Penerbit
Akademia Baru

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

Anti-Cryptojacking Detection Systems (ACDS): Safeguarding Against Covert Cryptocurrency Mining

Nor ‘Afifah Sabri[1,*], Zanariah Zainudin[2], Norliana Muslim[1], Shakiroh Khamis[2], Nurul Syafidah Jamil[2]

[1] Department of Computer and Communication Technology, Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, 31900, Kampar, Perak, Malaysia
[2] Department of Digital Economy Technology, Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, 31900, Kampar, Perak, Malaysia

ARTICLE INFO

Article history:
Received 27 March 2025
Received in revised form 7 April 2025
Accepted 18 October 2025
Available online 12 December 2025

ABSTRACT

Cryptojacking is a form of cybercrime where attackers secretly use victims' devices to mine cryptocurrencies without their knowledge. It involves embedding malicious scripts in computers, smartphones, or servers to mine for digital money like Monero. The motive behind cryptojacking is profit, exploiting victims' resources while remaining hidden. Cryptojacking attacks has greatly impacted business organizations as mining involves solving highly complex mathematical problems which require immense computational power and browser applications are mainly targeted to conduct the mining. In addition, the damage that can cause to business organizations is uncharitable, it may disrupt business services to run accordingly, loss of reputation and most importantly time and cost needed for the invasion while user will have to sacrifices their computing power for hacker’s financial gain. Preventive measures include monitoring device resources, using browser extensions to block mining, updating software, and blocking known cryptojacking sites. To prevent cryptojacking, individuals and organizations can use and maintain appropriate systems. Thus, this paper introduced Anti-Cryptojacking Detection Systems (ACDS) which is developed using Python programming. The ACDS uses a web crawler built with the LXML library in Python. This library processes XML and HTML. The web crawler accesses a targeted webpage and retrieves all its HTML code without rendering the webpage. The ACDS result classified the type of cryptojacking and the level of conditions of the cryptojacking and all results from ACDS can be exported into csv file. In conclusion, ACDS can detect webpages that use any type of cryptojacking and are capable of detecting six different types of crypto mining services running on the webpage. In future work, it suggested having GUI interface for ACDS.

Keywords:

Cryptojacking, Script Injection, Cryptocurrency, Malware Detection

1. Introduction

In our daily lives, we often browse web pages for online shopping, watch TV series on Netflix, stream online radio and many other activities that can lead to serious cybersecurity issues which are known as cryptojacking. Cryptojacking is the term for the covert mining of cryptocurrencies using a

[*] *Corresponding author*
E-mail address: afifahs@utar.edu.my

https://doi.org/10.37934/ard.135.1.264274

computer device. It is a technique used to covertly install a programme that mines cryptocurrencies in the background [1, 2]. Furthermore, by taking advantage of victims' computer systems, cryptojacking is a popular way to obtain extra, free processing power. There are two types of cryptojacking. Firstly, as with other malware attacks, a user is tricked into downloading a mining application onto their computer and starting the mining service. Secondly, another, simpler variant of cryptojacking involves injecting cryptojacking scripts into legitimate webpages and executing them without the consent of website visitors.

Cryptojacking represents an illicit mining operation wherein the computing resources of a victim's computer is exploited to mine cryptocurrency without consent. This illicit activity significantly drains computational resources, resulting in a notable reduction in the victim's computer processing efficiency. Furthermore, sophisticated attackers may leverage this attack vector to amplify their processing power, posing a threat to any blockchain network reliant on mining [3,4,5,6,7]. When computer resources are used as "hash power" in a mining process that involves making intricate and well-informed mathematical guesses to solve equations in order to obtain the block reward for mining cryptocurrency, this is known as cryptojacking—a purposeful attempt to steal the user's computer resources for financial gain in the form of cryptocurrency [8, 9]. Drive-by-download and drive-by-mining are two categories into which cryptojacking falls. Drive-by-mining employs third-party API JavaScript that is injected into the website; when the website is executed, the cryptojacking process starts instantly, whereas drive-by-download uses an exploit kit on a web browser to download malware to the victim's PC. In addition, there are more and more browser-based cryptojacking attempts that use mining scripts and secretly exploit users' system resources and put users in an extremely painful state because the longer the user stays, the more performance is affected. In the worst-case scenario, the user's system hardware life is shortened if the scripts are run at a high throttling rate [10].

Installing or providing consent is not necessary for browser-based cryptojacking mining code, which operates automatically anytime a user opens a webpage. While there is nothing intrinsically harmful or dangerous about JavaScript-based mining, the likelihood of being cryptojacked while online has significantly grown as more websites turn to crypto-mining services in place of traditional advertisement-based monetization. With that being said, prevention should always be considered before browsing information using a web browser [1,10].

While cryptojacking detection techniques such as browser extensions and antivirus software offer some level of mitigation, they only provide a partial solution to the cryptojacking menace. Attackers can easily circumvent these measures through the use of obfuscation techniques or by frequently changing domains and malicious scripts [8]. Cryptojacking boomed with the birth of service providers that offer ready-to-use implementations of mining scripts in web browsers [11,12,13,14]. Example on cryptojacking type are include Coinhive [8,11], CryptoLoot [9] and others, enabling attackers to get a wider pool of victims through compromised websites.

