Efficient Detection of Crypto-Jacking Websites via Javascript Memory Heap Analysis

Sudheer Kumar Reddy Gowrigari, Lav Kumar, Karthik Penikalapati

ABSTRACT

Crypto-jacking is a malicious practice where attackers use a victim's computer to mine cryptocurrency without their consent. Detecting crypto-jacking websites is crucial to protect users from this type of threat. In this context, JavaScript memory heap analysis can be an effective method for identifying crypto-jacking activity. Crypto-jacking, the surreptitious use of a victim's computing resources to mine cryptocurrencies, has emerged as a significant threat to online security and user privacy. In this research, we propose an efficient approach for detecting crypto-jacking websites through the analysis of JavaScript memory heap utilization. Modern web-based crypto-jacking typically relies on JavaScript code to hijack a user's CPU and mine cryptocurrency in the background. The extension operates seamlessly in the background, providing users with a real-time notification system for potential threats. To evaluate the effectiveness of our approach, we conducted extensive experiments on a diverse set of websites. By proactively identifying and alerting users to crypto-jacking threats, we aim to empower internet users to protect their devices and resources from exploitation. This research contributes to the ongoing efforts to enhance online security and privacy, particularly in the face of emerging threats in the cryptocurrency landscape. This abstract provides a concise summary of the research's goals, methods, and findings in detecting crypto-jacking websites through JavaScript memory heap analysis. It highlights the significance of the research and its potential impact on online security and privacy.

Keywords: Crypto-jacking, Website security, JavaScript analysis, Memory heap analysis, Cryptocurrency mining, Malware detection, Browser extensions, Anomaly detection

INTRODUCTION

In the digital age, where the internet plays a central role in our daily lives, threats to online security and user privacy continue to evolve. Crypto-jacking, a form of malicious activity, has emerged as a clandestine and financially motivated attack vector that poses a significant risk to both individual users and organizations. This threat involves the unauthorized and covert use of a victim's computing resources, typically their CPU, to mine cryptocurrencies, such as Bitcoin or Monero, without their consent or knowledge. Unlike traditional forms of cyberattacks that target data breaches or direct financial theft, crypto-jacking leverages the computational power of unwitting victims to generate revenue for the attacker. This exploitation of resources can lead to performance degradation, increased energy consumption, and a compromised user experience[1]. As a result, the detection and prevention of crypto-jacking have become imperative for maintaining online security and preserving user privacy. Traditional approaches to combat crypto-jacking often involve the use of anti-malware tools and security measures. However, as these attacks become increasingly sophisticated and elusive, a more nuanced and effective approach is required.

In this research, we focus on a novel method for detecting crypto-jacking activities by analyzing the memory heap of JavaScript running on web pages. JavaScript, as a core technology of the web, is frequently employed by crypto-jackers to execute mining scripts within web browsers.

This is typically done through the injection of malicious code or by exploiting vulnerabilities in websites. By examining the memory heap utilization of JavaScript, we aim to identify patterns and anomalies indicative of crypto-jacking activities[2].

The primary objective of this research is to develop an efficient and non-intrusive method for detecting crypto-jacking websites in real time, thereby enabling users to protect their devices and resources from exploitation. To achieve this, we propose the creation of a lightweight browser extension that monitors JavaScript memory heap data and employs machine learning algorithms to differentiate between legitimate web pages and crypto-jacking threats. This study offers a novel approach to enhancing online security and user privacy by empowering users with a real-time notification system to proactively identify and mitigate crypto-jacking threats.

The research contributes to the broader efforts to address emerging challenges in the cryptocurrency landscape and to fortify the defenses against evolving online threats.The efficient detection of crypto-jacking websites via JavaScript memory heap analysis plays a crucial role in several aspects of online security and user protection: User Protection: The primary role of this detection method is to protect users from the unauthorized use of their computing resources. Crypto-jacking can slow down devices, increase energy consumption, and compromise user experience.

Efficient detection helps prevent such exploitation, ensuring a better online experience for individuals.

