Detecting Cryptomining Traffic in IoT Networks

Publisher:
Institute of Electrical and Electronics Engineers (IEEE)

Published
DOI:

Detecting Cryptomining Traffic in IoT Networks

Luca Mannella*‡
Dip. di Automatica e Informatica
Politecnico di Torino
Turin, Italy
luca.mannella@polito.it

Daniele Canavese*
IRIT
CNRS
Toulouse, France
daniele.canavese@irit.fr

Leonardo Regano*
Dip. di Ing. Elettrica ed Elettronica
Università degli Studi di Cagliari
Cagliari, Italy
leonardo.regano@unica.it

***Abstract*—With the proliferation of Internet of Things (IoT) devices in smart home environments, the threat landscape has expanded to include cryptojacking attacks, which exploit computational resources for unauthorized cryptocurrency mining. In response to these challenges, this paper extends the security capability of the IoT Proxy—–a recently proposed solution to enhance the security of resource-constrained IoT devices—–allowing users to take advantage of its security functionalities also in smart home environments. By integrating a new machine learning-based cryptojacking detection module, this enhanced version of the IoT Proxy can identify these emerging threats, safeguarding both the integrity and performance of all the devices in the smart home network. The IoT Proxy serves a dual purpose by providing security functionalities for resource-constrained IoT devices and offering additional layers of defense for less constrained devices, thereby ensuring comprehensive protection across the smart home ecosystem. The conducted experiments demonstrate the efficacy of the approach without relying on Deep Packet Inspection (DPI), thus preserving user privacy and making it applicable even to encrypted traffic—–a prevailing characteristic of contemporary network communication.**

***Index Terms*—Cryptomining, Gateways, Internet of Things (IoT), Intrusion Prevention Systems (IPS), Machine Learning (ML), Network Security**

I. Introduction

The Internet of Things (IoT) still grows year by year. According to IoT Analytics [5], the already considerable number of connected endpoints registered in 2022 (14.4 billion) will increase to around 30 billion by 2027. With the proliferation of IoT devices, ranging from smart home appliances to smart health solutions and industrial sensors, an urgent need arises for robust security solutions capable of safeguarding these devices. Considering that there are scenarios in which IoT devices are limited in terms of battery, computational power, or memory, it became crucial to find a workaround to protect those resource-constrained machines [35]. Indeed, traditional security mechanisms often prove inadequate in addressing the diverse and evolving threats targeting IoT ecosystems. In addition, when novice or hobbyist developers are in the loop, they could overlook security issues and unintentionally develop insecure IoT solutions [10], [12]. Moreover, the interconnection of IoT devices with conventional computing systems in mixed networks, such as smart homes, introduces additional complexities and vulnerabilities that must be addressed to ensure comprehensive security.

As the digital realm has become an integral part of modern life, the motivation behind cyberattacks has increasingly intertwined with economic incentives. The allure of financial gains has emerged as a principal driving force behind the evolution and sophistication of malicious activities in the digital sphere [2]. Amidst this backdrop, the surge in the popularity and value of cryptocurrencies has added a new layer of complexity to the economic motivations of cybercriminals. Cryptocurrencies, epitomized by the groundbreaking advent of Bitcoin [23], captivated not only investors' imagination but also those seeking to exploit the decentralized nature of these digital assets. Indeed, they provide a fertile ground for malicious actors seeking untraceable financial transactions, giving rise to a spectrum of cyber threats [3]. For instance, ransomware, a prevalent cyber threat, exemplifies the intersection of digital currencies and economic exploitation. Ransomware writers typically demand payment from their victims in cryptocurrency, leveraging digital currencies' decentralized and pseudonymous nature to facilitate untraceable transactions [4]. This tactic has become a hallmark of ransomware attacks, where victims are coerced into paying ransom to regain access to encrypted data. As the economic allure of cryptocurrencies grew, the cyber threat landscape witnessed a proliferation of attacks driven by the promise of financial gain. Cryptojacking represents a covert assault on computational resources, as threat actors surreptitiously embed mining scripts within unsuspecting users' devices, harnessing their processing power for unauthorized cryptocurrency mining. The impact is multifaceted, affecting individuals and organizations alike, with enterprises particularly vulnerable to severe consequences such as escalated electricity costs, diminished system performance, and potential disruptions to critical business operations [1].

