

## 5.1

### XSS Basics

XSS is the most prevalent web application vulnerability found in the wild. It afflicts the vast majority of live applications, including some of the most security-critical applications on the Internet, such as those used by online banks. As research into client-side attacks has developed, the discussion has focused on numerous other attacks that are at least as convoluted to exploit as an XSS flaw. And numerous real-world attacks have occurred in which XSS vulnerabilities have been used to compromise high-profile organizations. XSS often represents a critical security weakness within an application. It can often be combined with other vulnerabilities to devastating effect. An XSS attack can be turned into a virus or self-propagating worm in some situations.

### Anatomy of an XSS exploitation

A cross-site scripting attack is typically done with a specially crafted URI that an attacker provides to their victim. The XSS attack could be considered analogous to a buffer overflow, where the injected script is similar to overwriting an EIP. In both techniques, there are two options once a successful attack has occurred: insert funny data or jump to another location. Insertion of funny data during a buffer overflow typically results in a denial of service attack. In the case of an XSS attack, it allows the attacker to display arbitrary information, and suppress the display of the original webpage. When jumping to another location during a buffer overflow attack, the new location is another location in memory where shellcode or other important data resides – allowing the attacker to take control of the flow of the program. Within the XSS context, the attacker instead jumps the victim to another location on the Internet (typically under the attacker’s control), hijacking the victim’s web browsing session.

### Types of XSS

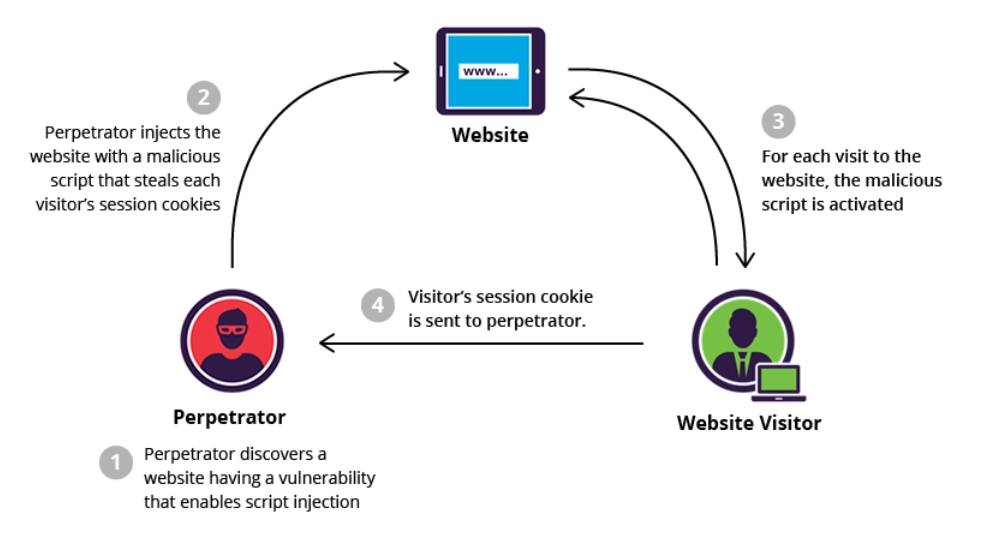
#### Reflected XSS Vulnerabilities

Reflected XSS attacks, also known as non-persistent attacks, occur when a malicious script is reflected off of a web application to the victim's browser. The script is activated through a link, which sends a request to a website with a vulnerability that enables the execution of malicious scripts. The vulnerability is typically a result of incoming requests not being sufficiently sanitized, which allows for the manipulation of a web application’s functions and the activation of malicious scripts.

To distribute the malicious link, a perpetrator typically embeds it into an email or third party website (e.g., in a comment section or in social media). The link is embedded inside an anchor text that provokes the user to click on it, which initiates the XSS request to an exploited website, reflecting the attack back to the user.

#### Stored XSS Vulnerabilities

Stored XSS, also known as persistent XSS occurs when a malicious script is injected directly into a vulnerable web application. To successfully execute a stored XSS attack, a perpetrator has to locate a vulnerability in a web application and then inject malicious script into its server (e.g., via a comment field). One of the most frequent targets are websites that allow users to share content, including blogs, social networks, video sharing platforms and message boards. Every time the infected page is viewed, the malicious script is transmitted to the victim’s browser.



#### DOM-Based XSS Vulnerabilities

DOM XSS stands for Document Object Model-based Cross-site Scripting. A DOM-based XSS attack is possible if the web application writes data to the Document Object Model without proper sanitization. The attacker can manipulate this data to include XSS content on the web page, for example, malicious JavaScript code.

The Document Object Model is a convention used to represent and work with objects in an HTML document (as well as in other document types). All HTML documents have an associated DOM that consists of objects, which represent document properties from the point of view of the browser. When a client-side script is executed, it can use the DOM of the HTML page where the script runs. The script can access various properties of the page and change their values.

An attacker may use several DOM objects to create a Cross-site Scripting attack. The most popular objects from this perspective are document.url, document.location, and document.referrer.

### Finding XSS

### XSS Exploitation

### Browsers and same origin policy

### Cookie stealing through XSS

### Defacement

### Advanced phishing attacks

## 5.2

### SQLi Basic

### Types of SQLi

### SQLi Exploitation

## 5.3

### Analyzing Network Traffic

### Performing Man in the Middle Attack

### 