POSTER: Content-Agnostic Identification of Cryptojacking in Network Traffic

Yebo Feng
University of Oregon
yebof@cs.uoregon.edu

Devkishen Sisodia
University of Oregon
dsisodia@cs.uoregon.edu

Jun Li
University of Oregon
lijun@cs.uoregon.edu

ABSTRACT

In this paper, we propose a method that detects cryptojacking activities by analyzing content-agnostic network traffic flows. Our method first distinguishes crypto-mining activities by profiling the traffic with fast Fourier transform at each time window. It then generates the variation vectors between adjacent time windows and leverages a recurrent neural network to identify the cryptojacking patterns. Compared with the existing approaches, this method is privacy-preserving and can identify both browser-based and malware-based cryptojacking activities. Additionally, this method is easy to deploy. It can monitor all the devices within a network by accessing packet headers from the gateway router.

CCS CONCEPTS

• **Networks** → **Network monitoring**; • **Security and privacy** → **Intrusion/anomaly detection and malware mitigation**.

KEYWORDS

cryptojacking, anomaly detection, network traffic classification

ACM Reference Format:
Yebo Feng, Devkishen Sisodia, and Jun Li. 2020. POSTER: Content-Agnostic Identification of Cryptojacking in Network Traffic. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS '20), October 5–9, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3320269.3405440

1 INTRODUCTION

Cryptocurrencies have become popular, in large part due to their tremendous value gains and innovative concepts. One such concept is cryptocurrency mining (crypto-mining), which not only provides a means to verify cryptocurrency transactions and generate new cryptocurrency, but more importantly, helps establish consensus, which is critical in any blockchain-based system. In fact, with computing resources and electricity, plus an Internet connection, people can easily transform their computers, or any Internet-connected devices, into cryptocurrency-making machines by having them conduct cryptocurrency mining.

Unfortunately, the lucrative potential of crypto-mining has caught the attention of hackers. Hackers have been compromising personal computers, servers, or even Internet-of-Things (IoT) devices, such as smart TVs, to mine cryptocurrencies (e.g., Bitcoin, Monero) [9]. Such activity is called *cryptojacking*, which is the unauthorized use of someone else's computing resources to mine cryptocurrency. Hackers usually conduct cryptojacking by getting the victim to click on a malicious link in an email that downloads crypto-mining code onto their computer, infecting a website with JavaScript code that automatically runs itself once downloaded into a victim's browser, or compromising the servers to stealthily run the mining programs in the background. Although Coinhive, an in-browser mining service provider, shutdown on March 2019, cryptojacking is still alive and evolving [8]. Recently, The Next Web reported that unsuspecting victims were cryptojacked 52.7 million times in the first half of 2019 [5]. CoinDesk also asserted that hackers infected 50,000 servers with sophisticated crypto-mining malware [1], causing enormous economic loss to the virtual server providers.

To detect those malicious activities and protect users' computing resources, researchers have proposed various cryptojacking detection approaches. Hong et al. proposed CMTracker [4], a cryptojacking detection tool that monitors the browser behaviors and discovers cryptojacking scripts based on hash-based and stack-based profiling. Tahir et al. proposed a machine learning solution based on hardware-assisted profiling of browser code to detect cryptojacking [7]. Darabian et al. studied the potential of using deep learning techniques to detect crypto-mining malware by utilizing both static and dynamic analysis approaches [2]. However, all these approaches require access to the end-users' private data such as browser behaviors, system metrics, or binary code of the software. Additionally, they cannot tackle malware obfuscations.

In this paper, we seek to detect the traffic generated by cryptojacking software in a network by analyzing only the network traffic flows. The traffic flow is metadata of network operations and contains no content information from the sender. Recently, detecting application-layer anomalies with network traffic flows has become a trend [3], because such approaches are content-agnostic, which can protect user privacy while simultaneously detecting potential malicious activities in a network. With this feature, our approach is privacy preserving compared with the aforementioned cryptojacking detection solutions. Moreover, our approach is easy to deploy and use because users can install the detection system at their network's gateway, such as the border router of a network or a main networking switch, to monitor all the devices in their network.