In response to these challenges, this paper extends the security capability of the *IoT Proxy* [6]—a recently proposed solution to enhance the security of resource-constrained IoT devices—allowing users to take advantage of its security functionalities also in smart home environments. On the other hand, we build on our previous experience in developing a Machine Learning-based system for detecting cryptomining connections in classical computer networks [25].

This work was partially supported by Fondazione CRT (Cassa di Risparmio di Torino) and by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan, which is funded by the European Union - NextGenerationEU.

(*) The authors contributed equally to this work.

(‡) Correspondence: luca.mannella@polito.it

The purpose of the IoT proxy is twofold. On the one hand, it can provide security functionalities when the IoT devices are not powerful enough to protect themselves. Indeed, resource-constrained IoT devices can mitigate security risks by offloading security functions to a dedicated proxy without compromising their performance or functionality. On the other hand, even when the devices are not so constrained, it can be helpful to comply with the security-in-depth paradigm [28]. Indeed, it can provide additional layers of security.

By integrating a new cryptojacking detection module, this enhanced version of the IoT proxy can identify these emerging threats, safeguarding the integrity and performance of all the devices in the smart home network. Indeed, in addition to IoT devices, smart homes typically encompass a variety of computing devices, including PCs, smartphones, and tablets, further diversifying the attack surface. By extending security measures to encompass the entire network environment, the IoT proxy ensures comprehensive protection against a wide range of cyber threats, enhancing the overall security posture of mixed networks. To rigorously evaluate the effectiveness of our approach, we deploy a range of machine learning models, including decision trees, random forests, Gaussian Support Vector Machines (SVMs), and k-Nearest Neighbors (kNN). Crucially, our methodology sets itself apart by not relying on Deep Packet Inspection (DPI), preserving user privacy, and making it applicable even to encrypted traffic — a prevailing characteristic of contemporary network communication [13].

The rest of this paper is structured as follows. Section II defines what is a cryptocurrency, the crypto-mining process, and crypto-jacking attacks. Section III describes how we built our crypto-mining traffic data set, while Section IV presents the results obtained in our experimental setup. Section V compares our approach with other scholars' proposals, and Section VI concludes the paper providing also insights for future research.

II. Background

Cryptocurrencies are a new type of digital exchange that uses cryptographic primitives to regulate currency issuance and transaction verification in a decentralized manner. They were first introduced with Bitcoin (BTC), described in a seminal paper by Satoshi Nakamoto [23]. Cryptocurrencies operate without a central authority, like peer-to-peer (P2P) file-sharing networks. Transactions are validated through a distributed ledger [29], facilitated by Blockchain technology [33]. This ledger records transactions, and a set of transactions (also known as block) is validated through a computationally intensive process called mining. Mining guarantees the validity of each block, protecting the integrity of the system and ensuring the uniqueness of each transaction (i.e., it is not possible to double-spend a cryptocurrency). Mining is based on solving cryptographic puzzles. Miners are motivated to participate in this process by receiving rewards in the form of newly minted cryptocurrency units. However, mining has become increasingly complex over time, requiring specialized hardware, such as Application-Specific Integrated Circuits (ASICs), due to the computational demands.

To overcome the hardware requirements barrier, mining pools have emerged. These pools allow multiple miners to collaborate and use their computational resources to validate transactions and share rewards. A famous protocol used to distribute mining tasks within these pools is Stratum [32], introduced by the Slushpool mining pool [31]. Stratum establishes a bidirectional communication channel between mining pool servers and participants, facilitating efficient task allocation and submission. To enhance security measures, Stratum messages may be encrypted. A common way to achieve this goal is embedding them in Transport Layer Security (TLS) frames [27]. The protocol operates over the JavaScript Object Notation (JSON) Remote Procedure Call (RPC) 2.0 specification [16] and involves the following key steps: session initialization, challenge distribution, and solution submission.

