Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

Crypto Jacking

[1]Mohammed Nadeem Shareef, [2]Junaid Hussain [3]Mohammed Khaja Adnan Ali Khan,

[4]Dr. Mohammed Abdul Bari

[1]BE Student – Dept. Of CSE, ISL Engineering College, TS, India

[2]BE Student – Dept. Of CSE, ISL Engineering College, TS, India

[3]BE Student – Dept. Of CSE, ISL Engineering College, TS, India

[4]Professor – Dept. Of CSE, ISL Engineering College, TS, India

Article Info

Page Number: 1581 - 1586

Publication Issue:

Vol 72 No. 1 (2023)

Abstract

The aim of this project is to develop a Chrome extension that can detect crypto-jacking attempts in real-time and block the websites containing such attempts. Crypto-jacking is the unauthorized use of someone's computing resources to mine cryptocurrencies. This type of attack can go undetected for long periods and cause damage to the user's device, such as overheating and reduced battery life. To detect crypto-jacking, the extension will use the blacklisting-based approach combined with a resources monitoring approach. This approach uses a blacklist of known crypto-jacking scripts and domains frequently associated with crypto-jacking. If the extension detects any of these scripts or domains on a website, it will block the website from opening in the first place. This approach prevents crypto-jacking attempts by blocking websites with known crypto-jacking codes.

Overall, this project is crucial to protect users from the increasing prevalence of crypto-jacking attacks. By blocking known crypto-jacking websites, the extension will prevent attackers from using the user's computing resources for their gain.

Article History

Article Received: *15 October 2022*

Revised: *24 November 2022*

Accepted: *18 December 2022*

Keywords: Extension, crypto-jacking, cryptocurrencies, blacklisting-based.

1. Introduction

1.1 Introduction

Crypto jacking has become a serious threat in the modern web environment, where malicious script code is automatically executed on the client side without the user's consent or knowledge.[1] The attackers use the victims' computer resources, such as CPU and memory, for cryptocurrency mining, which generates profit for them[2]. This type of cyberattack is known as cryptojacking, and

it can cause significant damage to users' devices, resulting in slower performance and potential hardware damage[1][12].

To protect users from cryptojacking attacks, it is essential to detect and filter out cryptojacking websites in the web environment. However, this task is becoming increasingly challenging due to the use of script code obfuscation techniques[3][11], which makes it harder to detect cryptojacking websites based on static analysis. Moreover, the characteristics of cryptojacking websites, such as high resource consumption and numerous threads, now appear on normal websites, making it more difficult to detect them accurately. Existing approaches for detecting cryptojacking websites use various techniques, such as blacklisting-based[4][13], resource monitoring-based [7][15], thread count-based [14][8], and WebAssembly-based approaches [9]. However, these approaches have limitations in terms of the precise detection of cryptojacking. For example, blacklisting-based approaches can be easily bypassed by script code obfuscation [19][3] or a domain generation algorithm, while resource-monitoring-based [7][15] and thread count-based [8] approaches yield numerous false positives. WebAssembly-based [9] approaches have low detection coverage and cannot detect the most common JavaScript-based cryptojacking websites.

Therefore, there is a need for an effective and accurate approach to detecting cryptojacking websites. Our project aims to address this challenge by developing a Chrome extension that uses a blacklisting-based approach [4][20] to detect and block cryptojacking websites. This approach stores elements with unique cryptojacking characteristics as keywords in a blacklist and uses them to detect cryptojacking. If a website contains any of the stored keywords, it will be considered a cryptojacking website and will be blocked
.by the extension. By using this approach, we aim to provide users with a reliable and efficient way to protect their devices from cryptojacking attacks. [5]

1.2 Scope of the Project

The project's primary goal is to protect users from the increasing prevalence of cryptojacking attacks by preventing attackers from using the user's computing resources for their gain.[6] The project will use a blacklisting-based approach that stores elements with unique cryptojacking characteristics as keywords in a blacklist and uses them to detect cryptojacking [4].

2. Literature Review

Cryptojacking is a type of cyberattack where the attacker uses a victim's computer resources, such as CPU and memory, to mine cryptocurrency without the victim's consent. This can be very profitable for the attacker, who can use the cryptocurrency for their own benefit. In the past, cryptojacking was executed by tricking users into downloading and running malicious programs, similar to other types of malware. However, more recently, a more dangerous form of cryptojacking has emerged, where malicious scripts are automatically executed on a user's computer when they visit a cryptojacking website [10][18].

