Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

An Insight into Deep Learning based Cryptojacking Detection Model

S. S. Sivaraju

Professor, Department of Electrical and Electronics Engineering, R V S College of Engineering and Technology, Coimbatore, Tamil Nadu, India

E-mail: sssivaraju@gmail.com

Abstract

To autonomously identify cyber threats is a non-trivial research topic. One area where this is most apparent is in the evolution of evasive cyber assaults, which are becoming better at masking their existence and obscuring their attack methods (for example, file-less malware). Particularly stealthy Advanced Persistent Threats may hide out in the system for a long time without being spotted. This study presents a novel method, dubbed CapJack, for identifying illicit bitcoin mining activity in a web browser by using cutting-edge CapsNet technology. Thus far, it is aware that deep learning framework CapsNet is pertained to the problem of detecting malware effectively using a heuristic based on system behaviour. Even more, in multitasking situations when several apps are all active at the same time, it is possible to identify fraudulent miners with greater efficiency.

Keywords: Deep Learning (DL), CapsNet, cyber threats, cryptojacking, mining detection, cryptocurrency.

1. Introduction

The degree to which a cyber system can identify and categorise the programmes and services hosts as a key factor in establishing the system's security independence. If the system is crucial to the operation of the business, it must swiftly adapt to new security threats via learning and retraining, regardless of how much processing power it has.

1.1 Overview of Crypto-Mining

By combining the processing power of many individuals, cryptomining pools make the formerly inefficient process of mining more democratic and efficient. Systems participating in a crypto mining pool are basically virtual employees; their employment

DOI: https://doi.org/10.36548/jtcsst.2022.3.006
Received: 08.08.2022, received in revised form: 26.08.2022, accepted: 11.09.2022, published: 21.09.2022

involves finding an appropriate constant to solve a new hash. This workforce is compensated in the same cryptocurrency that was used to create the new block.

Several cyberattacks and security breaches have their origins in software vulnerabilities [1], and the current solutions for detecting evasive malware and other vulnerabilities rely on static rule-based systems and machine learning systems that rely on human experts or static heuristics for feature selection (the factors that are important to make an accurate prediction). The feature selection process is sometimes subjective and biased when performed by human specialists. To solve this problem, it is ideal to have many humans participating in the feature selection process, each of whom may discover their own emotional qualities that can successfully train the model and then choose the best features from among those identified [2-4]. However, the problem's intricacy suggests that fixing it may require a lot of physical labour. Furthermore, since Advanced Persistent Threats (APTs) analyse system defences and devise novel methods to circumvent them, rule-based systems and heuristic-informed feature selection generally fail to keep up with them as they evolve. Figure 1 shows type of cryptojacking basic methods.