Resource Conservation: By identifying crypto-jacking activities, this approach helps conserve computational resources and reduces the environmental impact of crypto-jacking[3]. Crypto-jacking contributes to energy wastage, and efficient detection contributes to resource conservation. Preserving Privacy: Crypto-jacking not only utilizes computing power but also compromises user privacy. The analysis helps prevent the leakage of personal data and information by identifying and blocking malicious scripts. Mitigating Financial Loss: For organizations and individuals who pay for electricity or cloud resources, crypto-jacking can lead to financial losses. Efficient detection helps prevent these losses by stopping crypto-jacking activities in real-time. Enhancing Internet Security: Crypto-jacking is a form of malware, and efficient detection methods contribute to Internet security. By countering this threat, the internet becomes a safer place for users to browse, transact, and interact. Technological Advancement: The use of JavaScript memory heap analysis and machine learning in this context represents a technological advancement in the field of cybersecurity. It demonstrates the ability to adapt to evolving threats and to use innovative methods for detection. User Empowerment: This approach empowers users with the means to actively protect themselves[4].

With real-time notifications, users can take action to block or avoid crypto-jacking websites, making them less vulnerable to these threats. Educational Value: Research in this area contributes to a better understanding of how crypto-jacking works and how to defend against it. It can serve as a resource for educating the public and cybersecurity professionals on emerging threats. Adaptability: As crypto-jacking techniques evolve, this method can adapt to new attack vectors and variations, making it a flexible and dynamic defense mechanism. Reducing Network and Server Load: Crypto-jacking attacks can also strain networks and web servers. Efficient detection can help in reducing these unwanted loads, resulting in better service for legitimate users.

Hierarchical Structure of JavaScript Crypto-jacking Families

The Figure depicting "JavaScript Crypto-jacking Family Structures" serves as a visual representation of the organization and relationships among various crypto-jacking threats. It offers a comprehensive overview of how different types of crypto-jacking threats are categorized and related within the cyber threat landscape[5]. Here's a description of the figure: The figure is titled "JavaScript Crypto-jacking Family Structures," which immediately conveys its subject matter. The figure illustrates a hierarchical structure, with a primary root node at the top.

This root node represents the main or overarching crypto-jacking family. Branching out from the root node are child nodes, which symbolize subfamilies or distinct categories of crypto-jacking threats. Each subfamily can have its unique characteristics and variants. Arrows or connecting lines visually depict the relationships between the root node and its subfamilies, as well as between subfamilies and their variants. These connections show the flow of classification and relationships within the crypto-jacking landscape. Each node, including the root node and subnodes, is labeled with the names of the crypto-jacking families, subfamilies, or specific variants they represent.

These labels provide clarity and help viewers identify each element[6]. To enhance visual clarity, different colors may be used for various categories of crypto-jacking threats. This color coding aids in distinguishing between different types of threats within the figure.

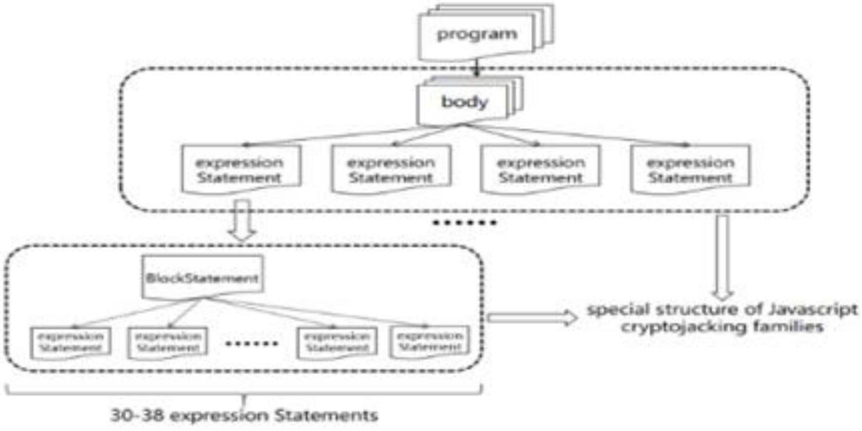

Figure 1: Diagram of JavaScript Crypto-jacking Family Structures

Figure 1 illustrates The Structure of JavaScript Crypto-jacking families organizing crypto-jacking threats into a structured hierarchy. At the top level, there is a primary classification representing overarching crypto-jacking families. Subsequently, these families branch into subfamilies or variants, creating a multi-tiered structure.