Due to Coinhive's propensity to have its computer code utilised on compromised websites in order to steal the processing power of its visitors' machines, a number of security organisations have lately named the cryptocurrency mining service as the biggest harmful threat to anyone using the internet [3]. Coinhive has grown in popularity among hackers for use in cryptojacking, which uses it to mine the cryptocurrency Monero, despite the fact that Coinhive is not intrinsically harmful. Coinhive, the cryptocurrency mining service that enabled websites to use browser CPUs to mine Monero, shut down on March 8, 2019, due to economic problems. In summary, Coinhive shut down in 2019 due to economic problems, and its closure was expected to end cryptojacking attacks [1,13]. However, the Coinhive JavaScript code is still in place on many websites, and if the value of Monero increases significantly, it is possible that attackers could use it again for cryptojacking.

A crypto-mining service called CryptoLoot looks to be exactly like Coin-Hive, but it claims that using its miner will be "unnoticed by users if the threads are set between 2-4." [8,14]. CryptoLoot considers itself one of the early adopters by providing one of the first NFT marketplaces for exclusively trading GameFi NFT. However, it is important to note that the search results do not provide comprehensive information on CryptoLoot's NFT marketplace.

The rapid growth of browser-based cryptojacking scripts poses a significant threat to users' online experience and the health of their hardware. These scripts secretly mine cryptocurrency on victims' devices and utilize browser resources without users' consent or knowledge [15]. This insidious activity leads to a gradual but demonstrably measurable degradation of the user experience, characterized by performance degradation [16]. The severity of this degradation is directly related to the duration of exposure to the malicious script. Furthermore, the continuous, unauthorized use of system resources for cryptocurrency mining can shorten the lifespan of user hardware and, in extreme cases, even cause permanent damage [17]. These detrimental effects are often exacerbated by the automatic execution of these scripts when a page is loaded, bypassing the user's installation and opt-in mechanisms. The additional danger of scripts being surreptitiously downloaded and executed via phishing websites emphasizes the prevalence and stealthy nature of this threat [17]. Although JavaScript-based mining is not inherently associated with malicious intent, its increasing prevalence on websites, especially as an alternative to ad revenue, significantly increases the likelihood of users unwittingly participating in cryptojacking activities. To address this expanding issue and safeguard consumers' devices and online experiences, it is crucial to create efficient detection, prevention, and mitigation solutions.

Thus, this paper proposes the development of an Anti-Cryptojacking Detection System (ACDS) to protect users from unethical website owners who inject cryptojacking scripts without the user's consent. An independent command line interface program makes up the ACDS. By entering the target's Uniform Resource Locator (URL), it allows users to undertake cryptojacking script scanning to shield them from cryptojacking and avert the unintended consequences of visiting fraudulent websites. The ACDS specifically addresses the vulnerability of all web users to cryptojacking attacks, regardless of their browsing frequency or purpose. The existence of a cryptojacking threat arises from the utilization of malicious JavaScript embedded within web pages. It operates covertly, executing without users' consent, and only impacting through web browsers that can run JavaScript, encompassing both desktop and mobile versions. Therefore, through the development of ACDS, forensics professionals and security experts can utilize the collected data for future study and analysis. By proactively detecting malicious scripts, the ACDS aims to reduce the rising threat of cryptojacking.

2. Methodology

This section explains the methodological development of ACDS and the required functions for scanning webpage URLs. This includes the Agile Model, system architecture and implementation process for developing ACDS.

2.1 Agile Model

The application development methodology selected for this research is the Agile Model as illustrated in Figure 1. The Agile Model represents an adaptive software development approach renowned for its significant flexibility [18,19,20]. The Agile Model for ACDS focuses on an iterative development cycle in which cross-functional teams work together to deliver small, functional software increments within a short period. Continuous integration and testing procedures ensure that the software increments are thoroughly tested for quality and functionality, reducing the risk of errors and improving overall reliability. It is also an ideal method for circumstances where changes are required, including fixing bugs and correcting mismatched requirements after testing. This model allows the project to be delivered with first-class quality as the developer can continuously meet the requirements of the target user.

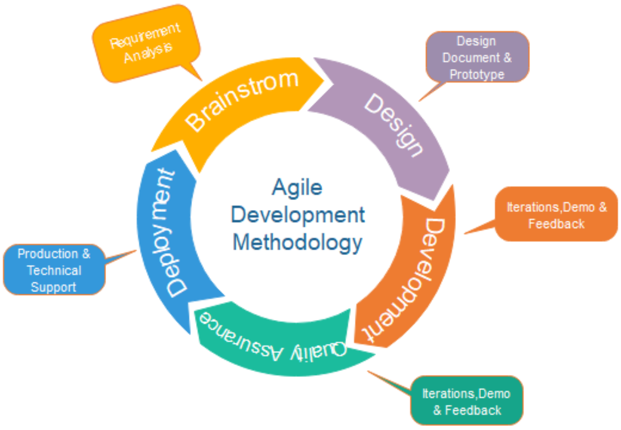

Requirement Analysis
Brainstrom
Design Document & Prototype
Design
Agile Development Methodology
Iterations,Demo & Feedback
Development
Production & Technical Support
Deployment
Quality Assurance
Iterations,Demo & Feedback