In the first step, clients initiate or resume sessions with the mining pool server and specify the mining software they will use to solve assigned challenges. Authorization for each client device (also called worker) involved in the mining process is also established. The second step occurs when the server sends challenges to the connected workers. Workers attempt to solve these challenges using their computational resources. If they find a solution, the process proceeds to the third step, the submission of a solution. In this last step, workers submit their solutions to the server using the submit RPC method. The server then verifies the correctness of the solutions and distributes new challenges accordingly.

XMR-Stak [15] is a software designed for mining various cryptocurrencies, including Monero (XMR), that utilizes the CryptoNight algorithm. It facilitates mining operations by employing the Stratum protocol for communication with mining pools, either in plaintext or encrypted with TLS. Notably, XMR-Stak features an HTTP interface that allows remote monitoring of the hash rate generated from Monero mining and provides insights into mining results and connection statistics. Moreover, it offers optimization options tailored to the hardware used, such as CPU-only mode for devices lacking GPU support and NVIDIA/AMD mode for systems equipped with compatible GPUs. Due to its widespread use, XMR-Stak was selected for generating a portion of the Stratum traffic—both encrypted and plaintext—for training classifiers in our implementation, as many malware variants incorporate XMR-Stak or similar software to conduct cryptojacking on infected devices.

Although cryptomining has potential benefits, the emergence of cryptojacking attacks presents significant challenges for both companies and end-users. Cryptojacking is the unauthorized use of a victim's computing power to mine cryptocurrency [34]. This type of cyber attack is often complex to detect because it operates covertly, especially when it is executed in the form of micro mining[1]. Unlike other cyber threats that result in direct economic losses or ransom demands, cryptojacking attacks do not always have an immediate impact on the victim.

Since there is no immediate financial loss, victims typically

[1] https://www.investopedia.com/terms/m/micro-mining-cryptocurrency.asp, last visited on March 15, 2024.

do not perceive these attacks as problematic. However, the implications of cryptojacking can be significant, especially for a resource-constrained device. Indeed, the performance of this kind of device can be significantly deteriorated by cryptojacking, introducing a direct impact on their daily activities. This not only degrades device performance but also impacts battery life and can shorten the device's lifespan. In addition, considering an office network, the unauthorized use of company resources for mining cryptocurrencies within corporate networks can result in employees making illicit financial gains at the expense of the company's finances. Indeed, cryptojacking poses a broader threat, endangering both individual users and organizations. Cybercriminals exploit the economic potential of cryptocurrency mining by distributing malware that installs mining processes on victim devices without their knowledge.

There are two main types of cryptojacking attacks: browser-based and malware-based. Browser-based cryptojacking refers to the illicit practice of leveraging users' computational resources to mine cryptocurrency while they visit a webpage. Coinhive [18] is a notable example of JavaScript-based mining software that can be seamlessly integrated into websites. It enables website owners to monetize their content by utilizing visitors' CPU power to mine cryptocurrencies (like Monero).

MadoMiner [26] is a malicious worm equipped with keylogger functionalities capable of mining Monero. Discovered in September 2018 by researcher J. Quinn, MadoMiner propagates through a malicious tool called `ZombieBoyTools.exe`, which installs a Dynamic-Link Library (DLL) file using known exploits. Upon execution, the DLL file downloads and concurrently runs two UPX-packed executables—`Install.exe` and `Mask.exe`. While `Install.exe` is responsible for worm propagation, `Mask.exe` orchestrates Monero mining on infected devices, effectively forming a botnet. Additionally, `Mask.exe` downloads additional modules, including `360Safe.exe`, which enhances malware resilience and stealthiness. Reports suggest that MadoMiner operators managed to earn 6015$ per month by utilizing 50% of the computational resources on infected systems. Communications among botmasters, miners (i.e., infected devices), and mining pools occur via the previously described Stratum protocol, operating over TCP without TLS encryption.

To conclude, the goal of this paper is to propose a new module for the IoT proxy [6] based on a machine-learning solution capable of distinguishing both forms of cryptojacking attacks. Identifying cryptojacking-related traffic is a crucial Indicator of Compromise (IoC) that enables proactive detection of compromised devices, even before malware signatures are included in antivirus databases. Detecting and blocking cryptojacking activity through network traffic analysis not only mitigates the strain on IoT devices but also serves as a deterrent against illegal mining communications.

