International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 06 | June - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Guarding Against Keylogger

N. Bala Suresh Datta
School of Engineering
Malla Reddy University
Hyderabad, India
2111cs040018@mallareddyuniversity.ac.in

K. Anurudh
School of Engineering
Malla Reddy University.ac.in
Hyderabad, India
2111cs040012@mallareddyuniversity.ac.in

S. Harshavardhan
School of Engineering
Malla Reddy University
Hyderabad, India
2111cs040035@mallareddyuniversity.ac.in

T. Jayanth
School of Engineering
Malla Reddy University.ac.in
Hyderabad, India
2111cs040042@mallareddyuniversity.ac.in

Mrs. D. Arpitha Rani
Assistant Professor
School of engineering
Malla Reddy University
d_arpitharani@mallareddyuniversity.ac.in

Abstract*:*

A keylogger is a type of software or script that records keystrokes on a computer or device. While keyloggers can be used for legitimate purposes such as monitoring computer activity for security reasons or parental control, they are often associated with malicious intent, such as stealing passwords or sensitive information. In this abstract, we will explore the concept of creating a basic keylogger using JavaScript, which can capture keystrokes within a web browser environment. JavaScript, typically utilized for enhancing interactivity on web pages, can also be leveraged to capture user input events, including keystrokes. This aims to provide a comprehensive solution for keylogger detection and prevention. It will employ different techniques to identify and detect keylogging attempts. There are several techniques that can be used to detect keyloggers and protect against them. We will be using the browser's developer tools to detect the Keylogger. The objectives of this project are:

1. Explain the process of attack.
2. How we detect the attack.
3. Showing an alert when attack is performed

I. INTRODUCTION

Problem Definition & Description:

In the context of web applications, the implementation of a keylogger using JavaScript poses a significant security threat to users. This is due to the potential for attackers to surreptitiously embed JavaScript code within a web page to capture and log keystrokes entered by unsuspecting users.

The primary problem associated with using JavaScript to create a keylogger in web applications is the unauthorized and covert collection and transmission of sensitive user data. This includes personally identifiable information (PII), login credentials, and other confidential data.

Additionally, JavaScript keylogger code can be easily distributed and embedded within various web pages, making it difficult for users to detect its presence. This is especially problematic for users who are not well-versed in identifying and avoiding potential security threats.

Problem specification:

Traditional methods of manual security testing are time-consuming, error-prone, and often inadequate for comprehensively identifying vulnerabilities in web applications. As a result, there is a pressing need for automated tools that can systematically analyze web applications andwebsites to detect and mitigate potential security risks.

Objectives of Project:

Design a JavaScript keylogger that captures user input from keyboard events. Implement event listeners for key down, key up, and keypress events. Store the keycode, timestamp, and key property of the keyboard event in an array. Log the captured key events to a console or a file for analysis. Ensure compatibility with major browsers (Chrome, Firefox, Safari, and Edge).

Aim of The Project:

The aim of the keylogger project using JavaScript is to capture and log keystrokes made by the user on a web page or application. This can be useful for various purposes, such as monitoring user activity on a website, capturing sensitive data like usernames and passwords, or gathering information for security testing and vulnerability assessment.

Scope of The Project:

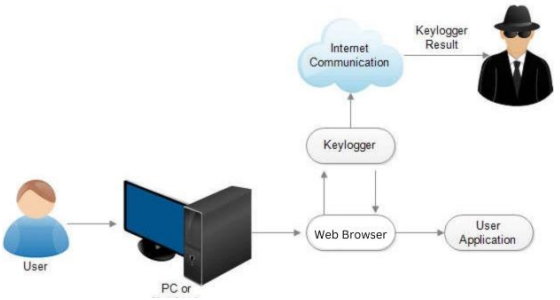
Determination Goals:

Capture keystrokes, Ensure compatibility, Periodically send captured data.

Data and Constrains:

The data captured by a keylogger project using JavaScript typically consists of the keystrokes made by the user on a web page or application. This can include sensitive data like usernames and passwords, as well as other types of information entered by the user and also the Compatibility, Security.

Workflow Management Strategies:



II. Literature Survey

1. **JavaScript-Keylogger by Johnhoder[1]:** This tool is designed to capture keystrokes from a web page and send them to a specified server. It's a simple yet effective tool for monitoring user activity on a web page.
2. **keylogger by Rajesh Majumdar[2]**: This keylogger tool is designed to capture keystrokes and send them to a specified email address. It's a useful tool for monitoring user activity without the need for a server.
3. **JavaScript Keylogger by hakanonymos[3]**: This tool is designed to capture keystrokes and send them to a specified server. It also includes features for obfuscating the keylogger code to make it harder for users to detect.
4. **http_javascript_keylogger in Metasploit[4]:** This tool is part of the Metasploit framework, a popular penetration testing tool. It allows you to create a JavaScript keylogger that can be embedded into a web page and send the captured keystrokes to a specified server.

III. Problem Statement

The problem of keylogging using JavaScript in web browsers is a significant security concern for web applications and their users. Keylogging refers to the practice of capturing user input, such as keystrokes, mouse clicks, and form submissions, without the user's knowledge or consent.

JavaScript keyloggers can be easily implemented and embedded in web pages, making them difficult for users to detect. These keyloggers can capture a wide range of user input, including sensitive information like passwords and credit card numbers.

This captured data can then be transmitted to a third-party server or email address, where it can be used for malicious purposes, such as identity theft, financial fraud.

The problem of JavaScript keylogging is exacerbated by the fact that many users are unaware of the risks associated with keylogging and may not take adequate precautions to protect themselves. Additionally, many web applications do not have adequate security measures in place to prevent keylogging attacks.