To achieve the goal, we propose a two-phase detection procedure, which identifies crypto-mining traffic first, then detects cryptojacking activities. In the first phase, our approach captures the packet headers and converts them to a representation in the frequency domain with fast Fourier transform. We preset the profiles of crypto-mining traffic in the frequency domain and select the incoming traffic that matches these profiles as crypto-mining traffic. In the

ASIA CCS '20, October 5–9, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-6750-9/20/10.
https://doi.org/10.1145/3320269.3405440

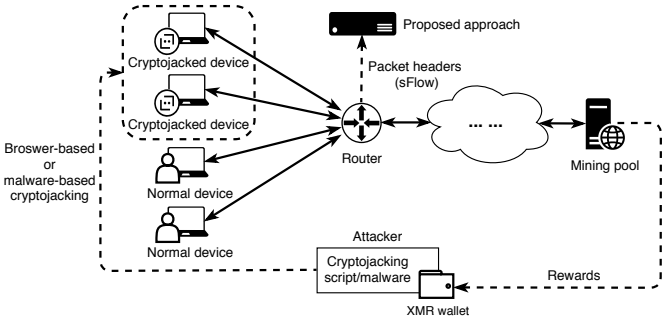

Figure 1: Operational model of the proposed approach.

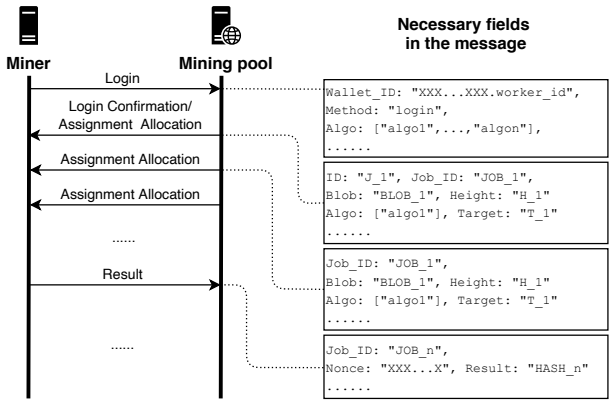

Figure 2: Network messages for mining.

second phase, our approach infers the hash-rate variations by analyzing the results from the first phase, then leverages a recurrent neural network to detect the cryptojacking activities.

2 METHODOLOGY

Both the network administrators and users can deploy our cryptojacking detection approach at any router or switch between the devices and the mining pools. Figure 1 shows the operational model of the proposed approach. The user needs to input certain fields in the IP packet headers to the detection system, which are the source and destination IP addresses, the source and destination ports, the packet size, and the timestamp. These data are information in the network for packet forwarding and contain no private content data of the senders. To obtain these content-agnostic data, we install sFlow in the router or switch, other packet capturing engines like NetFlow and Bro are also compatible with our detection system. Additionally, we design and evaluate the detection approach on the top of Monero-related (XMR) cryptojacking activities, since XMR is the dominated cryptocurrency that widely mined by cryptojacking attackers [4].

As stated in Section 1, the proposed method detects cryptojacking activities in two phases. The first phase inputs sFlow data and outputs identified crypto-mining traffic. The second phase inputs crypto-mining traffic and outputs detected cryptojacking traffic.

2.1 Phase One: detection of crypto-mining

In Phase One, the detection of crypto-mining traffic relies on the identification of the traffic between miners and the mining pool.

A mining pool is the pooling of resources by miners. As the mining difficulty keeps increasing, miners can hardly mine cryptocurrencies only by themselves. Hence, miners throughout the Internet form massive mining pools to conduct the mining tasks together and split the rewards according to their contribution. The communication between the miners and the mining pool should follow particular protocols or regulations so that miners from different locations can join the mining pool easily. Stratum is the most widely used communication protocol for mining [6], which has observable traffic patterns in the network. However, the cryptojacking attackers may harness some unpopular mining pools with unknown communication protocols, making the traffic easy to escape naïve flow-level detection approaches.