III. Data set construction

The data set that we used to train our ML models[2] (see Section IV) is composed of four different sources:

- benign web traffic, consisting of web browsing sessions that we collected using different browsers (Chrome, Edge, and Firefox) on Windows — we extrapolate this data from a dataset [7] we previously generated for a previous work [8];
- flows generated by XMR-Stak using both cleartext and encrypted Stratum connections to prove that our approach is independent of the use of TLS;
- MadoMiner Stratum connections;
- Coinhive mining sessions.

To gather the necessary network traffic capture files (PCAPs), we used a PC and a Raspberry Pi 4 Model B[3] (specifications of both devices are listed in Table I). We used the PC to mine Monero coins via Coinhive using a variety of browsers, and we intentionally infected it with the Madominer malware to acquire its network traffic. Furthermore, we installed and executed XMR-Stak on the Raspberry PI to gather some Stratum traffic.

TABLE I
Specifications of our test platforms.

component	PC	Raspberry Pi
CPU	Intel® i7-1065G7	ARM® Cortex-A72
RAM	8 GiB	8 GiB
OS	Windows 10	Raspberry Pi OS 10
Python	3.8.10	
scikit-learn	1.1.0	
Tstat	3.1.1	
Tshark	3.0.6	

We used Wireshark, a widely utilized network protocol analyzer, to capture network traffic generated during the cryptomining and cryptojacking operations. We then cleaned the obtained network traffic capture files by removing the unnecessary connections, leaving only the mining ones. Table II reports the number of TCP flows per category.

TABLE II
Data set composition.

type	flows
benign (browser traffic)	140343
XMR-Stak, (encrypted) Stratum over TLS	6011
XMR-Stak, (plain) Stratum over TCP	1159
MadoMiner	701
Coinhive	634
total	148848

Finally, we used `tstat` [14], a network analysis tool, to extract various traffic statistics. In particular, we used all the 32 TCP statistics listed in Table III. The features marked with

[2] The adopted dataset is available on Kaggle at: https://www.kaggle.com/datasets/danielecanavese/cryptomining-data-set/, last visited on April 30, 2024.

[3] https://www.raspberrypi.com/products/raspberry-pi-4-model-b/, last visited on March 2, 2024.

'both directions' are available from both endpoints' points of view; for instance, we have at our disposal both the number of packets sent by the client (e.g., XMR-Stak) and by the server (e.g., the mining pool servers). To comply with GDPR regulations, we refrained from inspecting payload data and focused solely on analyzing TCP/IP header data. This approach ensures user privacy as no user data is processed in any way (the application layer and its payloads are not involved at all).

TABLE III
TCP FEATURES.

FEATURE	UNIT
# packets (both directions)	*packets*
# packets with payload (both directions)	*packets*
# retransmitted packets (both directions)	*packets*
# out of sequence packets (both directions)	*packets*
# packets with ACK set (both directions)	*packets*
# packets with ACK set and no payload (both directions)	*packets*
# packets with FIN set (both directions)	*packets*
# packets with RST set (both directions)	*packets*
# packets with SYN set (both directions)	*packets*
# payload bytes excluding retransmissions (both directions)	*bytes*
# payload bytes including retransmissions (both directions)	*bytes*
# retransmitted bytes (both directions)	*bytes*
flow duration	*ms*
relative time of first payload packet (both directions)	*ms*
relative time of last payload packet (both directions)	*ms*
relative time of first ACK packet (both directions)	*ms*
TCP connection correctly terminated	*boolean*

IV. EXPERIMENTAL RESULTS

This section details our findings about distinguishing a crypto-mining or crypto-jacking flow w.r.t. to some benign traffic. All our model training and testing experiments were conducted using the platform whose specifications are listed in Table I.

We tested different ML models for classification: decision trees, random forests, gaussian SVMs, and kNN. We then tested three different binary classification scenarios to be able to distinguish benign (web) traffic against the crypto-mining flows produced by XMR-Stak, MadoMiner [26], and Coinhive.

We randomly selected 70% of the flows for the training phase, while the remaining 30% were used for testing our models. The hyperparameter search was conducted using a simple grid search algorithm with 5-fold cross-validation since the training phase was fast enough to allow it.