Fig.1. Agile Model for ACDS [18]

2.2 *System Architecture for ACDS*

System architecture diagrams for ACDS serve as visual representations of a system's components and their interactions, providing a clear understanding of its functionality and potential enhancement. The system architecture based on a use case diagram is illustrated in Figure 2. The use case diagrams serve as a valuable tool for identifying and prioritizing system requirements, guiding the iterative development process of ACDS by focusing efforts on delivering features. In addition to the use case diagram, the system architecture of ACDS may also be depicted using other graphical representations, such as a class diagram or sequence diagram, offering different perspectives on its structure and behavior. These architectural diagrams not only aid in comprehending the system’s current state but also facilitate communication among stakeholders, fostering collaboration and alignment objectives throughout the development process [21, 22].

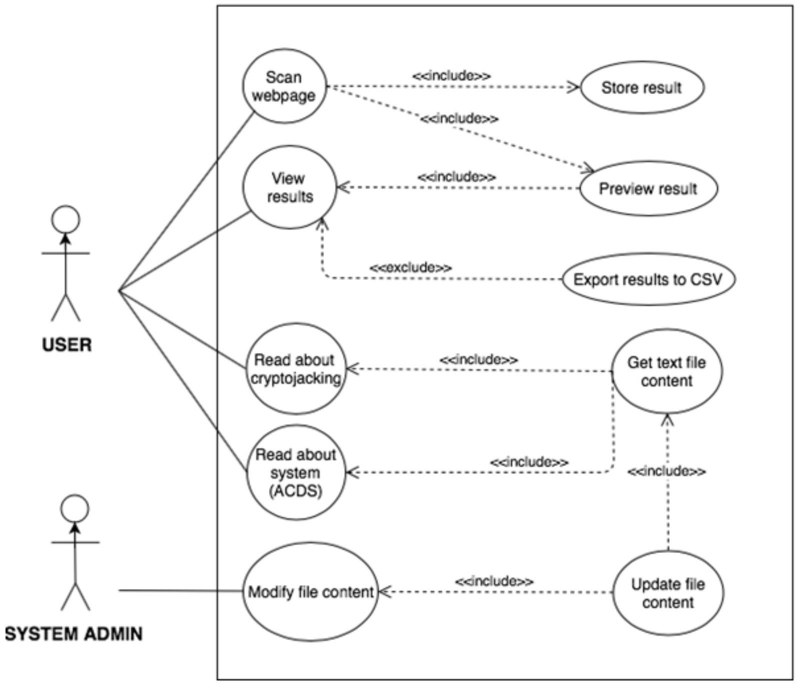


Fig. 2. Use Case Diagram for ACDS

Referring to Figure 2, two primary actors namely the user and the system admin were identified. The user possesses capabilities of reading about cryptojacking and the system, retrieving text file content, modifying file content, and previewing results. On the other hand, the system admin can execute actions such as scanning webpages, viewing results, storing results, exporting results to Comma-Separated Values (CSV), and updating file content. Table 1 shows the use-case specification for the ACDS where all the actions in the use-case diagram have been elaborated.

Table 1
Use-case specification for ACDS

Function	Description
Scan webpage	It begins with the launch of an effort to scan a URL and takes care of everything from scanning the target webpage to storing and displaying the results of the scan. When the user stops reading the post-scanning results, this use case comes to an end.
View results	From the system's main menu, it begins when the user chooses to view the result. When the user returns to the system's main menu, this use case comes to an end.
Read about cryptojacking	From the system's main menu, users can read more details and become knowledgeable about the hazards of cryptojacking.
Read about system	Users can read details about the capabilities and goals of the developing system.
Modify file content	It starts when the system administrator chooses to change any text file's content (regarding cryptojacking or the system).

Referring to the Table 1, the ACDS functions in the system is Scan webpage, View results, Read about cryptojacking, Read about system, and Modify file content. The function Scan webpage will scan website using its' URL, the function View results will publish all the scanning results, the function Read about cryptojacking will explain more details about the cryptojacking, the function Read about system allow users to read details about the system, and lastly the function Modify file content will allow system administrator to change any text file's content.

2.3 Implementation of ACDS

The ACDS development was implemented with PyCharm on MacOS, using Python 3.7 as the primary programming language. Python was considered the most suitable choice due to its efficiency in web crawler development. Throughout the development process, key libraries such as XML, Requests, pymysql, Logging and Socket played an essential role in achieving the project goal.