To make the detection of crypto-mining traffic robust, we study the communication principles between the miners and the mining pool and present the results in Figure 2. No matter what types of protocols the mining pool utilizes, there should be at least four types of messages to cover the necessary operations:

- *login message*, enabling the miners to join the mining pool, can be as small as 75 bytes per message;
- *login confirmation message*, confirming the login status, sometimes comes with an assignment allocation;
- *assignment allocation message*, allocating the most recent mining task to the miner, should at least have 285 bytes;
- *result message*, returning the calculated result to the mining pool, usually has more than 120 bytes.

The same type of messages usually have similar lengths. Besides, the login message and the confirmation message only appear once during each connection. Thus, assignment allocation messages and result messages dominate the whole mining process. Moreover, the mining pool will adjust the sending rate of assignment allocation messages to ensure the miner always have valid unfinished tasks. The blockchain is also growing continuously, making the allocated assignments easy to expire. Therefore, the frequency of assignment allocation messages is significantly higher than the frequency of result messages.

To profile these features and further leverage them to identify the crypto-mining traffic, we apply fast Fourier transform (FFT) to rapidly convert packets from the time domain to a representation in the frequency domain. Traffic generated from other activities, such as browsing webpage, DNS queries, and Telnet remote controlling, tends to have complicated and randomized frequency patterns. Conversely, mining traffic has clean and periodic frequency patterns, making it easy to be identified by matching the frequency profiles with preset threshold values.

We define a sliding time window to monitor the ongoing traffic. During each time window t, we collect a set of packets P ($P = \{p_1, p_2, p_3, ..., p_n\}$) from the same source IP and source port number to the same destination IP and destination port number, then represent them in a domain of time and number of bytes. Once this time window is reaching the end, we use FFT to represent P

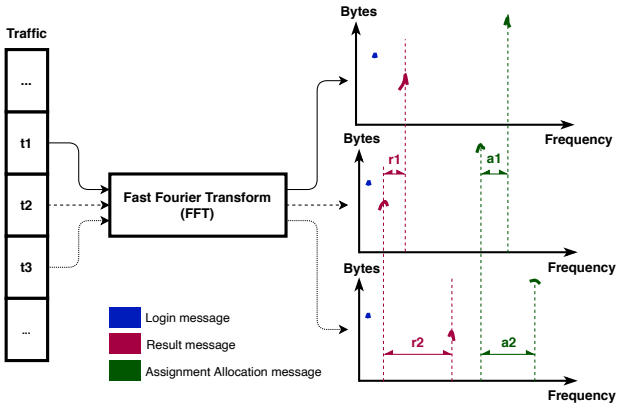

Figure 3: Generation of the variation vectors. The frequency curves in this figure only show the relevant peaks. Other irrelevant parts are removed in the preprocessing step.

in a domain of frequency and number of bytes. We call this representation as the *profile* of P. In the end, we identify P as packets generated by crypto-mining if they have a similar profile with the predefined crypto-mining profile.

2.2 Phase Two: detection of cryptojacking

In Phase Two, the proposed method inputs the detected crypto-mining traffic and output the identified cryptojacking traffic among them.

An essential concept of crypto-mining is the hash rate, the speed at which a device is completing an operation in the crypto-mining code. We found that a higher hash rate will trigger higher frequencies of result messages and assignment allocation messages. Furthermore, after studying the cryptojacking activities, we found that they differ from legitimate crypto-mining activities in the following aspects:

- the hash rate of legitimate crypto-mining is more stable than the hash rate of cryptojacking because cryptojacking scripts usually rely on some existing software running in the system such as the browser, terminal, or Apache server, which makes the computing resources devoted to the mining calculation erratic;
- the hash rate of cryptojacking is usually lower than the hash rate of legitimate crypto-mining, since cryptojacking scripts or malware cannot easily invoke GPU or dedicated ASIC chips to mining, further leading to a lower message rate.