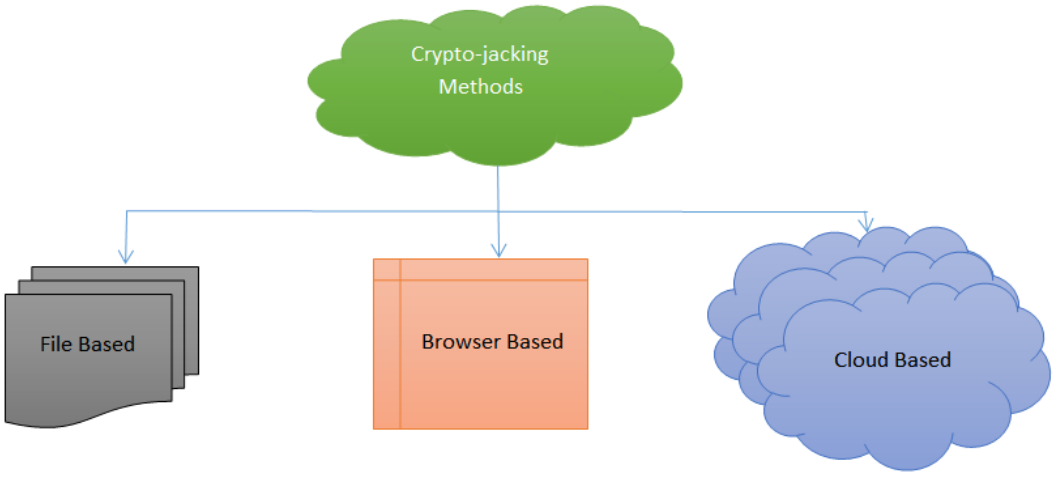

Figure 1. Types of Cryptojacking Methods

1.2 Conventional Malware Detection Methods

The signature matching and heuristic detection are the backbones of conventional malware detection methods, although rule matching isn't very good at generalizing or detecting novel malware [5]. However, machine learning methods have rapidly advanced in recent years to become an additional critical approach to malware identification. Since deep learning can successfully predict labels in high-dimensional spaces, it also has an advantage in the substantial space search issue [6, 7]. When it comes to malware detection, however,

feature selection is still governed by the extent to which the final training model can generalise.

The researchers in certain malware detection investigations employed a data-driven feature to create malware detectors, such as features based on the byte level [8] or the PE format [9]. The conventional malware analysis group, on the other hand, has a different take on how to classify malware based on its behaviour; therefore, this kind of feature design approach is at odds with their thinking. Also, several studies on malware rely significantly on methodologies from the more conventional area of machine learning, with practices like random splitting and small datasets as the gold standards for test dataset development. These testing approaches fail because, in practice, malware detectors are trained on limited datasets before being put to the test in a scenario with numerous malware samples [10-13].

1.3 Motivation

The results of this exploratory investigation and the knowledge gained from it have inspired to implement the most recent version of the Capsule Network (CapsNet). Biological neuronal organisation has been closely mimicked by a machine learning system recently suggested in numerous research publications. The design is inspired by research showing that accurate categorization and object identification depend on maintaining hierarchical pose connections between object pieces. CapsNet achieves this by augmenting a convolution neural network with structures called capsules and using dynamic routing to link capsules in order to quantitatively express tight interactions between objects as a pose matrix. One of its many strengths is that it can identify overlapping items with high accuracy. In contrast to CNN, coinciding digits may now be detected, as shown in the landmark work [14].

2. Literature Survey

Examples of evasive malware include cryptomining malware (in which cybercriminals discreetly install cryptocurrency miners on victims' computers and exploit their CPU resources to mine cryptocurrencies) and ransomware (in which cybercriminals encrypt victims' files and then demand a ransom). Both assaults use cryptographic calculations and encryption to hide their true nature while pretending to be useful tools like compression or encoding programmes. Crypto mining malware, also known as cryptojacking, is chosen as the example of stealthy malware in this study because of the additional risk it poses i.e., in addition to imitating legitimate programmes in order to steal processing power, cryptojacking can cause excessive CPU usage, which in turn slows down the system and

prevents mission-critical systems from carrying out their duties. The mobile platform is not immune to cryptojacking [15]. In addition, mobile devices are vulnerable to thermal damage because of their fragile, non-tamper-proof construction. The practice of cryptojacking is spreading rapidly. More than 8 million crypto mining malware assaults [16] were discovered by Symantec in only three months between December 2017 and February 2018.

When it comes to APT, cryptomining malware is a newly uncovered subtype. The use of deep learning to identify crypto mining malware is widespread [17]. The sequences of system calls used in these models are taken from real-world crypto mining malware samples distributed in Portable Executable (PE) format. However, crypto mining malware using complex PoW algorithms like CryptoNight may mask system calls and modify call sequences to remain undetected. CapJack [18] employed a Deep Capsule Neural Network (effective for modelling hierarchical relationships and a version of CNN) with the following features: Central Processing Unit (CPU) consumption, memory utilisation, disc read/write, and network interface. To complete, the solution makes heavy use of CPU, random access memory cache selection, disc reading/writing, and communication. Malware designed for cryptomining may limit CPU utilisation, limiting the frequency with which the device reads and writes data. Lightweight machine learning approaches, in addition to deep learning, have been presented in the literature as methods of discovering crypto mining malware. Outguard employed Wrappalyzer's categorised collection of crypto mining malware libraries (a web-based detection tool). Solution libraries were categorised as either harmful or safe using a Support Vector Machine (SVM) model. The approach is limited in its ability to detect evasive APTs like cryptojacking due to the fine-grained nature of the library analysis. In [19], a classifier based on the Random Forest algorithm was suggested.