A. XMR-Stak traffic

Table IV reports our four models' accuracy, AUC, and F-score for identifying XML-Stak flows. In this case, we recall that the XMR-Stak class contains Stratum over TCP and TLS flows. All the models perform well, with the only notable exception of the Gaussian SVM, which exhibits a lower AUC and F-score.

Figure 1 plots instead a chart of a decision tree (our best classifier for XMR-Stak) and how its AUC changes as more packets per flow are received. As foreseeable, the AUC tends to increase steadily as more packets i.e., data) is received. After only four exchanged packets, the AUC reaches about 99%.

TABLE IV
XMR-STACK CLASSIFICATION METRICS.

MODEL	ACCURACY [%]	AUC [%]	F-SCORE [%]
decision tree	99.76	99.19	97.51
random forest	99.67	99.54	95.78
gaussian SVM	98.02	81.61	75.64
kNN	99.67	98.57	96.59

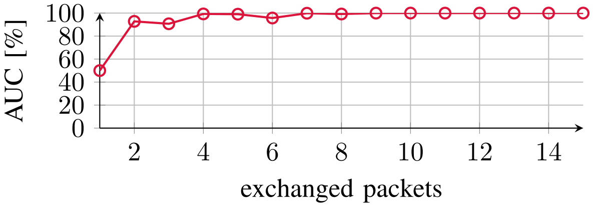

Fig. 1. XMR-Stack decision tree: AUC vs exchanged packets.

B. MadoMiner traffic

Table V reports several metrics of our MadoMiner classifiers. MadoMiner covertly installs XMRIG, a legitimate miner, in the victim's system and performs the mining by opening a series of Stratum over TCP connections (non-encrypted). For this reason, the results we obtained are similar to our XMR-Stak classifiers. In particular, even in this case, the decision tree seems to be the most effective one.

TABLE V
MADOMINER CLASSIFICATION METRICS.

MODEL	ACCURACY [%]	AUC [%]	F-SCORE [%]
decision tree	99.96	98.67	95.70
random forest	99.76	98.38	80.32
gaussian SVM	99.60	66.19	44.58
kNN	99.93	95.81	93.01

Figure 2 reports the AUC of our trained decision tree vs the number of exchanged packets per flow. After only three packets, the detection of this malware stabilizes at about 99%.

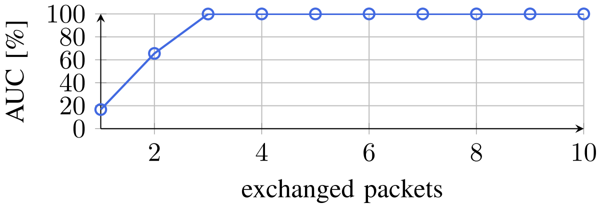

Fig. 2. MadoMiner decision tree: AUC vs exchanged packets.

C. Coinhive traffic

Table VI lists the classification metrics for our Coinhive models. Coinhive is harder to detect than XMR-Stak and Madominer, as the F-score of our classifiers is significantly lower than in the previous scenarios. This is most likely because Coinhive does not directly open a Stratum connection (which is easy to spot by our models) but embeds such protocol inside a WebSocket connection, which is much more similar to a traditional HTTP/HTTPS flow.

TABLE VI
COINHIVE CLASSIFICATION METRICS.

MODEL	ACCURACY [%]	AUC [%]	F-SCORE [%]
decision tree	99.75	91.04	74.46
random forest	98.48	94.09	34.79
gaussian SVM	99.31	55.23	11.82
kNN	99.74	82.76	69.28

Analogously, Figure 3 shows the AUC trend plot for our Coinhive decision tree, our best model. In this case, we reach a stable AUC only after about ten packets. This slower increase is possibly due to the similarity of a WebSocket and HTTP connection in their early beginning.

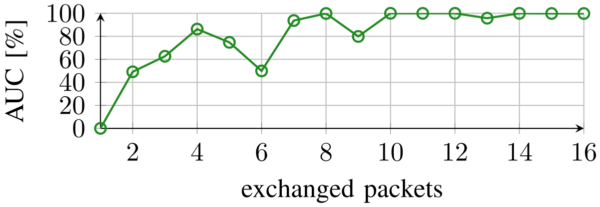

Fig. 3. Coinhive decision tree: AUC vs exchanged packets.

