IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 5, May 2023

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205436 |

Browser Security:(Web Security Vulnerabilities)

Dhanashree Kengale [1], Dr.Rama Bansode[2]

P.G. Student, PES's Modern College of Engineering, Pune, Maharashtra, India[1]

Astt.Professor, P.E.S. Modern College of Engineering, Pune, Maharashtra, India[2]

ABSTRACT: Browser security is a critical concern in the digital age, as browsers act as gateways to the internet, exposing users to numerous risks and vulnerabilities. This abstract explores the essential elements of browser security and emphasizes the need for robust protective measures. It discusses secure browsing practices, including regular updates, strong passwords, and cautious behavior while browsing. The importance of implementing HTTPS with SSL/TLS encryption for secure communication is highlighted. Additionally, it addresses the significance of carefully selecting and managing browser extensions and add-ons. The abstract underscores the role of browsers in phishing and malware protection, stressing the need for built-in security features. It emphasizes privacy and data protection measures such as private browsing, cookie management, and anti-tracking mechanisms. Furthermore, it highlights the value of multi-factor authentication in enhancing browser security. The abstract concludes with a call for security awareness and education to empower users to make informed decisions and safeguard themselves against online threats. By prioritizing these aspects, users can fortify their browser security and enjoy a safer online experience.

KEYWORDS: Web Security Vulnerabilities,SQL Injection, Cross Site Scripting, Security Standards, Insecure Direct Object References (IDOR), Cross-Site Request Forgery (CSRF)

I.INTRODUCTION

Browser protection is a category that encompasses the technology, gear, systems, and practices that rework browsers into relaxed environments. these solutions enable the internet get admission to to programs and websites at the same time as protecting the enterprise's structures and statistics.
With a browser security answer, companies can come across and block web-borne threats and risks that try to attack the browser itself or exploit it as an assault vector. those encompass malware, statistics theft, social engineering, information exfiltration, and other assault strategies. Browsers and browser packages will be attacked by websites, SaaS apps, and unsanctioned apps.

Browser protection platforms offer comprehensive protection because they guide each inbound facts protection and outbound information protection. They cannot get replaced with the aid of networking and endpoint security answers. Browser safety systems consciousness on live web session hobby, i.e. the actual non-encrypted internet page that the browser renders and shows. network and endpoint solutions, alternatively, best view the network traffic and technique execution components of the net consultation. CASBs are confined to sanctioned apps and their APIs.
Browser safety structures enforce secure internet browsing and browser utilization across the staff without negatively impacting user productivity or the person reveling in. They had evolved as a reaction to the safety desires of the cloud-first and hybrid corporations that rely on the browser as their center working interface.

The Most Common Web Security Vulnerabilities
Website security vulnerabilities can vary depending on various factors such as the technology stack, programming languages, frameworks, and the specific implementation of a website. However, here are some of the most common website security vulnerabilities:

1. **Cross-Site Scripting (XSS)**: XSS occurs when an attacker injects malicious scripts into a website, which are then executed by the users' browsers. This vulnerability allows attackers to steal sensitive information, such as login credentials or session cookies, or even control the user's interactions with the website.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205436 |

2. **SQL Injection**: SQL injection happens when an attacker injects malicious SQL code into a vulnerable application's database query. This can lead to unauthorized access, data manipulation, or even the complete compromise of the database.

3. **Cross-Site Request Forgery (CSRF):** CSRF occurs when an attacker tricks a user's browser into making a malicious request on behalf of the user, typically targeting websites where the user is authenticated. This can lead to actions being performed without the user's consent, such as changing passwords or making financial transactions.

4. **Insecure Direct Object References (IDOR):** IDOR vulnerabilities arise when a website exposes internal implementation details, such as database keys or file paths, in its URLs or parameters. Attackers can manipulate these references to access unauthorized resources or perform unauthorized actions.

5. **Server-Side Request Forgery (SSRF):** SSRF occurs when an attacker can make a server perform requests to other internal or external resources on their behalf. This can be used to access sensitive information, attack internal systems, or perform remote code execution.

6. **File Inclusion Vulnerabilities**: These vulnerabilities occur when a website allows users to include files or execute code dynamically without proper validation. Attackers can exploit this to include malicious files or execute arbitrary code on the server.

7. **Remote Code Execution (RCE**): RCE vulnerabilities allow an attacker to execute arbitrary code on the server. This can lead to a complete compromise of the system, data theft, or unauthorized access to other parts of the infrastructure.

To ensure website security, it is crucial to follow secure coding practices, conduct regular security assessments, implement strong access controls, keep software and systems up to date, and educate developers about common vulnerabilities and secure development practices.

