FEROOT.COM

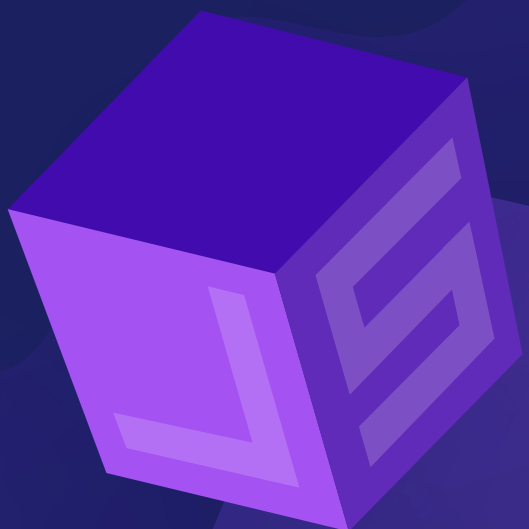
feroot

The Ultimate Guide to

JavaScript Security

EXECUTIVE SUMMARY

Learn everything you need to know about JavaScript security to protect your web applications and websites from cyberthreats.



While there are countless benefits to JavaScript, there are also a few major downsides, not the least of which is JavaScript's inherent 'hackability.' In our new e-book: The Ultimate Guide to JavaScript Security, Feroot provides web application developers and security professionals with a guide for understanding JavaScript within the context of securing modern websites and web applications. The guide highlights the fundamental risks associated with using JavaScript in an unprotected client-side environment and what web application developers and security professionals can do to better protect their websites and website users.

The power of JavaScript is evident across today's digital landscape. Almost 98% of all websites use JavaScript as the client-side programming language to add interactive behavior to webpages and to create web and mobile apps. E-commerce sites depend on JavaScript heavily to support the user experience during the shopping and purchasing process; banking websites use it to support customer forms; and businesses use it for advertising and to track web analytics. JavaScript is a crucial component of front-end development, yet it remains extremely vulnerable to attacks, since it is easy for hackers and attackers to manipulate JavaScript code to access, steal, or contaminate protected data.

It is no longer enough to simply secure the perimeter with tools like firewalls. Organizations must secure the JavaScript that drives the front end or client side of their websites and web applications to protect customers, minimize risk, and ensure business growth.

E-book Components

This book contains five primary sections: Client-Side Attacks and JavaScript Code, Securing JavaScript, JavaScript Security Approaches & Technologies, JavaScript Risks and Threats, and JavaScript Security: Teams and Collaboration.

Each of these sections addresses key issues related to JavaScript security from a client-side perspective and provides the reader with guidance on how to better protect customers and businesses.

E-book Highlights

Client-Side Attacks and JavaScript Code

- JavaScript serves as one of the core technologies used to build web applications and websites, with 98% of websites using it for client-side webpage behavioral elements and 80% of websites employing third-party JavaScript libraries or web frameworks for their client-side scripting.
- JavaScript is a client-side language processed by the web browser, not the web server, which means client-side activity happens outside of the business's security perimeter. Traditional security technologies, like firewalls, will not protect the end-user from malicious activity occurring on the client side.
- Research suggests that website and web application attacks account for more than a quarter of all data breaches.
- Common attack types include e-skimming, Magecart, sideloading, cross-site scripting (XSS), and formjacking.
- JavaScript is vulnerable because it is easy for hackers and other threat actors to input query strings into forms to access, steal, or contaminate protected data. By default, JavaScript environments do not have a security permissions model built in.
- Third- and fourth-party scripts, plug-ins, and extensions are usually written in JavaScript. These tools present added risk since vulnerabilities and bugs can be embedded in the code.
- JavaScript code can lie undetected and seemingly benign, while performing countless nefarious acts such as intercepting customer information from a form or stealing credit card information.

Securing JavaScript

- Processes for security JavaScript include code reviews, authentication, authorization, and code testing.

JavaScript Security Approaches & Technologies

- Automated scanning tools are the best way to detect, identify, and alert on behavior anomalies within JavaScript.
- There are seven primary tools available to help prevent client-side attacks, each offering varying degrees of protection—web application firewalls (WAF), content security policy (CSP), penetration testing and assessments (vulnerability and security), client-side vulnerability scanning, code scramblers and obfuscators, client-side attack surface monitoring, and JavaScript security permissions.

JavaScript Risks and Threats

- JavaScript risks and threats include at least 25 different types: cross-site scripting (XSS), DOM-based XSS, directory traversal or path traversal, web skimming, e-skimming, Magecart, e-commerce platform skimming, drive-by web skimming, trusted cloud-hosted platform skimming, public Wi-Fi skimming, anti-forensic skimming, self-cleaning skimming, stealth data skimming, malicious script injection attacks, JavaScript injection, SQL injection, XML entity injection, formjacking, chainloading and sideloading, JavaScript sniffing, broken link hijacking, request forgery, server-side request forgery, and cross-site request forgery.
- Industries at high risk of a JavaScript attack include financial services & banking, insurance, healthcare & medical, e-commerce & retail, travel & hospitality, communication, social media, & content producers, and cryptocurrency exchanges & blockchain.

JavaScript Security: Teams and Collaboration

- Cybersecurity professionals should work with all business teams, particularly application development, marketing, privacy & compliance, and product security (as applicable) to:
  - Build strong relationships.
  - Promote a secure business mission, remove friction in the customer journey, and facilitate success for the business.
  - Understand current or emerging privacy trends or regulations and apply them within a cybersecurity context.
  - Develop a strong security architecture.

To secure JavaScript web applications, organizations should:

- Deploy JavaScript security processes, procedures and technologies to continuously protect webpages and web applications.
- Understand and prepare for JavaScript risks and threats.
- Build strong cross-functional relationships with security, marketing, product management and compliance teams.

To download the full e-book, please click here.