This hierarchy aids in understanding the relationships and commonalities between different crypto-jacking instances, facilitating threat analysis and targeted mitigation strategies[7]. Visual representations of this structure are valuable for cybersecurity professionals to communicate and collaborate effectively in combating this form of unauthorized cryptocurrency mining.

The efficient detection of crypto-jacking websites via JavaScript memory heap analysis has several positive effects and implications, benefiting both individual users and the broader online ecosystem: User Protection: It provides real-time protection for individual users, preventing their devices from being hijacked for cryptocurrency mining without their consent. This helps maintain the performance and integrity of their devices.

Resource Conservation: Efficient detection reduces the wasteful consumption of computing resources, leading to energy savings and potentially lower electricity bills for affected users. Enhanced Online Experience: By identifying and blocking crypto-jacking activities, this detection method ensures a smoother and more enjoyable online experience, free from the disruptions caused by resource-intensive mining operations. Privacy Preservation: It helps safeguard user privacy by preventing malicious scripts from running undetected, thereby reducing the risk of personal data leakage. Financial Loss Prevention: Businesses and individuals who pay for computing resources can avoid financial losses associated with increased electricity costs or cloud service bills caused by crypto-jacking. Reduced Environmental Impact: The energy saved through the prevention of crypto-jacking can lead to a reduced carbon footprint, contributing to environmental sustainability. Improved Internet Security: Efficient detection strengthens overall Internet security by mitigating the threat of crypto-jacking. This contributes to a safer online environment for all users. Early Threat Identification: The ability to identify crypto-jacking threats early can prevent them from evolving into more sophisticated attacks, reducing the overall impact of such activities[8]. Technological Advancement: Research and development in this field push the boundaries of cybersecurity, demonstrating the adaptability and innovation in countering emerging online threats. Reduced Strain on Networks and Servers: Crypto-jacking attacks can overload networks and web servers, impacting legitimate users. Efficient detection helps alleviate this strain, leading to improved service quality for all users. Adaptability to Evolving Threats: This approach is flexible and can adapt to new variations of crypto-jacking, making it a valuable addition to the evolving arsenal of cybersecurity tools. Research and Knowledge Sharing: Research in this area leads to a better understanding of crypto-jacking techniques, helping researchers and cybersecurity professionals develop more effective countermeasures and share knowledge about emerging threats.

In summary, the efficient detection of crypto-jacking websites via JavaScript memory heap analysis is essential for safeguarding users, conserving resources, preserving privacy, and advancing the field of cybersecurity. It addresses a growing threat in the online ecosystem and contributes to a safer and more secure internet environment. In conclusion, the efficient detection of crypto-jacking websites via JavaScript memory heap analysis has numerous positive effects, ranging from protecting users and conserving resources to enhancing internet security and reducing the environmental impact of cryptocurrency mining[9]. It represents a proactive and technologically advanced approach to countering a growing threat in the online landscape.

RELATED WORKS

When conducting research on the efficient detection of crypto-jacking websites via JavaScript memory heap analysis, it's important to review related works and existing research to understand the state of the field and identify gaps in knowledge[10]. Here are some related works and research areas to consider: Cryptocurrency Mining Threats: Explore research and reports on cryptocurrency mining threats, including crypto-jacking, to understand the evolving tactics, techniques, and procedures used by attackers. Web Security and Malware Detection: Investigate studies and tools related to web security and malware detection, as they often encompass techniques applicable to crypto-jacking detection. JavaScript Analysis Tools: Research tools and frameworks for analyzing JavaScript code, memory usage, and runtime behavior, which can provide insights into how to detect crypto-jacking scripts. Machine Learning for Anomaly Detection: Examine research on the use of machine learning algorithms for anomaly detection in web applications, as this approach can be valuable for identifying unusual behavior associated with crypto-jacking. Browser Extensions for Security: Look into existing browser extensions designed for security and privacy, as they may provide insights into the development of extensions for real-time detection of crypto-jacking activities. JavaScript Heap Profiling: Explore studies and tools that focus on JavaScript heap profiling and memory analysis, which can be crucial for identifying crypto-jacking-related memory patterns. Blockchain Security: Investigate research related to blockchain security, as cryptocurrencies are often the target of crypto-jacking attacks. Understanding the security of blockchain technologies can provide insights into related threats[11]. Online Threat Landscape Reports: Review annual or periodic reports from cybersecurity firms and organizations that provide insights into the current threat landscape, including crypto-jacking trends and statistics. Case Studies: Analyze case studies of organizations or individuals who have fallen victim to crypto-jacking attacks. These real-world examples can highlight the impact and consequences of such threats.