II.LITERATURE SURVEY

This researchpaper aims to discuss the security and privacy issues when dealing with web browsers, both generally and specific to seven of the most popular web browsers. Scholarly articles and research reports were referenced to provide us with a foundation to build on research and support findings from the data collection of the web browser user survey. Web browsers are applications that are used to access the internet [2]. Web browser users must consider the security and privacy of the web browser(s) that they choose to use. Privacy is how a user chooses to have their personal information used and controlled. Security on the other hand is how personal information is protected while using the services the web browser provides. Web browsers work hard to protect their users from hackers, businesses, websites, advertisers, internet service providers, and government agencies. You are being tracked by each click that you make, information is being sent to third parties, and you are being exposed to malicious or annoying ads, among many other threats [1]. The information that you have stored in your web browser and the personal data that you are passing through websites on the web browsers is what must be protected. In the world that we live in today, 69% of Americans have purchased something online which means they participated in online shopping which requires them to put in credit card information and log into their banking and insurance portals/websites, which, if hacked while doing so, opens pandora's box [3]. There are a multitude of web browser security vulnerabilities that can be exploited and interrupt the user's browser experience. The top four common web browser vulnerabilities mentioned by Gergely Kalman, a chief technology officer (CTO) are injection flaws, broken authentication, cross-site scripting (XSS), and insecure direct object references. The first vulnerability, injection flaws, is one, not many daily users are familiar with. Injection flaws are an effect of a lack of filtering of untrusted input, and even in the case of trusted input, you can never be 100% sure [4]. Everything has to be considered untrusted input because if not then you are more at risk of being exploited and it is better to be safe rather than sorry. The next most common web vulnerability is broken authentication. Broken authentication happens when in the process of proving the identity of a user, an interruption occurs. An interruption is when an attacker can divert or capture the authentication methods that are put into place. Weaknesses that open the door for interruption could be the use of plain text, encrypted, or weakly hashed passwords, ineffective or missing multi-factor authentication, the allowance of fragile, default, or common passwords, and the allowance of weak credential recovery processes [5]. The next vulnerability is cross-site scripting (XSS). Cross-sites scripting happens when a hacker, or browser user with malicious intent uses a web application to send malicious code to another user's computer, or the introduction of harmful script while loading a web application [6]. The last of the top four common

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205436 |

web browser mistakes is having insecure direct object references (IDOR). Insecure direct object references can lead to vulnerabilities because users are provided with direct access to important items like files and database keys, and hackers can take the reference and cause damage [4].

III.METHODOLOGY

Website security vulnerabilities can vary depending on the specific technologies and code used in the development of the website. Here are some common vulnerabilities and examples of code snippets that demonstrate the issues:

1. **Cross-Site Scripting (XSS):**
XSS occurs when untrusted data is included in a web page without proper sanitization, allowing attackers to inject malicious scripts that are executed by users' browsers.

Example vulnerability:
```javascript
var userInput = "<script>alert('XSS Attack');</script>";
var output = "Welcome, " + userInput;
document.getElementById("output").innerHTML = output;
```

Example secure code:
```javascript
var userInput = "<script>alert('XSS Attack');</script>";
var output = "Welcome, " + encodeURIComponent(userInput);
document.getElementById("output").textContent = output;
```

2. **SQL Injection:**
SQL injection occurs when user-supplied data is not properly sanitized or validated, allowing an attacker to execute malicious SQL queries on the underlying database.

Example vulnerability:
```php
$username = $_POST['username'];
$password = $_POST['password'];
$query = "SELECT * FROM users WHERE username = '$username' AND password = '$password'";
```

Example secure code (using prepared statements):
```php
$username = $_POST['username'];
$password = $_POST['password'];
$stmt = $pdo->prepare("SELECT * FROM users WHERE username = ? AND password = ?");
$stmt->execute([$username, $password]);
```

3**. Cross-Site Request Forgery (CSRF):**
CSRF occurs when an attacker tricks a user into unknowingly submitting a malicious request on a trusted website where the user is authenticated.

Example vulnerability (without CSRF protection):
```html
<form action="https://example.com/account/delete" method="POST">
<!-- Malicious site -->
<input type="hidden" name="delete" value="1">
<input type="submit" value=" Click here to claim your prize!">
</form>
```

