**Essay Paper: Cross-Site Scripting Attack**

**Introduction:**

Cross-Site Scripting (XSS) is a website attack where malicious codes get injected into existing websites. This type of cyber-attack is also known as a client-side code injection attack. In the event of an XSS occurrence, the attacker sends malicious codes and scripts to the user's website. The method aims to run and execute the codes on the end user's device (Rodríguez et al., 2020). Furthermore, the attacker's strategy works only when the targeted user visits the intended website to run the codes. The process utilizes unvalidated inputs to change the outputs from the unsuspecting user. The common avenues where XSS attacks occur include message boards, forums, and web pages that permit comments.

The rise in technological advancement contributes to an increase in the sophistication of internet growth. The internet is the principal tool for communication, especially in business, due to its flexibility and reliability. Most of the businesses rely on the internet in sending information to their staff as well as clients. Besides, business organizations monitor their financial transactions through the internet. This practice results in having sensitive info left on the web pages in the form of cookies (Ayeni et al., 2018). Therefore, the web attackers invent methods such as XSS to mine and alter such information for their use. This essay will discuss how XSS attacks target specific people with various motivations and analytics on the findings of XSS attacks.

**Review of Literature:**

According to research by Rodríguez et al. (2020), web applications are insecure by default. This makes the website vulnerable to manipulation from malicious persons. Additionally, the flaws from the web development process give the attackers the advantage of penetrating through the protocols (Rodríguez et al., 2020). The consequences of such attacks include thefts of cookies data to steal the identity of the user. Also, one may control the end-user browser without their consent if they have access to their input codes. The findings by the authors indicated XSS is the second most vulnerable attacks on websites.

Additionally, the authors stated that the evaluation of XSS occurs in five parameters. They include business impacts, exploitable, technical impacts, weakness detectability, and weakness prevalence (Rodríguez et al., 2020). Moreover, the Common Weakness Enumeration (CWE) shows porous defenses as being the primary cause of XSS attacks. However, Rodríguez et al. (2020) indicated XSS might occur when a web application dynamically generates untrusted data. This phenomenon occurs differently from when an attacker sends the unvalidated inputs to the user.

Similar research by Ayeni et al. (2018) stated that an XSS attack might occur even in cases where the database engine or the servers contain no vulnerability. This characteristic makes XSS the most common web attack currently. Moreover, the authors stated that XSS's vulnerability increases when the input data is not properly sanitized (Ayeni et al., 2018). The targeted input sources by the attackers include cookies, HTML forms, and hidden fields. Hence the result of the attacks includes identity theft, litigations, and loss of customers in case there is a breach of private customer data.

**How XSS Works:**

The XSS begins when an attacker identifies their targeted website to infect with the malicious codes. The website has to be vulnerable where untrusted data can enter in the web application. Through a web request, the targeted victim visits the website through a browser (Ayeni et al., 2018, 2018). The script then gets executed as dynamic content without validation for malicious content. Therefore, the attacker's code might siphon information from the website, especially the user's profile and other relevant details.

**Types of XSS Attacks:**

**Direct/Persistent XSS Attacks:**

The persistent XSS attacks involve injecting the malicious code directly into the website or the targeted vulnerable site. The method uses scripts such as JavaScript to execute the codes. This type of attack makes the code permanent to the website (Ayeni et al., 2018). It only requires first execution from a user and then recurs every time a user visits the website. The code will then execute the programmed actions in the browser. This type of attack is the most dangerous since the malicious codes get permanently stored in the servers (Ayeni et al., 2018). An attacker might then use the privilege to mine data anytime the targeted users visit the websites.

**Indirect/Non-persistent XSS Attacks:**

In the non-persistent XSS, the attacker steals the victim's cookie sessions through the use of malicious scripts. The stolen cookie then allows the attacker to impersonate the victim in their sessions (Rodríguez et al., 2020). The malicious codes get injected in the website through cookies, Uniform Resource Locator (URL), and forms. This XSS attack method utilizes web applications' vulnerability to get the user's info to generate a response (Rodríguez et al., 2020). The attackers may use links to lure the user into submitting their information. The user’s browser will then execute the codes since they come from trusted servers.