V. Related Work

Since the boom of cryptocurrencies, scientific literature has started to explore many different ways of detecting cryptomining connections. As highlighted by a recent survey [24], the detection of cryptomining activities has garnered significant attention in recent years. Most of the existing approaches primarily focus on detecting cryptomining software modules running on victim devices by analyzing some hardware metric (e.g., CPU consumption) or analyzing network traffic to identify suspicious connections. Machine Learning (ML) seems a highly promising approach in this area and many scientific works use this approach for high detection rates.

Huang et al. [17] used bidirectional Gated Recurrent Unit (GRU) neural networks with an attention mechanism to analyze temporally the packet flows and identify cryptomining connections. This approach, however, might be unsuitable to perform live analysis in networks where the throughput is particularly high since it can introduce a significant delay.

Mu noz et al. [22] evaluated machine learning techniques to identify non-encrypted cryptomining Stratum flows, achieving promising results in distinguishing cryptomining flows from regular traffic. However, their approach only considers non-encrypted traffic, limiting its effectiveness in real-world scenarios where encrypted traffic is prevalent. Swedan et al. [30] proposed an architecture for detecting and blocking non-encrypted cryptomining flows produced by browsers, but their reliance on Deep Packet Inspection (DPI) makes it ineffective against encrypted communications.

In contrast, Carlin et al. [9] and Liu et al. [21] employed dynamic analysis techniques to detect in-browser cryptominers, achieving high accuracy in classifying mining and non-mining scripts. These approaches leverage machine learning models to analyze browser behavior and identify cryptomining activities based on opcode counts and memory analysis. Additionally, Kharraz et al. [19] developed Outguard, an ML-based monitoring agent able to detect cryptojacking websites by analyzing browser behavior and computing simple numerical features.

Despite these advancements, detecting cryptojacking activities in IoT environments remains a challenge due to the resource constraints of IoT devices and the diverse nature of cryptojacking attacks. This motivates the need for innovative solutions, like the IoT proxy, that can effectively detect and mitigate cryptojacking threats in IoT networks while minimizing resource overhead and preserving device performance.

VI. Conclusions

In response to the escalating threat landscape posed by cryptojacking attacks within smart home environments, this paper extends the security capabilities of the *IoT Proxy* [6]. By integrating a dedicated cryptojacking detection module, this enhanced version of the IoT Proxy offers users a robust defense mechanism against this pervasive threat.

The IoT Proxy serves a dual purpose in fortifying smart home security. Firstly, it addresses the inherent limitations of resource-constrained IoT devices by providing essential security functionalities, thereby mitigating potential risks without compromising device performance. Secondly, it adheres to the security-in-depth principle by furnishing additional layers of defense even for less constrained devices, thereby ensuring comprehensive protection across the smart home ecosystem.

With the integration of the cryptojacking detection module, the IoT Proxy is empowered to identify and mitigate cryptojacking threats effectively. This proactive stance not only safeguards the integrity and performance of individual IoT devices but also bolsters the overall security posture of the smart home network. Given the heterogeneous nature of smart homes, encompassing a myriad of computing devices, extending security measures across the network environment is imperative to combat the evolving threat landscape effectively.

To evaluate the efficacy of our approach, we conducted experiments employing various machine learning models, including decision trees, random forests, Gaussian Support Vector Machines (SVMs), and k-Nearest Neighbors (kNN). Notably, our methodology eschews reliance on Deep Packet Inspection (DPI), thus preserving user privacy and ensuring applicability even in the face of encrypted traffic—a prevalent characteristic of modern network communication.

In conclusion, the integration of a cryptojacking detection module into the IoT Proxy marks a significant stride toward enhancing the security posture of smart home environments. By leveraging machine learning techniques and prioritizing user privacy, our solution provides a robust defense against cryptojacking threats, underscoring the importance of proactive security measures in safeguarding IoT ecosystems.

