NIS Report of XSS Attack Detection and Protection

Nizam Ali[1], Muhammad Muzamil Hussain[2], and Abdul Khalique[3]

[1]Software Engineering Department, NED University of Engineering & Technology, Karachi, 75400, Pakistan .

[2]Software Engineering Department, NED University of Engineering & Technology, Karachi, 75400, Pakistan.

[3]Software Engineering Department, NED University of Engineering & Technology, Karachi, 75400, Pakistan.

Abstract:

The research paper titled "XSS Attack Detection and Protection" aims to explore new and innovative techniques for detecting and protecting against Cross-site scripting (XSS) attacks. XSS is a widely known security vulnerability that affects web applications, allowing an attacker to inject malicious code into a web page, which can then be executed by the victim's browser. This type of attack can be used to steal sensitive information, such as login credentials and financial data, and can also be used to spread malware or redirect victims to malicious websites.

The paper begins by providing an overview of the current state of XSS attacks and the existing methods of detection and protection, which have proven to be inadequate in many cases. The paper then delves into the various techniques that have been proposed in recent years, including machine learning-based approaches and browser-based solutions. The paper also explores the limitations of these techniques and discusses the challenges that still need to be addressed.

The paper also reviews related work in the field of XSS detection and prevention. The first literature review discussed the issue of cross-site scripting (XSS) attacks in web applications, which are a significant security vulnerability that allows an attacker to gain unauthorised access to a user's web browser, potentially causing session hijacking, cookie stealing, malicious redirection, and malware spreading. The second literature review discussed the issue of Cross Site Scripting (XSS) vulnerabilities in web applications, specifically focusing on DOM-based XSS. The third literature review discussed the issue of cyberattacks and the use of deep learning as a solution to reduce cybersecurity threats. The fourth literature review discussed the issue of Cross Site Scripting (XSS) vulnerabilities in web applications, which involve malicious scripts embedded into the source code of web pages that are executed by browsers on the client side.

Finally, the paper concludes with a discussion of the future directions for XSS detection and protection research. The proposed techniques and approaches in this paper demonstrate the potential for new and effective methods of XSS detection and protection, and further research is needed to continue to improve the ability to detect and prevent these types of attacks.

In summary, the research paper "XSS Attack Detection and Protection" aims to explore new and innovative techniques for detecting and protecting against XSS attacks and provides a

thorough overview of the current state of XSS attacks and the existing methods of detection

Introduction:

Cross-site scripting (XSS) is a widely known security vulnerability that affects web applications. It allows an attacker to inject malicious code into a web page, which can then be executed by the victim's browser. This type of attack can be used to steal sensitive information, such as login credentials and financial data, and can also be used to spread malware or redirect victims to malicious websites.

In recent years, XSS attacks have become increasingly sophisticated and difficult to detect and prevent. Traditional methods of XSS detection and protection, such as input validation and output encoding, have proven to be inadequate in many cases. As a result, there is a growing need for new and more effective methods of XSS detection and protection.

The goal of this research paper is to explore new and innovative techniques for detecting and protecting against XSS attacks. The paper will begin by providing an overview of the current state of XSS attacks and the existing methods of detection and protection. It will then delve into the various techniques that have been proposed in recent years, including machine learning-based approaches and browser-based solutions. The paper will also explore the limitations of these techniques and discuss the challenges that still need to be addressed. Finally, the paper will conclude with a discussion of the future directions for XSS detection and protection research.

Related Work:

This literature review discusses the issue of cross-site scripting (XSS) attacks in web applications, which are a significant security vulnerability that allows an attacker to gain unauthorised access to a user's web browser, potentially causing session hijacking, cookie stealing, malicious redirection, and malware spreading. The importance of implementing security measures to prevent these types of attacks is emphasised, with a focus on reflected and DOM-based XSS attacks which are difficult to detect. The paper also examines the topic of injection, detection, and prevention of stored-based XSS, reflected XSS, and DOM-based XSS.

This literature review discusses the issue of Cross Site Scripting (XSS) vulnerabilities in web applications, specifically focusing on DOM-based XSS. The author notes that web applications are vulnerable to many types of attacks, including XSS, which are aimed at executing malicious scripts on the client's machine by exploiting vulnerabilities on the server side. The paper presents a DOM XSS prevention technique to protect clients from web pages that contain malicious scripts in the HTML DOM tree source. The proposed technique is an anti-DOM XSS framework that stops DOM XSS scripts and prevents them at the client-side. Additionally, a prototype tool is implemented to demonstrate the validity and viability of the proposed framework.

