewer security.

Deutsche Kurzzusammenfassung der Doktorarbeit “Sicherheitsanalyse von Webbrowsern und PDF Leseprogrammen mithilfe von DOM-Graph-Vergleich”

Dominik Trevor Noß

09.09.2025

1 Englische Originaltitel

Der englische Originaltitel lautet “Security Analysis of Web Browsers and PDF Viewers by Comparing DOM Graphs”.

2 Deutsche Kurzzusammenfassung

Webbrowser und PDF-Viewer führen JavaScript-Code aus, der über das Document Object Model (DOM) mit dem Dokument interagiert und somit Zugriff auf sensible Benutzerdaten gewährt, weshalb dessen Sicherheit für den Datenschutz der Benutzer von entscheidender Bedeutung ist. Die Sicherheitsanalyse des DOM wird durch dessen Komplexität und das Fehlen von Methoden zur umfassenden Untersuchung der Angriffsfläche erschwert.

DOM-Graphen sind mathematische Darstellungen, die die Struktur von JavaScript-Umgebungen erfassen, indem sie Objekte als Knoten und ihre Beziehungen als beschriftete Kanten in einem gerichteten Graphen modellieren. Dieser Ansatz ermöglicht eine systematische Analyse komplexer DOM-Strukturen.

DOM-Graph-Crawling ist der automatisierte Prozess des Durchlaufens aller erreichbaren JavaScript-Properties und -Objekte innerhalb einer Browser- oder PDF-Viewer-Umgebung, um die gesamte Angriffsfläche systematisch zu ermitteln und zu dokumentieren. Ein spezieller Crawler durchläuft vom Global Object aus alle zugänglichen Properties, sammelt detaillierte Informationen über jedes gefundene Element und speichert die Ergebnisse in einem strukturierten Graphenformat.

Systematischer DOM-Graph-Vergleich wendet diese Methodik in großem Maßstab durch automatisierte Massenauswertungs-Frameworks an, die umfangreiche Testreihen generieren und verarbeiten. Ich habe Tools entwickelt, die über 150 000 Testfälle für mehrere Browser erstellen und auswerten, DOM-Graphen unter verschiedenen Bedingungen automatisch extrahieren und sie mithilfe eines neuartigen Graphvergleichsalgorithmus vergleichen, der Unterschiede durch Entfernen gemeinsamer isomorpher Teilgraphkomponenten identifiziert.

Cross-Site-Leaks (XS-Leaks) sind Schwachstellen, die es Angreifern ermöglichen, durch Ausnutzen beobachtbarer Unterschiede im Browserverhalten unbefugte Statusinformationen von anderen Websites abzuleiten. Durch die systematische Anwendung des DOM-Graphenvergleichs auf zustandsabhängige Ressourcen unter verschiedenen Bedingungen entdeckt diese Methodik automatisch alle Leak-Techniken. Dieser Ansatz identifizierte mehrere neue, inzwischen verantwortungsvoll veröffentlichte Klassen von XS-Leaks, was zu einer verbesserten Browsersicherheit führte.

PDF-Viewer stellen aufgrund inkonsistenter, vielfältiger Implementierungen und undokumentierter Funktionen, die die Sicherheitsanalyse erschweren, zusätzliche Herausforderungen dar. Ich habe die DOM-Graph-Methodik auf PDF-JavaScript-Umgebungen ausgeweitet und neuartige Extraktionstechniken entwickelt, die trotz begrenzter Exportmöglichkeiten funktionieren. Dies ermöglicht eine statistische Analyse und Schnittmengenanalyse von JavaScript-Implementierungen in verschiedenen PDF-Viewern, quantifiziert Implementierungsunterschiede und deckt den fragmentierten Zustand der PDF-JavaScript-Unterstützung auf.

Meine Forschung trägt zu einer reproduzierbaren, systematischen Methodik für die vergleichende Sicherheitsanalyse von JavaScript-Umgebungen, Open-Source-Tools und empirischen Studien bei, die sowohl die Browser- als auch die PDF-Viewer-Sicherheit verbessern.