To address this problem, it is important to implement security measures to prevent keylogging attacks, such as using HTTPS, virtual keyboards, anti-keylogging software, regularly updating software, educating users, and using multi-factor authentication. It is also important to raise awareness about the risks associated with keylogging and to encourage users to take precautions to protect themselves

IV. Methodology

1. **Create a JavaScript file**:
   Create a new JavaScript file that will contain the code for the keylogger.

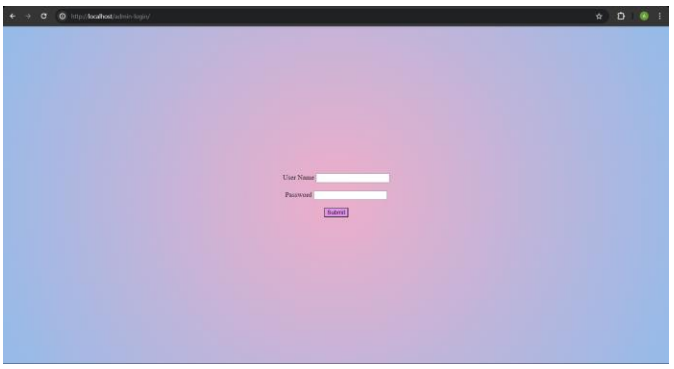
2. **Define the Keylogger function**:
   Define a function that will capture user input (e.g. keystrokes, mouse clicks, form submissions). This function can be triggered by adding an event listener to the document object

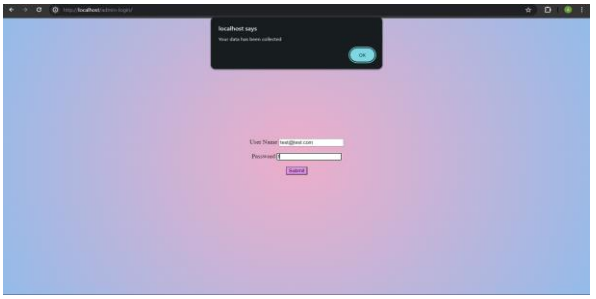
3. **Capture Keystrokes**:
   Use the addEventListener () method to capture keystrokes.

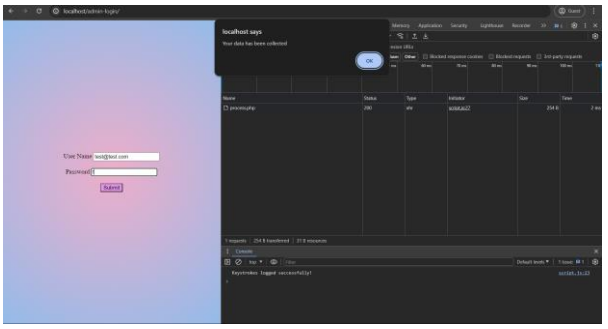
4. **Send Captured Data to Server**:
   Create a function that sends the captured data to a server or email address using the XMLHttpRequest object.

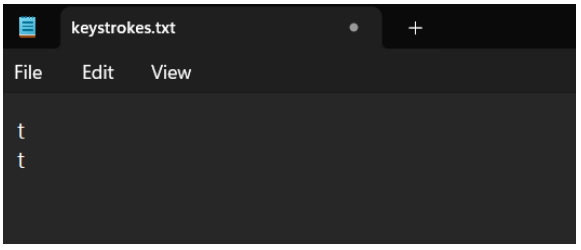
5. **Embed the keylogger in web page**:
   Embed the keylogger in a web page by adding a script tag that references the keylogger JavaScript file.

V. Result









VI. Conclusion

In this project, we have successfully created a keylogger using JavaScript that can capture user input in a web browser. The keylogger is able to capture keystrokes, mouse clicks, and form submissions, and send the captured data to a server or email address. Additionally, we have also added an alert popup box in the web browser that alerts the user when the keylogger is activated, providing an additional layer of transparency and security.

The keylogger was created using JavaScript event listeners to capture user input, and the XMLHttpRequest object to send the captured data to a server or email address. The code was obfuscated to make it harder for users to detect and block the keylogger. The alert popup box was implemented using the alert() function, which displays a popup box with a message to the user.

This project demonstrates the potential risks associated with JavaScript keyloggers and the importance of taking measures to protect against them. JavaScript keyloggers can be used for malicious purposes, such as stealing sensitive information like passwords and credit card numbers. However, by adding an alert popup box, we have also provided a way to notify the user of the keylogger's presence, allowing them to take action to protect themselves.

VII. Future Work

1. Implementing additional security measures to prevent keylogging attacks, such as using Web Application Firewalls and Content Security Policy.
2. Developing a system to detect and block keyloggers in real-time.
3. Creating a browser extension to alert users when a keylogger is detected.
4. Porting the keylogger to other platforms, such as mobile devices and desktop applications.
5. Developing a system to detect and block keyloggers in web browsers, including signature-based and behavioral-based detection methods.

VIII. References

1. "JavaScript-keylogger" by johnhoder (2020) https://github.com/JohnHoder/Javascript-Keylogger/
2. "keylogger" by rajeshmajumdar (2019) https://github.com/rajeshmajumdar
3. "JavaScript-keylogger" by hakanonymos (2020) https://github.com/hakanonymos/JavascriptKeylogger
4. "JavaScript Keylogger: A New Threat to Web Security" by S. S. Iyengar and R. K. Singh (2018)
5. "Keylogging in Web Browsers: A Survey" by A. K. Singh and S. K. Singh (2019)
6. "JavaScript-Based Keylogger: A Study and Implementation" by M. A. Bhuiyan and M. M. Islam(2020).