Detecting and blocking cryptojacking websites is essential to protect users' resources. However, it can be difficult to accurately detect these sites because of the use of script obfuscation techniques, which make it harder to identify malicious code through static analysis. Additionally, many normal websites also use a lot of computer resources, such as live-streaming sites, which can make it difficult to differentiate between legitimate and malicious sites.

Existing detection techniques for cryptojacking mainly fall into four categories: blacklisting-based, resource monitoring-based, thread count-based, and WebAssembly-based. While all of these techniques can provide some level of detection, they all have limitations in terms of accurately detecting cryptojacking. For example, blacklisting-based approaches can be easily bypassed by evasion techniques such as script obfuscation or domain generation algorithms. Resource monitoring and thread count-based approaches can produce a lot of false positives, as many legitimate websites also use a lot of resources. Finally, WebAssembly-based approaches have limited coverage and cannot detect the most common JavaScript-based cryptojacking sites.

The authors of this study have proposed a new approach for detecting cryptojacking websites called CIRCUIT. They focused on the JavaScript memory heap of websites to detect reference flows, which represent cryptojacking behavior and are resistant to script code obfuscation. By comparing the reference flows of a website to known cryptojacking signatures, CIRCUIT can accurately identify whether a website is running cryptojacking [10][16].

In their experiment, the authors collected over 300K real-world websites and found 1,813 cryptojacking websites using CIRCUIT, even though most of these sites used evasion techniques to avoid detection. The authors also modeled these evasion techniques to gain new insights into cryptojacking and demonstrated the limitations of existing resource monitoring and thread-count-based approaches.

Overall, I found this study to be very interesting and informative. The authors' new approach seems very effective in detecting cryptojacking, and their analysis of evasion techniques and their impact on cryptojacking detection provides valuable insights into this growing issue [10].

3. Proposed System

3.1 Proposed System Overview

The proposed system aims to provide a comprehensive solution for detecting and preventing cryptojacking attacks on websites. To achieve this goal, the system utilizes two different methods, namely the Blacklisting-based approach and the Resource monitoring-based approach, and combines them into a single solution.

The Blacklisting-based approach involves maintaining a blacklist of known cryptojacking domains and script codes that are associated with such attacks. Whenever a user visits a website, the system checks if the website's domain or scripts match any of the entries in the blacklist. If a match is found, the website is flagged as potentially malicious and blocked.

The Resource monitoring-based approach, on the other hand, relies on monitoring the computer resources (such as CPU usage) consumed by a website when a user visits it. If the resource usage exceeds a predetermined threshold, the website is flagged as potentially malicious and blocked.