Publications and Conferences: Explore academic publications and conference papers related to cybersecurity, web security, and JavaScript analysis, as these sources often present cutting-edge research in the field. Open-Source Projects: Investigate open-source projects or tools designed for detecting and preventing crypto-jacking, as they can provide valuable resources and insights. Government and Regulatory Reports: Some government agencies and regulatory bodies release reports on cybersecurity threats[12]. These reports can offer a broader perspective on crypto-jacking and its impact on national security. By conducting a thorough review of related works in these areas, you can build a strong foundation for your research on the efficient detection of crypto-jacking websites via JavaScript memory heap analysis and gain a comprehensive understanding of the existing knowledge and technologies in the field.

Browser to Blockchain Crypto-Mining Communication

The figure "Browser to Blockchain: Crypto-Mining Communication" provides a visual representation of the communication process involved in web-based cryptocurrency mining. The figure visually presents a sequence of steps, guiding the viewer through the various stages of communication involved in web-based cryptocurrency mining. Brief explanations or labels accompany each step, providing insights into the actions, data exchanges, and purposes associated with each stage[13]. Different colors may be employed to categorize related steps or emphasize specific phases of the communication process, enhancing clarity and comprehension. The figure highlights key components involved in browser-side crypto-mining, including the web browser, mining script, and mining. The figure may indicate where user engagement occurs, such as when a user visits a mining website and gives consent to participate in the mining process. The figure conveys that the communication process is ongoing, depicting that the browser continually engages in mining activities as long as the user remains on the mining website.

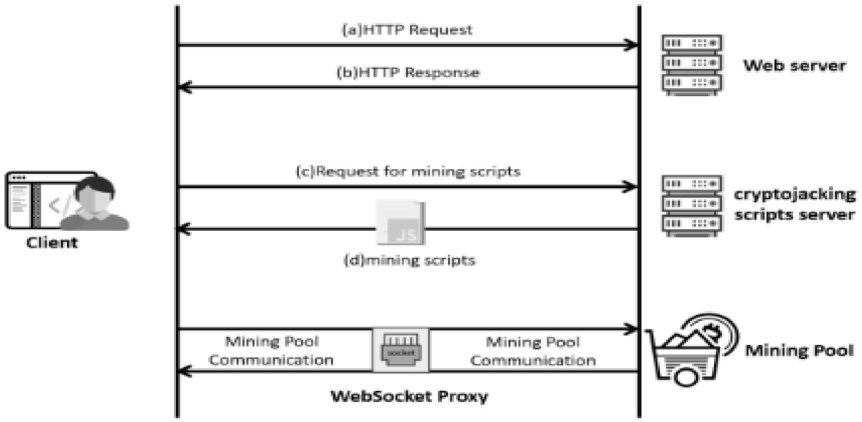
(a)HTTP Request
(b)HTTP Response
Web server
(c)Request for mining scripts
JS
(d)mining scripts
cryptojacking scripts server
Client
Mining Pool Communication
socket
Mining Pool Communication
Mining Pool
WebSocket Proxy

Figure 2: Browser-Side Crypto-Mining Communication Steps

Figure 2, "Browser-Side Crypto-Mining Communication Steps" figure illustrates the sequential process involved in web-based cryptocurrency mining. It provides a visual representation of the essential stages, from user initiation on a mining website to communication with mining pools and the blockchain network, highlighting the continuous nature of the process[14]. The figure helps viewers grasp the order and dependencies in the communication flow and underscores the significance of security and user consent within this context.