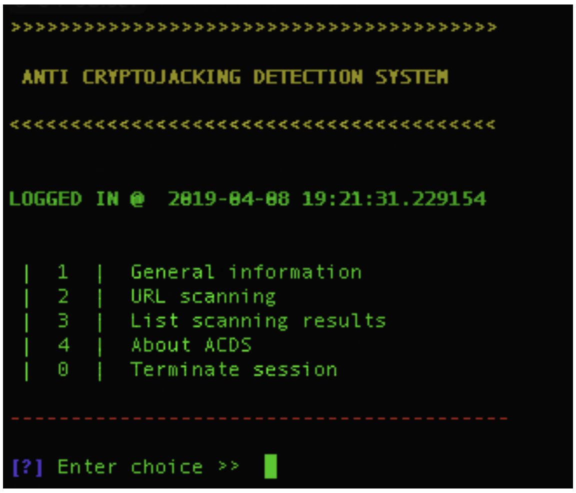
MySQL v8.0.15 was chosen as the database management system, i.e., data management is carried out via the MySQL Workbench. Compared to SQLite, MySQL stores the data in a centralized server, which is more useful when developing a system that can manage a large database with all meaningful data from different types of users [23]. The cryptojacking script needs to be performed while the website is active by monitoring the CPU usage of the system. Web browsers such Netscape Navigator 2.0, Microsoft Internet Explorer 3.0, Firefox, Safari, Opera, and Google Chrome (some with minimum version restrictions, others without) were able to support JavaScript.The web crawling library, namely XML, was used to access the targeted web page, then request all the HTML code and crawl back to ACDS without rendering the webpage. Script validation was performed on the system side to determine if the webpage had cryptojacking scripts.

3. Results

This section highlights the important result that was specially developed for detecting cryptojacking in web pages. This was followed by URL scanning and mining results.

3.1 Anti-Cryptojacking Detection System (ACDS)

ACDS is a system with a command line interface that serves as a tool for detecting cryptojacking scripts based on the URL entered. The user has the option of viewing the scan results previously saved in the system database and exporting the results to a CSV file. To gain more knowledge about cryptojacking threats, basic information about cryptojacking is included in the system for the user to read. Every action performed in the system is recorded in a log file, which helps the developer to detect and investigate all activities and fix problems. Figure 3 shows the landing page of the ACDS. From this main menu, the user can navigate to all options by entering their number, which is marked with an icon. In addition to the system name and the selection of options, the system also shows the user a time stamp that is recorded when the system is running. All user selections have been recorded in a log file for security reasons.

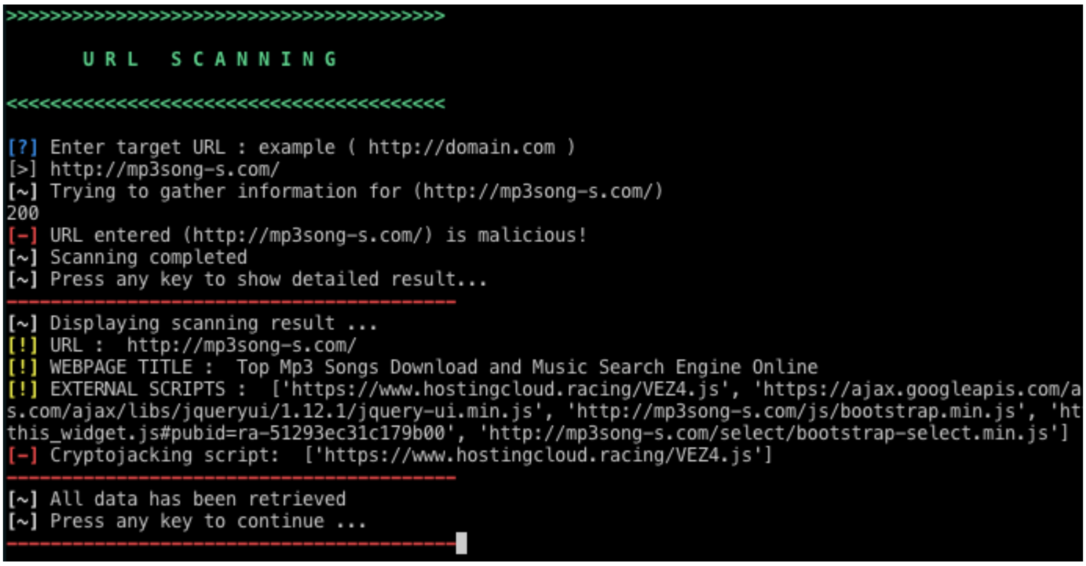


```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

 ANTI CRYPTOJACKING DETECTION SYSTEM

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<


LOGGED IN @  2019-04-08 19:21:31.229154


 |  1  |  General information
 |  2  |  URL scanning
 |  3  |  List scanning results
 |  4  |  About ACDS
 |  0  |  Terminate session

-----------------------------------------

[?] Enter choice >>
```

Fig. 3. Homepage

3.2 URL Scanning

The primary function of this application is URL scanning, which enables the user to start a cryptojacking script check by entering any URL in a valid format, as seen in Figure 4. If the URL is invalid or cannot be processed, the ACDS application handles this with exceptions such as invalid URL, connection error, timeout error, socket error and more, as shown in Figure 5.



```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

     U R L   S C A N N I N G

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

[?] Enter target URL : example ( http://domain.com )
[>] http://mp3song-s.com/
[~] Trying to gather information for (http://mp3song-s.com/)
200
[-] URL entered (http://mp3song-s.com/) is malicious!
[~] Scanning completed
[~] Press any key to show detailed result...
------------------------------------------
[~] Displaying scanning result ...
[!] URL :  http://mp3song-s.com/
[!] WEBPAGE TITLE :  Top Mp3 Songs Download and Music Search Engine Online
[!] EXTERNAL SCRIPTS :  ['https://www.hostingcloud.racing/VEZ4.js', 'https://ajax.googleapis.com/a
s.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js', 'http://mp3song-s.com/js/bootstrap.min.js', 'ht
this_widget.js#pubid=ra-51293ec31c179b00', 'http://mp3song-s.com/select/bootstrap-select.min.js']
[-] Cryptojacking script:  ['https://www.hostingcloud.racing/VEZ4.js']
------------------------------------------
[~] All data has been retrieved
[~] Press any key to continue ...
------------------------------------------
```