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205436 |

```

Example secure code (with CSRF protection):
```html
<form action="https://example.com/account/delete" method="POST">
<!-- CSRF token -->
<input type="hidden" name="csrf_token" value="csrf_token_value">
<input type="hidden" name="delete" value="1">
<input type="submit" value="Delete Account">
</form>
```

4. **Insecure Direct Object References (IDOR):**
IDOR occurs when an application exposes internal or sensitive objects (e.g., database records or files) through insecure or predictable URLs or parameters.

Example vulnerability:
```php
$userId = $_GET['id'];
$query = "SELECT * FROM users WHERE id = $userId";
```

Example secure code:
```php
$userId = $_GET['id'];
$stmt = $pdo->prepare("SELECT * FROM users WHERE id = ?");
$stmt->execute([$userId]);
```

These are just a few examples of common website security vulnerabilities and how code can be both vulnerable and secure. It's important to implement secure coding practices, use proper input validation and sanitization techniques, and follow security guidelines and best practices specific to the programming languages and frameworks you are using. Regular code reviews and security testing are also recommended to identify and address vulnerabilities in your website's code.

IV.ADVANTAGES

Browser security solutions provide multiple benefits to the enterprise's security teams and strategy, across security, user experience, and productivity requirements. These include:

- **Improved Resiliency**
  Protecting the enterprise from a wide scope of relevant web-borne threats, browsing risks, and insider threats. These include phishing, malware, malicious extensions, screen capture, and other types of attacks that target the browser or attempt to utilize it as an attack vector.

- **Secure Access Management**
  Enforcing principles and policies for authentication, identity mapping, and more. Enforcing these principles means the web browser security platform acts as an authentication factor for initial access to SaaS apps and enables access only through the platform, and as an authorization layer to enforce least privilege principles.

- **Third Party Security**
  Enforcing browsing security for supply chain players. This benefit ensures secure access to corporate apps even for unmanaged devices and prevents data exfiltration.
```

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205436 |

- **Securing All Devices**
  Browser security platforms are device agnostic and secure for both managed and unmanaged devices. Security is based on the monitoring and analysis of web session security and policy enforcement and can be applied for employees, contractors, vendors (and more) for managed devices, BYOD, or third-party unmanaged devices.

V.DISADVANTAGES

While web browsers play a crucial role in ensuring online security, they are not without their disadvantages. Here are some disadvantages of browser security:

- **Vulnerabilities**: Despite continuous efforts to enhance security, web browsers can still have vulnerabilities that cybercriminals exploit. Zero-day vulnerabilities, for example, are previously unknown flaws that can be targeted by attackers before they are patched.

- **Phishing Attacks**: Browsers may struggle to effectively detect and block sophisticated phishing attacks. Phishing involves tricking users into providing sensitive information, such as login credentials or financial details, by impersonating trustworthy websites or entities. Even with security features, some phishing attempts can go undetected.

- **Add-Ons and Extensions**: While browser extensions and add-ons offer additional functionality, they can also introduce security risks. Malicious or poorly coded extensions may compromise user privacy and security, as they can access browsing data, inject unwanted advertisements, or even serve as a gateway for malware.

- **Incompatibility**: Browser security measures can sometimes be incompatible with certain websites or web applications. As a result, users may need to disable certain security features or use alternative browsers to access specific content, potentially exposing themselves to security risks.

- **Exploitation of Browser Features**: Certain browser features, such as JavaScript or cookies, can be exploited by malicious actors. Cross-site scripting (XSS) attacks, for instance, take advantage of vulnerabilities in JavaScript code to execute unauthorized scripts on web pages, potentially compromising user data.

VI.CONCLUSION

This research paper provides a complete survey of current research results under web application security. We have covered all properties of web application development, understood the important securityfunctions and properties that secure web applications should use, and divided existing works into three major classes. We also discuss a few issues that still need to be considered.

To access a few out-of-the-box features in web applications various programming concepts and tools are taking place that cause essential securityaspects to our applications. Apart from this security researchers apply the required efforts to extend security features toweb applications with several tools and techniques.

Generally, our logic and crucial codes resident the client side which is our browser that exposes programmer concepts. Thusfor attackers, it becomes easy to intercept the logic and cause total damage to the server-side state of the application.

REFERENCES

[1]"Most Popular Web Browsers between 1995 and 2019 via Data is Beautiful," 4 September 2020. [Online]. Available: https://igguru.net/2020/09/04/mostpopular-web-browsers-between-1995-and-2019-viadata-is-beautiful/. [Accessed December 2021].
[2] D. Bodnar, "What Is a Web Browser? " 28 January 2021. [Online]. Available: https://www.avast.com/cwhat-is-a-web-browser#gref. [Accessed December 2021].
[3] M. Djordjevic, "26 Useful Statistics on Online Shopping vs in Store Shopping [The 2021 Edition]," 19 November 2021. [Online]. Available: https://savemycent.com/statistics-on-onlineshopping-vs-in-store-shopping/. [Accessed December 2021].

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205436 |

[4] G. Kalman, "10 Most Common Web Security Vulnerabilities," 29 May 2014. [Online]. Available: https://www.toptal.com/security/10-most-commonweb-security-vulnerabilities. [Accessed December 2021].
[5] D. Blazquez, "Broken Authentication," 9 October 2021. [Online]. Available: https://hdivsecurity.com/owasp-brokenauthentication. [Accessed December 2021].
[6] G. E. Rodriguez, J. G. Torres, P. Flores, and D. E. Benavides, "Cross-site scripting (XSS) attacks and mitigation: A survey," Computer Networks, vol. 166, 15 January 2020.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.423

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details