With these intuitions, we extract the variation vectors from different time windows to profile the changes in hash rates. For time window t_n and t_{n+1}, we generate a variation vector v_n ($v_n =< r_n, a_n >$) to describe the changes in frequencies of result messages (r_n) and assignment allocation messages (a_n). Figure 3 shows an example of the variation vector generation, where we derive two variation vectors from time window t_1, t_2, and t_n. r_n is the absolute difference between the result message frequencies in t_n and t_{n+1}. a_n is the absolute difference between the assignment allocation message frequencies in t_n and t_{n+1}.

Our approach collects variation vectors as time-series data, then inputs these vectors to a pre-trained recurrent neural network (RNN) model to distinguish cryptojacking traffic from legitimate crypto-mining traffic. Since there are no existing cryptojacking traffic datasets available in public repositories, we simulate both legitimate crypto-mining traffic and cryptojacking traffic to train the model. We will publish the dataset we use when the project is finished.

3 CONCLUSION AND FUTURE WORK

Cryptojacking attacks are becoming far more sophisticated and threatening than before. To solve this problem, we propose a privacy-preserving cryptojacking detection approach that only relies on content-agnostic network traffic flows to conduct detections. It applies a two-phase procedure to identify cryptojacking traffic, which first selects crypto-mining traffic by profiling the message frequency, then analyzes the frequency variances to recognize cryptojacking patterns.

Our approach is efficient and easy to deploy. With the computing power of a personal computer, it is capable of providing real-time detection of cryptojacking for a company-level network.

In the future, we will keep polishing and testing this approach in both simulated and realistic environments. To enhance the robustness of our approach, we will simulate the cryptojacking activities under different hardware, software, and network environments. Besides, we want to evaluate this approach comprehensively by measuring the system overheads, the detection accuracies, and the compatibility with different network infrastructures.

ACKNOWLEDGMENTS

This material is based upon work supported by Ripple Graduate Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Ripple Labs, Inc.

REFERENCES

[1] Benedict Alibasa. 2019. Hackers Infect 50,000 Servers With Sophisticated Crypto Mining Malware. https://www.coindesk.com/hackers-infect-50000-servers-with-sophisticated-crypto-mining-malware.
[2] Hamid Darabian, Sajad Homayounoot, Ali Dehghantanha, Sattar Hashemi, Hadis Karimipour, Reza M Parizi, and Kim-Kwang Raymond Choo. 2020. Detecting Cryptomining Malware: a Deep Learning Approach for Static and Dynamic Analysis. *Journal of Grid Computing* (2020), 1–11.
[3] Yebo Feng, Jun Li, Lei Jiao, and Xintao Wu. 2019. BotFlowMon: Learning-based, Content-Agnostic Identification of Social Bot Traffic Flows. In *IEEE Conference on Communications and Network Security (CNS)*.
[4] Geng Hong, Zhemin Yang, Sen Yang, Lei Zhang, Yuhong Nan, Zhibo Zhang, Min Yang, Yuan Zhang, Zhiyun Qian, and Haixin Duan. [n.d.]. How you get shot in the back: A systematical study about cryptojacking in the real world. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*.
[5] Yessi Bello Perez. 2019. Unsuspecting victims were cryptojacked 52.7 million times in the first half of 2019. https://thenextweb.com/hardfork/2019/07/24/cryptojacking-cryptocurrency-million-hits-first-half-2019/.
[6] Ruben Recabarren and Bogdan Carbunar. 2017. Hardening stratum, the bitcoin pool mining protocol. *Proceedings on Privacy Enhancing Technologies* 3 (2017), 57–74.
[7] Rashid Tahir, Sultan Durrani, Faizan Ahmed, Hammas Saeed, Fareed Zaffar, and Saqib Ilyas. 2019. The browsers strike back: countering cryptojacking and parasitic miners on the web. In *IEEE Conference on Computer Communications*.
[8] Said Varlioglu, Bilal Gonen, Murat Ozer, and Mehmet F Bastug. 2020. Is Cryptojacking Dead after Coinhive Shutdown? *arXiv preprint arXiv:2001.02975* (2020).
[9] Aaron Zimba, Zhaoshun Wang, Mwenge Mulenga, and Nickson Herbert Odongo. 2018. Crypto mining attacks in information systems: An emerging threat to cyber security. *Journal of Computer Information Systems* (2018), 1–12.