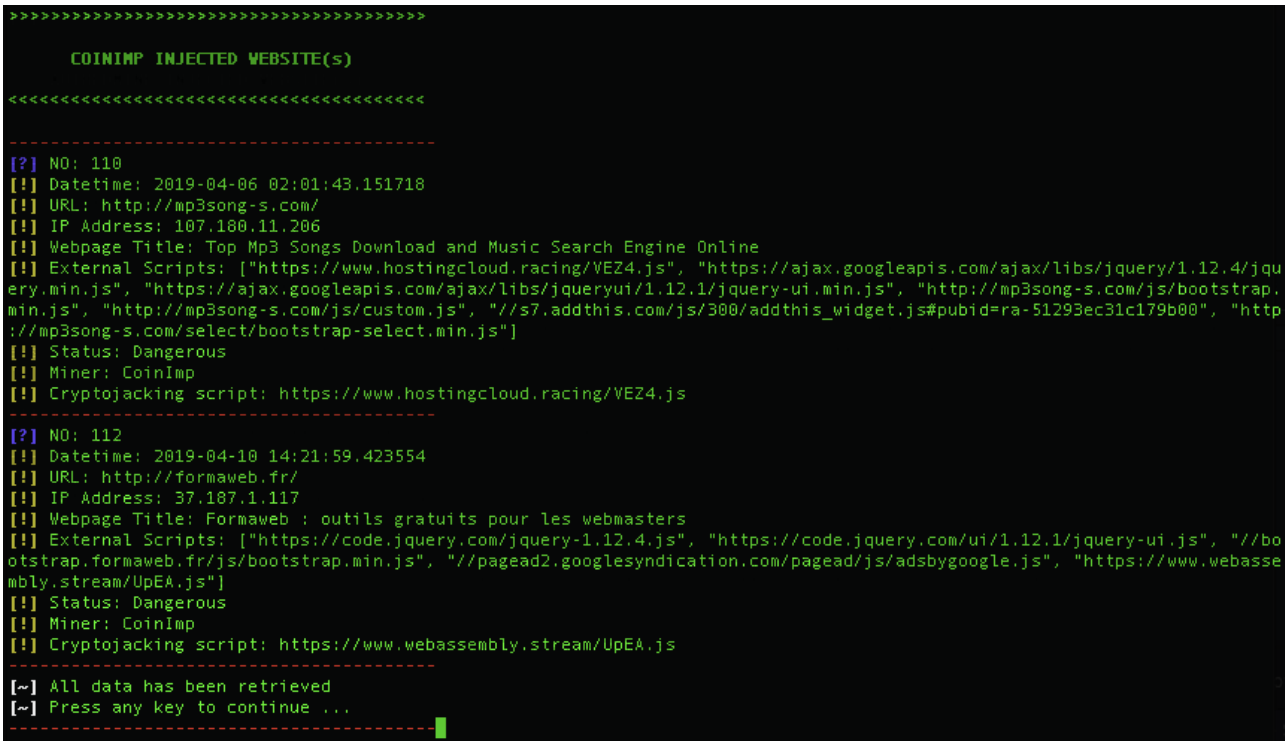
Fig. 4. URL Scanning for Malicious Website

The system used a web crawler to access the webpage and extract all static HTML code after a valid URL was entered into the ACDS to perform scanning. This application was checked against defined keywords to see if the webpage contained a cryptojacking script. The system then displays the scan results to the user and indicates whether the webpage is malicious by inserting a cryptojacking script along with detailed scan results such as webpage title, IP address and all Java Scripts used by the webpage.

3.2 Mining Results

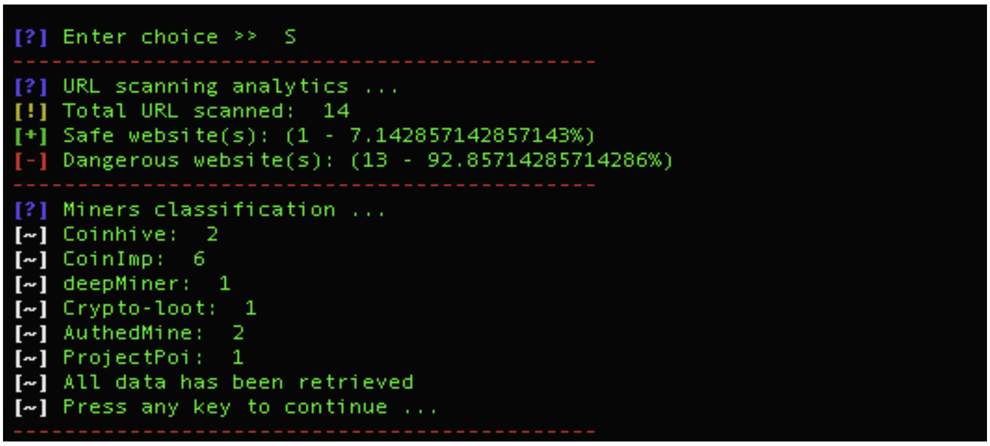
The purpose of this module is to allow users to view all previously scanned web pages. Findings indicated that through ACDS, a variety of options for the users to choose from were established. For instance, users can view all results, by different types of crypto miners and by conditions (safe or dangerous). In addition, ACDS allows users to view summarized results based on all previous scanned results, in which statistical data based on a predefined calculation is displayed.

