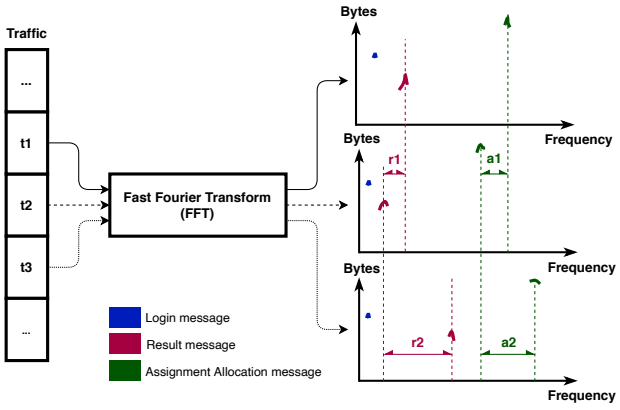

Figure 3: Generation of the variation vectors. The frequency curves in this figure only show the relevant peaks. Other irrelevant parts are removed in the preprocessing step.

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.