Looking ahead, there are several avenues for future research and development. Firstly, enhancing the scalability and efficiency of the cryptojacking detection module to accommodate the diverse and dynamic nature of smart home networks remains a priority. Exploring novel machine learning algorithms and techniques to bolster the detection capabilities and adaptability of the IoT Proxy represents a promising area for further investigation. In addition, to further enhance the effectiveness of the IoT Proxy, we are considering to add it the capability of using the Manufacturer Usage Description (MUD) [20] following the approach proposed in [11]. Introducing a module capable of reducing the possible incoming and outcoming network connection would help in detecting and stopping even cryptojacking data flows. Furthermore, integrating real-time threat intelligence and collaboration mechanisms to facilitate information sharing and collective defense strategies among IoT devices can augment the resilience of smart home environments against emerging cyber threats.

Acknowledgments

The authors thank Michele Maiullari for his valuable contributions to this research activity during his master's thesis.

References

[1] 2024 SonicWall Cyber Threat Report. Technical report, SonicWall. https://www.sonicwall.com/medialibrary/en/white-paper/2024-cyber-threat-report.pdf.

[2] Global Risks Report 2024. Technical report, World Economic Forum. https://www.weforum.org/publications/global-risks-report-2024/.

[3] Global Threat Report. Technical report, VMware Carbon Black. https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/docs/vmwcb-report-gtr-extended-enterprise-under-threat-global.pdf.

[4] Internet Organised Crime Threat Assessment (IOCTA) 2021. Technical report, Europol, 2021. https://www.europol.europa.eu/publications-events/main-reports/internet-organised-crime-threat-assessment-iocta-2021.

[5] Fernando Bruegge, Mohammad Hasan, Matthieu Kulezak, Knud Lasse Lueth, Eugenio Pasqua, Satyajit Sinha, Philipp Wegner, Kalpesh Baviskar, and Anand Taparia. State of IoT—spring 2023. Technical report, IoT Analytics GmbH, Astra Tower Zirkusweg 2, 20359 Hamburg, Germany, May 2023.

[6] Daniele Canavese, Luca Mannella, Leonardo Regano, and Cataldo Basile. Security at the Edge for Resource-Limited IoT Devices. *Sensors*, 24(2), 2024.

[7] Daniele Canavese, Leonardo Regano, Cataldo Basile, Gabriele Ciravegna, and Antonio Lioy. Data set and machine learning models for the classification of network traffic originators. *Data in Brief*, 41, 2022. Cited by: 3; All Open Access, Gold Open Access, Green Open Access.

[8] Daniele Canavese, Leonardo Regano, Cataldo Basile, Gabriele Ciravegna, and Antonio Lioy. Encryption-agnostic classifiers of traffic originators and their application to anomaly detection. *Computers & Electrical Engineering*, 97:107621, 2022.

[9] Domhnall Carlin, Philip O'Kane, Sakir Sezer, and Jonah Burgess. Detecting cryptomining using dynamic analysis. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–6. IEEE, 2018. Belfast (UK), 28-30 agosto.

[10] Fulvio Corno, Luigi De Russis, and Luca Mannella. Helping novice developers harness security issues in cloud-IoT systems. *Journal of Reliable Intelligent Environments*, 8(3):261–283, 2022.

[11] Fulvio Corno and Luca Mannella. A gateway-based MUD architecture to enhance smart home security. In *2023 8th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–6, Split/Bol, Croatia, June 2023. IEEE.

[12] Fulvio Corno and Luca Mannella. Security evaluation of arduino projects developed by hobbyist IoT programmers. *Sensors*, 23(5), 2023.

[13] Adrienne Porter Felt, Richard Barnes, April King, Chris Palmer, Chris Bentzel, and Parisa Tabriz. Measuring https adoption on the web. page 1323 – 1338, 2017. Cited by: 120.

[14] Alessandro Finamore, Marco Mellia, Michela Meo, Maurizio M. Munafo, Politecnico Di Torino, and Dario Rossi. Experiences of Internet traffic monitoring with Tstat. *IEEE Network*, 25(3):8–14, maggio 2011.

[15] fireice uk. XMR Stack: Free Monero RandomX Miner and unified CryptoNight miner. [Online] https://github.com/fireice-uk/xmr-stak, Accessed: February 13, 2024.