Figure 5 displays the results that the user retrieves using CoinImp, a cryptocurrency miner. The user can see the unique identifier key in the URL, the title of the webpage, the IP address, any external scripts that the webpage uses, and the malicious JavaScript that ACDS discovered attempting to mine the user's system resources (unsuitable for the selection of a secure webpage).



```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      COINIMP INJECTED WEBSITE(s)

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

-----------------------------------------
[?] NO: 110
[!] Datetime: 2019-04-06 02:01:43.151718
[!] URL: http://mp3song-s.com/
[!] IP Address: 107.180.11.206
[!] Webpage Title: Top Mp3 Songs Download and Music Search Engine Online
[!] External Scripts: ["https://www.hostingcloud.racing/VEZ4.js", "https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jqu
ery.min.js", "https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js", "http://mp3song-s.com/js/bootstrap.
min.js", "http://mp3song-s.com/js/custom.js", "//s7.addthis.com/js/300/addthis_widget.js#pubid=ra-51293ec31c179b00", "http
://mp3song-s.com/select/bootstrap-select.min.js"]
[!] Status: Dangerous
[!] Miner: CoinImp
[!] Cryptojacking script: https://www.hostingcloud.racing/VEZ4.js
-----------------------------------------
[?] NO: 112
[!] Datetime: 2019-04-10 14:21:59.423554
[!] URL: http://formaweb.fr/
[!] IP Address: 37.187.1.117
[!] Webpage Title: Formaweb : outils gratuits pour les webmasters
[!] External Scripts: ["https://code.jquery.com/jquery-1.12.4.js", "https://code.jquery.com/ui/1.12.1/jquery-ui.js", "//bo
otstrap.formaweb.fr/js/bootstrap.min.js", "//pagead2.googlesyndication.com/pagead/js/adsbygoogle.js", "https://www.webasse
mbly.stream/UpEA.js"]
[!] Status: Dangerous
[!] Miner: CoinImp
[!] Cryptojacking script: https://www.webassembly.stream/UpEA.js
-----------------------------------------
[~] All data has been retrieved
[~] Press any key to continue ...
-----------------------------------------
```

Fig. 5. List Results by Miner Type



```
[?] Enter choice >>  S
-----------------------------------------
[?] URL scanning analytics ...
[!] Total URL scanned:  14
[+] Safe website(s): (1 - 7.142857142857143%)
[-] Dangerous website(s): (13 - 92.85714285714286%)
-----------------------------------------
[?] Miners classification ...
[~] Coinhive:  2
[~] CoinImp:  6
[~] deepMiner:  1
[~] Crypto-loot:  1
[~] AuthedMine:  2
[~] ProjectPoi:  1
[~] All data has been retrieved
[~] Press any key to continue ...
-----------------------------------------
```

Fig. 6. Summarized Results

The results page, as seen in Figure 6, allows users to view all websites that have previously been scanned with ACDS. The menu has multiple options, including all, by status (malicious or safe), by type of crypto miner, and summarised results. The user selects an option from the menu, and detailed results are retrieved from the database and presented to the user. Results indicated that the ACDS successfully scanned 14 websites and detected 7.14% as safe websites, while 92.86% as dangerous websites. From 13 dangerous websites, further classification was attained such that 2 Coinhive, 6 CoinImp, 1 deepMiner, 1 CryyptoLoot, 2 Authermine and 1 Projectpoi. All of the results now can be exported into CSV file as shown in Table 2.

Table 2
Sample of ACDS Results

Id	URL	Web Page Title	Conditions	Type
3	http://legendaofficial .net/	Legenda Official	Dangerous	CryptoLoot
4	http://themelike .net/	All Design Template-Photoshop Vector	Dangerous	Coinhive
51	http://snapp-free.blog .ir/	-	Dangerous	AuthedMiner
106	http://iracom .ir/	-	Dangerous	deepMiner
107	http://fluxuslust.blogspot.com/	Fluxus Lust-Discreet Pleasures	Safe	Not detected
110	http://mp3song-s.com/	Top mp3 Songs Download	Dangerous	Coinimp

Table 2 shows six contents of the exported CSV file (out of 14 scanned websites). This table 2 presents the relevant information from the exported CSV file, showcasing the IDs, URLs, web page titles, cryptojacking conditions, and types detected by the ACDS. The CSV file displays the cryptojacking conditions available on the webpage that ACDS has scanned and the type of cryptojacking, such as CryptoLoot, Coinhive, etc. Six different IDs with corresponding URLs, web page titles and conditions have been successfully obtained. It emphasizes the successful detection of cryptojacking conditions across all scanned websites, indicating the effectiveness of the developed ACDS.