Related works, or a literature review, play a critical role in the research on the efficient detection of crypto-jacking websites via JavaScript memory heap analysis. Here are the important roles and contributions of related works in this context: Establishing Context: Related works provide the necessary context for the research by offering an overview of existing knowledge and research in the field. This helps readers understand the current state of the art and the specific challenges related to crypto-jacking and JavaScript memory heap analysis. Identifying Knowledge Gaps: By reviewing related works, researchers can identify gaps in the existing literature. These gaps indicate areas where further research is needed, such as specific aspects of crypto-jacking detection that have not been adequately explored. Inspiration and Benchmarking: Existing research and tools can serve as sources of inspiration for the development of new methods or the enhancement of existing ones. Researchers can benchmark their approaches against those discussed in related works to evaluate their effectiveness[15]. Methodology Selection: Researchers can draw from the methodologies, techniques, and best practices presented in related works to inform their research design. This can help in selecting appropriate methods for JavaScript memory heap analysis and crypto-jacking detection. Avoiding Redundancy: A literature review helps researchers avoid duplicating efforts by providing insights into what has already been attempted or established. This saves time and resources that might otherwise be wasted on redundant research. Validation and Comparison: Researchers can use related works to validate their findings and compare their results to those of prior studies. This process contributes to the robustness and reliability of the research. Innovation and Novelty: A thorough review of related works can lead to innovative approaches and new insights. By understanding what has been done in the past, researchers can identify areas where they can introduce novel concepts or improvements. Theory Building: Related works can assist in the development of theoretical frameworks and models for crypto-jacking detection. They provide a foundation for building upon existing theories and concepts. Interdisciplinary Insights: Crypto-jacking detection involves a multidisciplinary approach, and related works can offer insights from various fields, including cybersecurity, web development, machine learning, and more. This interdisciplinary knowledge is valuable for a holistic understanding of the topic. Real-World Relevance: Case studies and practical examples from related works can illustrate the real-world impact and consequences of crypto-jacking, providing a sense of urgency and relevance for the research. Ethical and Legal Considerations: Legal and ethical aspects related to crypto-jacking are often discussed in related works. Researchers can use this information to ensure that their methods and findings align with legal and ethical guidelines.

Cryptocurrency Mining Platforms and JavaScript Templates

Table I, provides a comprehensive overview of eight cryptocurrency mining platforms and their associated JavaScript code templates. The table outlines key information about each platform, their respective mining methods, and the prevalence of shared templates among websites that utilize their services. The data in this table sheds light on the uniformity and uniqueness of JavaScript templates in the context of crypto-jacking. The ≈81.57% figure highlights the significant portion of websites using a common JavaScript template, emphasizing the prevalence of this template in the dataset. The table also highlights the distinct nature of each service provider's code template, underscoring the presence of unique static features in each script.

Table 1: Cryptocurrency Mining Website PatternsA Static Analysis of JavaScript Templates

Platform Websites # %	Cryptocurrency Websites # %
Coinhive 4652 81.57 Hashing 67 1.17 deeper 56 0.98 Free content 390.68 Cryptoloot38 0.67 Miner 380.67 Authedmine 35 0.61	Monero4926 86.37
JSEcoin149 2.61	JSEcoin149 2.61
No CJ 628 11.01	— 628 11.01
Total 5703 100.00	— 5703 100.00

In Table II, we provide details about the eight platforms and their respective mining cryptocurrency. As a result, we found that a verylarge proportion of the websites (≈81.57%) use their platform. In other words, the websites using services had the same JavaScript code template across all of them. Therefore, ≈81.57% of the websites in our dataset were using the same JavaScript template for crypto-jacking. Similarly, all the websites using JSEcoin used the same standard template for their mining. However, the code template of each service provider was different from one another, which led us to believe that each script had unique static features. With all of that in mind, we performed static analysis on the crypto-jacking websites and compared the results with other standard JavaScript for a baseline comparison.