**Document Object Module (DOM) XSS Attacks:**

In the DOM-based XSS, the attackers may place malicious files or URLs on the client’s side. In such cases, the script is not sent to the server but executed by the browser. When the user clicks on the malicious URL, the browser executes the code where it an manipulate the elements of the web page (Rodríguez et al., 2020). This method of attack gives the attacker privileges of controlling some essentials on the user's webpage. Due to its effectiveness, DOM XSS accounts for over 60% of attacks between 2015 and 2019 (Rodríguez et al., 2020).

**Data and Analytics on XSS:**

According to an article by Farrell (2019), cyberattacks affected nearly 75% of large companies in North America and Europe throughout 2019. Moreover, 40% of the cyber attacks were due to XSS (Farrell, 2019). The data shows XSS is the most preferred tool by the cybercriminals. Since the attacks happened globally, some of the hackers indicated they participated as a challenge. Besides, 36% of the hackers stated they attacked the companies to test their web applications' security measures. The article further states 72.3% of the cyber-attacks happen on websites (Farrell, 2019). WordPress is among the prime targets, with 98% of its vulnerability related to plugins.

An article by Positive Technologies (2019) found more websites are vulnerable to information exposure. This leakage may occur when the sites get attacked by cybercriminals. In 2018, the percentage of websites vulnerable to attacks was 67 (Positive Technologies, 2019). Additionally, these websites had a high vulnerability to XSS attacks. The data indicated a 77% vulnerability in XSS attacks in 2017. The research showed configuration errors in four out of five websites, which made them susceptible to attacks (Positive Technologies, 2019). They include error reporting, default settings, standard passwords, and full path disclosure. In 2018, the susceptibility to XSS attacks rose to 88.5%.

The findings from these studies show how XSS is the most preferred way of attack by cybercriminals. Also, this nature through which the attackers can send malicious scripts to the users makes it convenient to mine clients' data. Besides, the method rarely leaves traces of the attacker’s info hence common with anonymous hackers.

**Conclusion:**

This essay discussed how the XSS attacks happen, their target, and the data showing the prevalence of cybercrime. XSS attacks involve sending malicious codes into users' websites to compromise their protocols (Rodríguez et al., 2020). Most of the attackers usually send scripts that would enable them to have access to private info from the users when executed. The execution of such malicious scripts is possible by the extensive use of internet technology. Research indicates that the websites' default vulnerability status gives the attackers the chance to manipulate the browsers of their targeted users (Ayeni et al., 2018). Besides, XSS attacks may occur even when browsers and the servers do not pose a threat to the users. The direct XSS attack allows the script to permanently become executed on the website causing more threats to the users. Moreover, the DOM attack provides control of some elements in the target's website without reaching the server.

Data from 2019 web analytics indicated 40% of website attacks occurred through XSS (Farrell, 2019). This figure showed a rise in website attacks from previous years. Therefore, the info indicates XSS as the preferred method by the cyber attackers. However, web developers could help prevent XSS by sanitizing the inputs. This process would ensure the input data get scanned for malicious code before generating output. Additionally, developers should restrict the use of HTML inputs among trusted users. The strategy would prevent malicious scripts from getting to the input data.

**References:**

Ayeni, B. K., Sahalu, J. B., & Adeyanju, K. R. (2018). Detecting cross-site scripting in Web applications using a fuzzy inference system. *Journal of Computer Networks and Communications*, *2018*. <https://doi.org/10.1155/2018/8159548>

Farrell, N. (2019). *Cross-Site Scripting was a Huge Chunk of 2019 Cyber-Attacks*. Fudzilla. <https://www.fudzilla.com/news/50014-cross-site-scripting-xss-were-a-huge-chunk-of-2019-cyber-attacks>

Positive Technologies. (, 2019). *Web Application Vulnerabilities: Statistics for 2018*. <https://www.ptsecurity.com/ww-en/analytics/web-application-vulnerabilities-statistics-2019/>

Rodríguez, G. E., Torres, J. G., Flores, P., & Benavides, D. E. (2020). Cross-site scripting (XSS) attacks and mitigation: A survey. *Computer Networks*, *166*, 106960. <https://doi.org/10.1016/j.comnet.2019.106960>