4. Conclusions

The prevailing increasing number of cryptojacking threats and the dearth of effective solutions that can both raise awareness and protect users are the only driving forces behind the concept of this work. The developed ACDS offers numerous potential benefits, including the protection of resources, data and privacy. By detecting and preventing unauthorized cryptocurrency mining activities, the ACDS can protect computer resources such as the Central Processing Unit or the Graphical Processing Unit. Cryptojacking malware can be distributed along with other malicious payloads that can lead to data breaches or contain sensitive information. By detecting and removing cryptojacking malware, the ACDS helps to protect data integrity and user privacy.

Since sample users have personally tested and assessed the system, it has been determined that the built ACDS is crash-free, responsive, and dependable, which will ultimately benefit the intended end user. First and foremost, the developer may draw the conclusion that testers are happy with the system's features if users deem it to be user-friendly and well-functioning based on the results of the user acceptance test. Furthermore, the system's learnability was improved by the developer to guarantee that every user has a broader understanding of the threat posed by cryptojacking.

In addition, the system operates incredibly quickly and retrieves data with accuracy and dependability. This is a result of the system's use of a commonly-used framework for data retrieval. Therefore, the user's experience with the system is more crucial. The ACDS has succeeded in this regard as the user did not encounter an application crash during interaction. In addition, some bugs were found as a result of the framework's inherent restrictions and internet regulations, such as the inability to access websites that are geographically restricted from the web crawler or websites that block HTTP requests from the web crawler. To further explain, webpage scanning for cryptojacking script detection requires an active internet connection from the user's machine. Since the internet is provided by the user's regional ISP, any blocked websites will not be accessible for data crawling. This issue can be resolved if users configure and use a virtual private network (VPN) before utilizing the system. A proposed solution is to include a VPN configuration option within the system for instances where websites are regionally blocked.

For future work, the developer would think about changing the system's command line interface to a graphical user interface in order to better improvise the current setup. Additionally, the developer plans to integrate an open source virtual private network (VPN) in the future, which will allow users to configure their connection in situations where the web crawler is unable to retrieve data from a website because the user's internet service provider has blocked it regionally. This will address the limitation faced by the web crawler due to blocked websites. Furthermore, extending the detection of cryptojacking scripts to various crypto miners would significantly increase the effectiveness and relevance of the system in combating emerging threats.

Acknowledgement

This research was conducted independently without external funding. The study benefited from the research facility support from Universiti Tunku Abdul Rahman.

References

[1] Ege Tekiner, Abbas Acar, A. Selcuk Uluagac, Engin Kirda, and Ali Aydın Selçuk. 2021. “SoK: Cryptojacking Malware.” ArXiv (Cornell University), March. https://doi.org/10.48550/arxiv.2103.03851.

[2] Mansor, Wan Nur Aaisyah Binti Wan, Azuan Ahmad, Wan Shafiuddin Zainudin, Madihah Mohd Saudi, and Mohd Nazri Kama. 2020. “Crytojacking Classification Based on Machine Learning Algorithm.” Proceedings of the 2020 8th International Conference on Communications and Broadband Networking, April. https://doi.org/10.1145/3390525.3390537.

[3] Xu, Guangquan, Wenyu Dong, Jun Xing, Wenqing Lei, Jian Liu, Lixiao Gong, Meiqi Feng, Xi Zheng, and Shaoying Liu. "Delay-CJ: A novel cryptojacking covert attack method based on delayed strategy and its detection." *Digital Communications and Networks* 9, no. 5 (2023): 1169-1179. https://doi.org/10.1016/j.dcan.2022.04.030.

[4] Caprolu, Maurantonio, Simone Raponi, Gabriele Oligeri, and Roberto Di Pietro. "Cryptomining makes noise: Detecting cryptojacking via machine learning." *Computer Communications* 171 (2021): 126-139. https://doi.org/10.1016/j.comcom.2021.02.016.

[5] Meland, Per Håkon, Bent Heier Johansen, and Guttorm Sindre. "An experimental analysis of cryptojacking attacks." In *Nordic Conference on Secure IT Systems*, pp. 155-170. Cham: Springer International Publishing, 2019. https://doi.org/10.1145/10.1007/978-3-030-35055-0_10.

[6] Kharraz, Amin, Zane Ma, Paul Murley, Charles Lever, Joshua Mason, Andrew Miller, Nikita Borisov, Manos Antonakakis, and Michael Bailey. "Outguard: Detecting in-browser covert cryptocurrency mining in the wild." In *The World Wide Web Conference*, pp. 840-852. 2019. https://doi.org/10.1145/3570361.3613283.

[7] Eskandari, Shayan, Andreas Leoutsarakos, Troy Mursch, and Jeremy Clark. "A first look at browser-based cryptojacking." In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 58-66. IEEE, 2018. https://doi.org/10.48550/arXiv.1803.02887.

[8] Hernandez-Suarez, Aldo, Gabriel Sanchez-Perez, Linda K. Toscano-Medina, Jesus Olivares-Mercado, Jose Portillo-Portilo, Juan-Gerardo Avalos, and Luis Javier García Villalba. 2022. "Detecting Cryptojacking Web Threats: An Approach with Autoencoders and Deep Dense Neural Networks" *Applied Sciences* 12, no. 7: 3234. https://doi.org/10.3390/app12073234.