[16] JSON-RPC Working Group. Json-rpc 2.0 specification. [Online] https://www.jsonrpc.org/specification/, Accessed: February 2, 2024.

[17] Yijie Huang, Wei Ding, and Yuxi Cheng. Cryptomining Traffic Detection Based on BiGRU and Attention Mechanism. In *2023 7th International Conference on Cryptography, Security and Privacy (CSP)*, pages 35–40, April 2023.

[18] Troy Hunt. Coinhive. [Online] https://github.com/troyhunt/Coinhive, Accessed: February 14, 2024.

[19] Amin Kharraz, 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 *World Wide Web Conference*, WWW '19, pages 840–852. ACM, 2019.

[20] Eliot Lear, Ralph Droms, and Dan Romascanu. Manufacturer Usage Description Specification. RFC 8520, March 2019.

[21] J. Liu, Z. Zhao, X. Cui, Z. Wang, and Q. Liu. A novel approach for detecting browser-based silent miner. In *IEEE 3rd International Conference on Data Science in Cyberspace (DSC)*, pages 490–497, 2018.

[22] J. Muñoz, J. Suárez-Varela, and P. Barlet-Ros. Detecting cryptocurrency miners with NetFlow/IPFIX network measurements. In *IEEE International Symposium on Measurements Networking*, pages 1–6, 2019.

[23] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. [Online] https://git.dhimmel.com/bitcoin-whitepaper/, Accessed: February 2, 2024, 2008.

[24] Otavio Kiyatake Nicesio and Adriano Galindo Leal. A systematic literature review of machine learning approaches for in-browser cryptojacking detection. In *2023 7th Cyber Security in Networking Conference (CSNet)*, pages 102–108, Oct 2023.

[25] Antonio Pastor, Alberto Mozo, Stanislav Vakaruk, Daniele Canavese, Diego R. López, Leonardo Regano, Sandra Gómez-Canaval, and Antonio Lioy. Detection of encrypted cryptomining malware connections with machine and deep learning. *IEEE Access*, 8:158036 – 158055, 2020. Cited by: 37; All Open Access, Gold Open Access, Green Open Access.

[26] James Quinn. Madominer part 1 - install. [Online] https://www.alienvault.com/blogs/labs-research/madominer-part-1-install, Accessed: February 12, 2024, Setptember 2018.

[27] Ruben Recabarren and Bogdan Carbunar. Hardening stratum, the bitcoin pool mining protocol. *Proceedings on Privacy Enhancing Technologies*, 3:54–71, 2017.

[28] Clifton L Smith. Understanding concepts in the defence in depth strategy. In *IEEE 37th Annual 2003 International Carnahan Conference onSecurity Technology, 2003. Proceedings.*, pages 8–16. IEEE, 2003.

[29] Ali Sunyaev. *Distributed Ledger Technology*, pages 265–299. Springer International Publishing, 2020.

[30] AbedAlqader Swedan, Ahmad Khuffash, Othman Othman, and Ahmed Awad. Detection and prevention of malicious cryptocurrency mining on internet-connected devices. In *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, ICFNDS '18, pages 23:1–23:10. ACM, 2018.

[31] Braiins Systems. Stratum mining protocol. [Online] https://slushpool.com/help/topic/stratum-protocol/, Accessed: February 2, 2024.

[32] Braiins Systems. Stratum v2 — the next generation protocol for pooled mining. [Online] https://stratumprotocol.org/, Accessed: February 2, 2024.

[33] Pinyaphat Tasatanattakool and Chian Techapanupreeda. Blockchain: Challenges and applications. In *2018 International Conference on Information Networking (ICOIN)*, pages 473–475, Chiang Mai, Thailand, Jan 2018. IEEE.

[34] Ege Tekiner, Abbas Acar, A. Selcuk Uluagac, Engin Kirda, and Ali Aydin Selcuk. Sok: Cryptojacking malware. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 120–139, Sep. 2021.

[35] Wade Trappe, Richard Howard, and Robert S. Moore. Low-Energy Security: Limits and Opportunities in the Internet of Things. *IEEE Security & Privacy*, 13(1):14–21, 2015.