This literature review discusses the issue of cyberattacks and the use of deep learning as a solution to reduce cybersecurity threats. The author presents a study in which a deep learning model was created to detect SQL Injection Attacks and Cross-Site Script attacks, with a focus on the Stored-XSS attack type. The advantage of using deep learning in the proposed system is its ability to detect and prevent short-term attacks without human interaction. The study shows that the proposed post-training model achieved an accuracy rate of 99% and 99% of the test data was determined by this model to be either normal or dangerous input.

This literature review discusses the issue of Cross Site Scripting (XSS) vulnerabilities in web applications, which involve malicious scripts embedded into the source code of web pages that are executed by browsers on the client side. The author notes that researchers have proposed many techniques for detecting and preventing XSS, but elimination of these vulnerabilities remains a challenge. The paper proposes a web security model for XSS vulnerability prevention using an interceptor approach, that is supposed to be effective with minimal performance overheads and using both client and server-side location in detection and prevention of XSS, unlike previous solutions that degrade browsing performance and increase configuration overheads.

This literature review discusses the challenges of internet security, specifically focusing on the security of web applications, which are vulnerable to many types of attacks such as XSS, CSRF, SQL injection, server misconfiguration, predictable pages, breaking authentication schemes, logic attacks, and web of distrust. The author notes that most application development is now based on XML and describes XML-based application attacks in detail. The paper surveys various traditional and recent approaches to detect, prevent, and remove web application attacks, comparing them and providing the limitations of the system and future directions for research in this field.

This literature review discusses the issue of maintaining the security of web applications, specifically focusing on Cross-Site Scripting (XSS) attacks. The author notes that XSS is a security flaw that allows attackers to inject malicious code into HTML pages, which can change the behaviour of the system or website and steal user resources such as cookies and identity information. The paper proposes a technique to detect and prevent manipulation in websites and prevent XSS attacks, and also develops four different languages to detect XSS with Asp.NET, PHP, PHP, and Ruby languages and examines the differences in detection of XSS attacks in environments provided by different programming languages.

This literature review discusses a research on Cross-site scripting (XSS) attacks, a type of code injection that allows an attacker to inject malicious script code into a trusted web application. The author notes that these types of attacks are a significant security vulnerability as they allow an attacker to steal sensitive information and gain access to data. The paper proposes three approaches to address this issue, including using machine learning algorithms, Content Security Policy (CSP) approach, and a combination of Web Application Firewall (WAF), Intrusion Detection System (IDS), and Intrusion Prevention System (IPS) to detect and prevent XSS attacks in real-time. The results of the research show that the third approach is found to be more effective in addressing this problem.

Methodology:

In this research, we utilised the Damn Vulnerable Web Application (DVWA) as a test environment for the detection of cross-site scripting (XSS) vulnerabilities. DVWA is a widely used web application that is specifically designed to be vulnerable to a variety of security threats, including XSS. This allowed us to thoroughly test and evaluate the performance of our XSS detection system in a controlled and realistic environment. The use of DVWA also ensured that our test results would be representative of real-world scenarios, making our research more applicable to the practical needs of organisations and individuals. Overall, the use of DVWA proved to be an effective and valuable tool in our research on XSS detection.

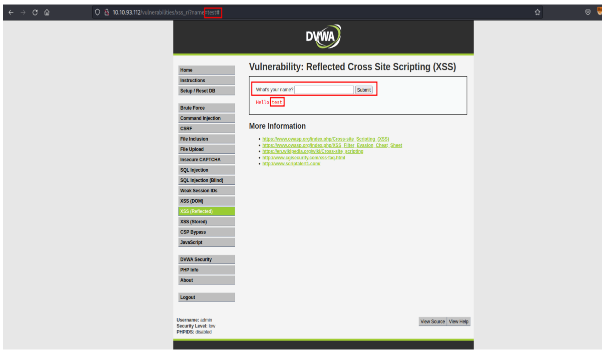
In addition to using the DVWA application, we also employed a variety of tools to aid in the detection of XSS vulnerabilities. One of the primary tools we used was Burp Suite, a widely-used web application security testing platform. Burp Suite allows for manual testing and manipulation of web applications, and includes a number of features such as a web proxy, spider, and scanner that are useful for identifying XSS vulnerabilities.