[9] Ying, Qianjin, Yulei Yu, Donghai Tian, Xiaoqi Jia, Rui Ma, and Changzhen Hu. "Cjspector: A novel cryptojacking detection method using hardware trace and deep learning." *Journal of Grid Computing* 20, no. 3 (2022): 31. https://doi.org/10.1007/s10723-022-09621-2.

[10] Aponte-Novoa, Fredy Andrés, Daniel Povedano Álvarez, Ricardo Villanueva-Polanco, Ana Lucila Sandoval Orozco, and Luis Javier García Villalba. 2022. "On Detecting Cryptojacking on Websites: Revisiting the Use of Classifiers" *Sensors* 22, no. 23: 9219. https://doi.org/10.3390/s22239219.

[11] Varlioglu, Said, Bilal Gonen, Murat Ozer, and Mehmet Bastug. "Is cryptojacking dead after coinhive shutdown?." In *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, pp. 385-389. IEEE, 2020. https://doi.org/10.48550/arXiv.2001.02975.

[12] Khan Abbasi, Muhammad Haris, Subhan Ullah, Tahir Ahmad, and Attaullah Buriro. "A Real-Time Hybrid Approach to Combat In-Browser Cryptojacking Malware." *Applied Sciences* 13, no. 4 (2023): 2039. https://doi.org/10.3390/app13042039.

[13] Sakas, Damianos P., Nikolaos T. Giannakopoulos, Dimitrios P. Reklitis, and Thomas K. Dasaklis. "The effects of cryptocurrency trading websites on airlines' advertisement campaigns." *Journal of Theoretical and Applied Electronic Commerce Research* 16, no. 7 (2021): 3099-3119. https://doi.org/10.3390/jtaer16070169.

[14] Sivaraju, S. S. 2022. "An Insight into Deep Learning Based Cryptojacking Detection Model." *Journal of Trends in Computer Science and Smart Technology* 4 (3): 175–84. https://doi.org/10.36548/jtcsst.2022.3.006.

[15] Zhang, Shize, Zhiliang Wang, Jiahai Yang, Cheng Xin, Xiaoqian Ma, Hui Zhang, Bo Wang, Zimu Li, and Jianping Wu. 2021. "MineHunter: A Practical Cryptomining Traffic Detection Algorithm Based on Time Series Tracking." *Annual Computer Security Applications Conference*, December. https://doi.org/10.1145/3485832.3485835.

[16] Caprolu, Maurantonio, Simone Raponi, Gabriele Oligeri, and Roberto Di Pietro. 2021. "Cryptomining Makes Noise: Detecting Cryptojacking via Machine Learning." *Computer Communications*, February. https://doi.org/10.1016/j.comcom.2021.02.016.

[17] Naseem, F., Ahmet Aris, Leonardo Babun, Ege Tekiner, and A. Uluagac. 2021. "MINOS: A Lightweight Real-Time Cryptojacking Detection System." Semantic Scholar. 2021. https://doi.org/10.14722/NDSS.2021.24444.

[18] Emmanuel Chibuike Daraojimba, Chinedu Nnamdi Nwasike, Abimbola Oluwatoyin Adegbite, Chinedu Alex Ezeigweneme, and Joachim Osheyor Gidiagba. 2024. "Comprehensive Review Of Agile Methodologies In Project Management" 5 (1): 190–218. https://doi.org/10.51594/csitrj.v5i1.717.

[19] Alsaqqa, Samar, Samer Sawalha, and Heba Abdel-Nabi. 2020. "Agile Software Development: Methodologies and Trends." *International Journal of Interactive Mobile Technologies* 14 (11): 246–70. https://doi.org/10.3991/ijim.v14i11.13269.

[20] Dingsøyr, Torgeir, Sridhar Nerur, VenuGopal Balijepally, and Nils Brede Moe. 2012. "A Decade of Agile Methodologies: Towards Explaining Agile Software Development." *Journal of Systems and Software* 85 (6): 1213–21. https://doi.org/10.1016/j.jss.2012.02.033.

[21] Aziah Asmawi, Ezzah Mawadah Saifulbahri, and Noor Afiza Mohd Ariffin. 2023. "Development of BlockScholar As an Educational Mobile Application on Blockchain Technology". Journal of Advanced Research in Applied Sciences and Engineering Technology 34 (1): 15-23. https://doi.org/10.37934/araset.34.1.1523.

[22] Mudiana Mokhsin, Mohd Husni Mohd Som, Muhammad Firdaus Suhaimi, Hamizan Abdul Halim, and Ahmad Zambri Shahuddin. 2023. "The Development of a Rental Equipment Mobile Application for UiTM Shah Alam Malay and Bumiputera Students (Rent2U)". Journal of Advanced Research in Applied Sciences and Engineering Technology 32 (3):282-95. https://doi.org/10.37934/araset.32.3.282295.

[23] Emily Sing Kiang Siew, Zhi Yan Chong, Sah Nah Sze, and Richki Hardi. 2023. "Streamlining Attendance Management in Education: A Web-Based System Combining Facial Recognition and QR Code Technology". Journal of Advanced Research in Applied Sciences and Engineering Technology 33 (2):198-208. https://doi.org/10.37934/araset.33.2.198208.