4. Implementation

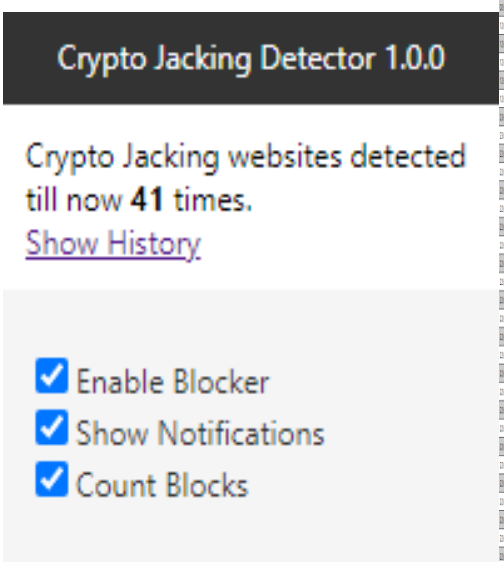

Fig 1: **Default Pop Up**

Fig 2: History Page

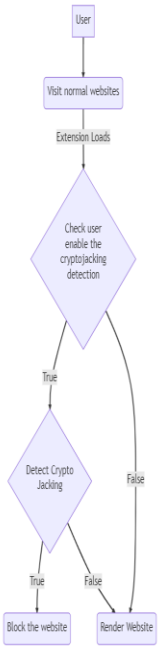

Fig 1: Architecture

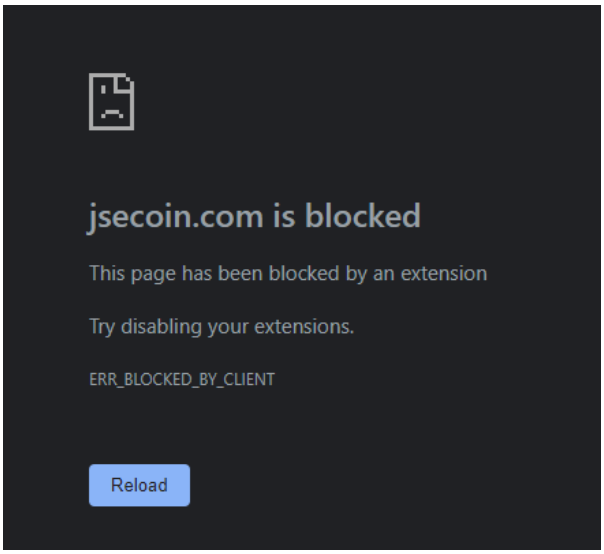

Fig 4: Blocked Website

Conclusion and Future Enhancement

In conclusion, cryptojacking is a growing threat to internet users, and several detection methods have been proposed to counter this threat. Existing detection methods include blacklisting-based, resource monitoring-based, thread count-based, and WebAssembly-based approaches. However, each of these methods has limitations that make them less effective in detecting cryptojacking websites. To address these limitations, we propose a system that combines the advantages of blacklisting-based and resource-monitoring-based approaches. Our proposed system involves building a browser extension that protects users from cryptojacking websites by detecting and blocking them. We believe that this proposed system will provide a more effective and efficient way to protect users from cryptojacking.

In the future, our proposed system can be enhanced by incorporating additional detection methods to improve its effectiveness in identifying cryptojacking attacks. For instance, machine learning algorithms can be integrated to identify patterns in web traffic and detect anomalies that may indicate cryptojacking. Additionally, integrating blockchain technology can provide an immutable and decentralized way to track and verify cryptocurrency transactions associated with cryptojacking. This can aid in identifying the source of the attack and possibly preventing future attacks. Moreover, the system can be expanded to support multiple browsers and platforms to reach

a wider user base. Finally, regular updates to the blacklist and resource thresholds can be implemented to ensure that the system is up-to-date and effective in detecting new forms of cryptojacking attacks.

Reference

[1].https://blogs.quickheal.com/cryptojacking-on-the-rise/

[2].https://thenewstack.io/cryptojacking-free-money-for-attackers-huge-cloud-bill-for-you/

[3].https://dl.acm.org/doi/pdf/10.1145/3507657.3528560

[4]. 1lastBr3ath/drmine: Dr. Mine is a node script written to aid automatic detection of in-browser cryptojacking. (github.com)

[5].https://zvelo.com/cryptojacking-infection-methods-identification-prevention-tips/

[6].https://www.varonis.com/blog/cryptojacking

[7]. G. Hong, Z. Yang, S. Yang, L. Zhang, Y. Nan, Z. Zhang, M. Yang, Y. Zhang, 979 Z. Qian, and H. Duan, ‘‘How you get shot in the back: A systematical 980

[18].Dr. M.A.Bari, ”EffectiveIDS To Mitigate The Packet Dropping Nodes From Manet “, JACE, Vol -6,Issue -6,June 2019

[19].M.A.Bari & Shahanawaj Ahamad,” Process of Reverse Engineering of Enterprise InformationSystem Architecture” in International Journal of Computer Science Issues (IJCSI), Vol 8, Issue 5, ISSN: 1694-0814, pp:359-365,Mahebourg ,Republic of Mauritius , September 2011

[20]. M.A.Bari & Shahanawaj Ahamad, “Code Cloning: The Analysis, Detection and Removal”, in International Journal of Computer Applications(IJCA),ISSN:0975-887, ISBN:978-93-80749-18-3,Vol:20,No:7,pp:34-38,NewYork,U.S.A.,April 2011

[21] Agrawal, S. A., Umbarkar, A. M., Sherie, N. P., Dharme, A. M., & Dhabliya, D. (2021). Statistical study of mechanical properties for corn fiber with reinforced of polypropylene fiber matrix composite. Materials Today: Proceedings, doi:10.1016/j.matpr.2020.12.1072

[22] Anupong, W., Azhagumurugan, R., Sahay, K. B., Dhabliya, D., Kumar, R., & Vijendra Babu, D. (2022). Towards a high precision in AMI-based smart meters and new technologies in the smart grid. Sustainable Computing: Informatics and Systems, 35 doi:10.1016/j.suscom.2022.100690