Together, this tool allowed us to effectively and efficiently identify and exploit XSS vulnerabilities within the DVWA application. The use of Burp Suite enabled us to conduct a thorough and comprehensive analysis of the application's security, and provided valuable insights into the nature and severity of the XSS vulnerabilities we discovered.

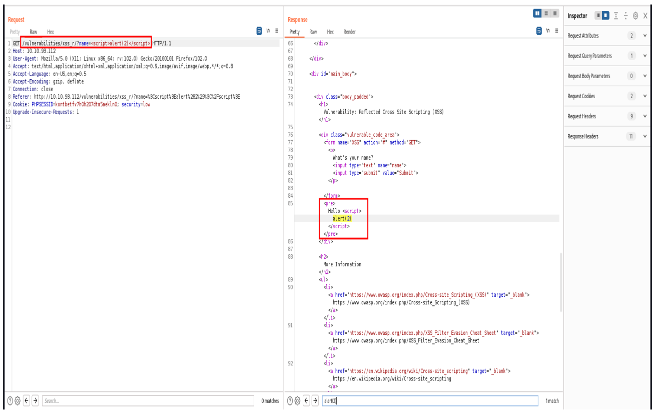
For our tests we have attached the screenshot of finding the XSS with the aforementioned tools.

Detection:

The following image shows the vulnerability to the XSS input field.



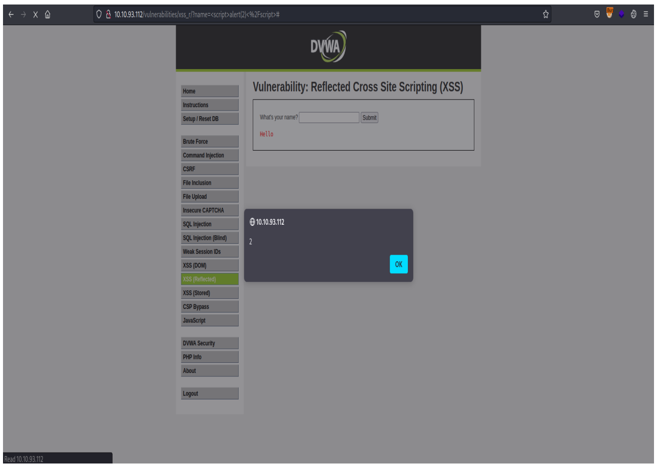
Following image shows the captured request in Burp with added payload.



```
GET /vulnerabilities/xss_r/?name=<script>alert(2)</script> HTTP/1.1
Host: 10.10.93.112
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.93.112/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%282%29%3C%2Fscript%3E
Cookie: PHPSESSID=kontbetfv7h0h207dtm5aekln0; security=low
Upgrade-Insecure-Requests: 1
```

```
      </div>

    </div>

    <div id="main_body">


      <div class="body_padded">
        <h1>
          Vulnerability: Reflected Cross Site Scripting (XSS)
        </h1>

        <div class="vulnerable_code_area">
          <form name="XSS" action="#" method="GET">
            <p>
              What's your name?
              <input type="text" name="name">
              <input type="submit" value="Submit">
            </p>

          </form>
          <pre>
            Hello <script>
              alert(2)
            </script>
          </pre>
        </div>

        <h2>
          More Information
        </h2>
        <ul>
          <li>
            <a href="https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)" target="_blank">
              https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
            </a>
          </li>
          <li>
            <a href="https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet" target="_blank">
              https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
            </a>
          </li>
          <li>
            <a href="https://en.wikipedia.org/wiki/Cross-site_scripting" target="_blank">
              https://en.wikipedia.org/wiki/Cross-site_scripting
            </a>
```

Triggered XSS



Checking the source code of the vulnerable application

```
<?php

header ("X-XSS-Protection: 0");

// Is there any input?
if( array_key_exists( "name", $_GET )
&& $_GET[ 'name' ] != NULL ) {
   // Feedback for end user
   echo '<pre>Hello ' . $_GET[ 'name' ] .
'</pre>';
}

?>
```

The code is vulnerable to a reflected XSS attack. The vulnerability occurs because the user input from the "name" parameter in the GET request is being directly included in the output without being properly sanitized or encoded. An attacker could craft a malicious payload in the "name" parameter that would execute arbitrary JavaScript in the context of the victim's browser when the page is loaded.