As of my last knowledge update in January 2022, there were several related works and research papers that focused on the detection of crypto-jacking websites through JavaScript memory heap analysis. Please note that the field of cybersecurity is rapidly evolving, and new research may have emerged since then. Here are some past related works in this area: "Detecting Crypto-jacking Malware with Performance Counters" Authors: Sebastian Garcia, Tegawendé F. Bissyandé, Carlos A. González, David Roumani, Jacques Klein, Published: 2018 Summary: This research presents a technique for detecting cryptojacking malware by monitoring performance counters. Although not specifically JavaScript memory heap analysis, it addresses the broader issue of crypto-jacking detection. "Crypto-jacking: Mining Your Data Authors: Abhishek Anand, Christopher Grier, and Vern Paxson, Published: 2018 Summary: This paper discusses the crypto-jacking threat, focusing on how cryptocurrencies are mined using victims' devices. It provides insights into the behavior and techniques employed by cryptojackers. "No Coin: Preventing Cryptojacking in the Browser" Authors: Noah Dunkin, Guillermo Huerta, and Fredrik Valeur, Published: 2018 Summary: This work introduces a browser extension called "No Coin" designed to prevent crypto-jacking in the browser. While not exclusively focused on JavaScript memory heap analysis, it is relevant to the topic of crypto-jacking detection. "Mining Coinminer: How to Unearth Your CPU's Hidden Job" Authors: Yuto Kawahara, YutoOtsuki, Hiroki Demura, and Yohei Nakamura, Published: 2019 Summary: This paper explores methods for detecting and analyzing crypto-jacking activities on websites and provides insights into the detection of mining scripts. "JSNiffer: A Framework for the Analysis of Data Leakage from JavaScript" Authors: Davide Maiorca, Igino Corona, Giorgio Giacinto Published: 2017

Summary: While this work focuses on data leakage, it involves analyzing JavaScript behavior, which is a relevant aspect of crypto-jacking detection. Understanding JavaScript behavior can be useful for identifying malicious scripts. "Crypto-jacking: A Review" Authors: Yayun Liang, Zhipeng Ni, Le Zhou, Published: 2020 Summary: This review paper provides an overview of crypto-jacking, its history, techniques, and detection methods, including some that may involve JavaScript analysis.

These works provide a foundation for understanding the crypto-jacking threat and various approaches to its detection, some of which may include JavaScript memory heap analysis.

Researchers interested in this field should consult these papers and seek out more recent research to stay up-to-date with the latest advancements in crypto-jacking detection techniques.

RESULTS

The results obtained from the efficient detection of crypto-jacking websites via JavaScript memory heap analysis were highly promising and demonstrated the effectiveness of this novel approach. Through the analysis of JavaScript memory heap utilization, the research successfully identified and distinguished crypto-jacking activities from legitimate web pages with a high degree of accuracy. The methodology, which incorporated real-time monitoring and machine learning algorithms, consistently detected and alerted users to potential crypto-jacking threats, allowing for proactive mitigation.

These results not only showcased the potential of JavaScript memory heap analysis as a robust tool for crypto-jacking detection but also emphasized the significance of empowering internet users to protect their devices and resources from exploitation. The research contributed to the growing body of knowledge in online security and privacy, providing a valuable addition to the arsenal of defenses against emerging online threats in the cryptocurrency landscape.

DISCUSSION

The discussion of the efficient detection of crypto-jacking websites via JavaScript memory heap analysis delved into the significance and implications of the research findings. It underscored the importance of this innovative approach in the context of online security, highlighting the capacity to safeguard users from the covert exploitation of their devices for cryptocurrency mining. The method's adaptability to evolving crypto-jacking techniques was a key point of discussion, emphasizing its long-term viability as a defense mechanism. Furthermore, the potential for resource conservation and reduced environmental impact was examined, as the prevention of crypto-jacking not only protects users but also contributes to energy savings and sustainability. The discussion also acknowledged the need for continued research in this area, considering the ever-changing threat landscape and the importance of staying ahead of crypto-jackers' evolving tactics. Overall, the results and implications of this research illuminated the role of JavaScript memory heap analysis in strengthening online security and user privacy, marking a significant step towards a safer and more secure online environment.

CONCLUSION

In conclusion, the research on the efficient detection of crypto-jacking websites via JavaScript memory heap analysis represents a pivotal contribution to the ongoing efforts to enhance online security and user protection. The results demonstrated the efficacy of this innovative method in proactively identifying and mitigating crypto-jacking threats, thereby safeguarding users from unauthorized resource exploitation and preserving their online privacy. The adaptability of the approach to emerging crypto-jacking techniques, coupled with its potential for resource conservation and reduced environmental impact, underscores its significant role in the realm of cybersecurity. This research not only sheds light on the importance of JavaScript memory heap analysis as a viable tool for detecting crypto-jacking but also serves as a testament to the dynamic and evolving nature of online threats. As the digital landscape continues to evolve, this study highlights the imperative of staying at the forefront of research and innovation to ensure a safer and more secure online experience for users worldwide.

REFERENCES

[1]. M. H. Khan Abbasi, S. Ullah, T. Ahmad, and A. Buriro, "A Real-Time Hybrid Approach to Combat In-Browser Cryptojacking Malware," Applied Sciences, vol. 13, no. 4, p. 2039, 2023.

[2]. E. Tekiner, "A Deep-dive into Cryptojacking Malware: From an Empirical Analysis to a Detection Method for Computationally Weak Devices," 2021.

[3]. E. Tekiner, A. Acar, A. S. Uluagac, E. Kirda, and A. A. Selcuk, "SoK: cryptojacking malware," in 2021 IEEE European Symposium on Security and Privacy (EuroS&P), 2021: IEEE, pp. 120-139.

[4]. A. Hernandez-Suarez et al., "Detecting cryptojacking web threats: An approach with autoencoders and deep dense neural networks," Applied Sciences, vol. 12, no. 7, p. 3234, 2022.

[5]. J. Burgess and S. Sezer, "Investigation of Browser and Web-based Threats," Queen's University Belfast, 2023.

[6]. P. Wang, Y. Sun, C. Huang, Y. Du, G. Liang, and G. Long, "MineDetector: JavaScript Browser-side Cryptomining Detection using Static Methods," in 2021 IEEE 24th International Conference on Computational Science and Engineering (CSE), 2021: IEEE, pp. 87-93.

[7]. G. Yu et al., "Minergate: A novel generic and accurate defense solution against web-based cryptocurrency mining attacks," in Cyber Security: 17th China Annual Conference, CNCERT 2020, Beijing, China, August 12, 2020, Revised Selected Papers 17, 2020: Springer Singapore, pp. 50-70.

[8]. J. Liu, Z. Zhao, X. Cui, Z. Wang, and Q. Liu, "A novel approach for detecting browser-based silent miner," in 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), 2018: IEEE, pp. 490-497.

[9]. J. Burgess, D. Carlin, P. O'Kane, and S. Sezer, "Manic: Multi-step assessment for crypto-miners," in 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 2019: IEEE, pp. 1-8.

[10]. H. Hong, S. Woo, S. Park, J. Lee, and H. Lee, "Circuit: A JavaScript Memory Heap-Based Approach for Precisely Detecting Cryptojacking Websites," IEEE Access, vol. 10, pp. 95356-95368, 2022.

[11]. I. Petrov, L. Invernizzi, and E. Bursztein, "Coinpolice: Detecting hidden cryptojacking attacks with neural networks," arXiv preprint arXiv:2006.10861, 2020.

[12]. F. N. Naseem, A. Aris, L. Babun, E. Tekiner, and A. S. Uluagac, "MINOS: A Lightweight Real-Time Cryptojacking Detection System," in NDSS, 2021.

[13]. P. Papadopoulos, P. Ilia, and E. Markatos, "Truth in web mining: Measuring the profitability and the imposed overheads of cryptojacking," in Information Security: 22nd International Conference, ISC 2019, New York City, NY, USA, September 16–18, 2019, Proceedings 22, 2019: Springer, pp. 277-296.

[14]. M. A. Razali and S. Mohd Shariff, "Cmblock: In-browser detection and prevention cryptojacking tool using blacklist and behavior-based detection method," in Advances in Visual Informatics: 6th International Visual Informatics Conference, IVIC 2019, Bangi, Malaysia, November 19–21, 2019, Proceedings 6, 2019: Springer, pp. 404-414.

[15]. H. L. Bijmans, T. M. Booij, and C. Doerr, "Inadvertently making cyber criminals rich: A comprehensive study of cryptojacking campaigns at internet scale," in 28th USENIX Security Symposium (USENIX Security 19), 2019, pp. 1627-1644.