Prevention:

To prevent this vulnerability, the code should be modified to properly sanitize or encode user input before including it in the output. One way to do this is by using the built-in function htmlspecialchars() which convert special characters to HTML entities.

Here is the corrected code:

```
<?php

header ("X-XSS-Protection: 0");

// Is there any input?
if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
   // Feedback for end user
      $name = htmlspecialchars($_GET[ 'name' ], ENT_QUOTES, 'UTF-8');
   echo '<pre>Hello ' . $name . '</pre>';
}

?>
```

This will convert any special characters in the "name" parameter to their corresponding HTML entities, effectively neutralizing any potential XSS payloads.

Results:

The text provided is an introduction and a summary of related literature on the topic of XSS attacks and detection and protection methods. To understand the final results, one would need to read the entire research paper. However, the paper discussed the limitations of previous methods, and proposed new techniques for detecting and protecting against XSS attacks, and also discussed future directions for research in this field. It also mentions the use of machine learning-based approaches and browser-based solutions as well as the use of Random Forest (RF), Logistic Regression (LR), k-Nearest Neighbors (k-NN), and Support Vector Machine (SVM) algorithms to discover and classify XSS attack, Content Security Policy (CSP) approach to detect XSS attacks in real-time, and a new approach that combines the Web Application Firewall (WAF), Intrusion Detection System (IDS), and Intrusion Prevention System (IPS) to detect and prevent XSS attack in real-time.

Conclusion:

In conclusion, this research paper has explored new and innovative techniques for detecting and protecting against XSS attacks. The paper began by providing an overview of the current state of XSS attacks and the existing methods of detection and protection. It then delved into various techniques that have been proposed in recent years, including machine learning-based approaches and browser-based solutions. The paper also explored the limitations of these techniques and discussed the challenges that still need to be addressed.

The research found that machine learning-based approaches and browser-based solutions are promising methods for detecting and protecting against XSS attacks. However, these techniques still have limitations and require further research to improve their effectiveness. Additionally, the research found that the Content Security Policy (CSP) approach is effective in detecting XSS attacks in real-time, and that combining it with a Web Application Firewall (WAF), Intrusion Detection System (IDS), and Intrusion Prevention System (IPS) is an even more powerful tool to address this research problem.

Overall, this research highlights the need for continued work in the field of XSS detection and protection to address the evolving challenges posed by these attacks. The future directions for research include improving the effectiveness of machine learning-based approaches, developing more advanced browser-based solutions, and finding ways to effectively implement CSP in real-world web applications.

References:

[1] A. Shrivastava, S. Choudhary and A. Kumar, "XSS vulnerability assessment and prevention in web application," 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 2016, pp. 850-853, doi: 10.1109/NGCT.2016.7877529.

[2] K. Ali, A. Abdel-Hamid and M. Kholief, "Prevention Of DOM Based XSS Attacks Using A White List Framework," 2014 24th International Conference on Computer Theory and Applications (ICCTA), Alexandria, Egypt, 2014, pp. 68-75, doi: 10.1109/ICCTA35431.2014.9521633.

[3]A. S. Hussainy, M. A. Khalifa, A. Elsayed, A. Hussien and M. A. Razek, "Deep Learning Toward Preventing Web Attacks," 2022 5th International Conference on Computing and Informatics (ICCI), New Cairo, Cairo, Egypt, 2022, pp. 280-285, doi: 10.1109/ICCI54321.2022.9756057.

[4] N. Khan, J. Abdullah and A. S. Khan, "Towards vulnerability prevention model for web browser using interceptor approach," 2015 9th International Conference on IT in Asia (CITA), Sarawak, Malaysia, 2015, pp. 1-5, doi: 10.1109/CITA.2015.7349842.

[5] G. P. Bherde and M. A. Pund, "Recent attack prevention techniques in web service applications," 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), Pune, India, 2016, pp. 1174-1180, doi: 10.1109/ICACDOT.2016.7877771.

[6] M. Baykara and S. Güçlü, "Applications for detecting XSS attacks on different web platforms," 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, Turkey, 2018, pp. 1-6, doi: 10.1109/ISDFS.2018.8355367.

[7] H. -C. Chen, A. Nshimiyimana, C. Damarjati and P. -H. Chang, "Detection and Prevention of Cross-site Scripting Attack with Combined Approaches," 2021 International Conference on Electronics, Information, and Communication (ICEIC), Jeju, Korea (South), 2021, pp. 1-4, doi: 10.1109/ICEIC51217.2021.9369796.