2.1 Concise Statement of the Issue

Will advanced machine learning methods be able to detect and distinguish crucial details that the human eye misses? When taking the CPU and other system variables into account, is there any noticeable improvement? This initial, very stern attempt to address these issues was machine learning. The use of Naive Bayes and Decision Trees, two machine learning methods, have been investigated in prior studies.

2.2 Questions for Further Study

The article's key research elements and issues are as follows:

1. Explores in depth how with high detection accuracy and little implementation overhead, image-based categorization of emerging malware may be utilised as a formidable defence against adversarial assaults.
2. Two novel assaults against DL-based malware classifiers are able to bypass cutting-edge DL-based malware detectors without compromising the modified malware's functionality.

3. Types of Cryptojacking

The similarities and distinctions between the various forms of crypto-jacking malware are discussed in this section.

3.1 Browser-based Cryptographic Hijacking

The evolution of web technology is lightning fast. For instance, web developers may communicate with computer components by issuing a few commands in the browser using JavaScript (JS) programming language libraries and WebAssembly (Wasm) open standards. Crypto-jacking malware is implemented in the browser with the help of these technologies by the attackers. For instance, all main browsers support Wasm, which enables the execution of low-level instruction codes at near-native rates in the browser [20].

3.2 Cryptojacking the Host Computer

The goal of host-based cryptojacking malware is to secretly mine cryptocurrencies using the victim's computer system. The cryptojacking software continues to generate profits as long as it runs on the victim's computer, which is the key purpose. There are three distinct kinds of dangers. In the risk analysis, it takes into account both browser-based and native crypto mining malware. Without permission from the user, the mining process is monitored and evaluated. A native mining programme, for instance, may be installed with the user's blessing, but it could secretly mine cryptocurrency without their knowledge or permission. In a similar vein, websites may harbour troubled youths, either knowingly or unknowingly on the part of the site's proprietor.

3.3 CapsNet based Mining Detection

Based on what it is found and what is learnt from their first investigation, [21] has decided to implement the most recent version of the Capsule Network (CapsNet). CapsNet operates in a different way than both CNN and KNN-MLL. To calculate posterior probability

for several classes, it first determines whether the sample has examples of each class's intellectual qualities. Those probabilities may be whatever is chosen, so long as their total equals 1. Cryptocurrency miners, along with other activities, may be thought of as spatial distributions of mixed data.

3.4 Model-Agnostic Mining Device Detection

CapsNet has been shown to be successful for smaller change identification, with a higher detection accuracy in many feature extraction processes. Although this seems to be promising, it should be noted that in the preceding explanation, the framework consists of both training and validation of the neural networks. The fact is that people are not only using a wide variety of devices but also often use devices from various generations. In an ideal world, training a CapsNet works across all platforms. It shows that, experiments in which training and testing data come from the same device tend to perform poorer than those in which training data comes from one device and testing data comes from a various framework with the same CapsNet model [22].

3.5 Advancing a Plan of Action

When fresh data is introduced to a trained CapsNet, it produces a 16 x 5 matrix or a set of 5 vectors. Each vector's length, which represents the likelihood of occurrence of the relevant application, is determined through vector components. Because of this, it has a 15 grid of possibilities. Each sample generates its own unique probability array. The existence of a miner is used to assign labels to the probability arrays. This allows to efficiently train an SVM using just a minimal amount of training data [23].

4. Comparative Observations

4.1 Possible Consequences of a Larger Application Pool

It is generally accepted that, as the number of classes in a machine learning model grows, its classifiable accuracy will get smaller. While it is reasonable to assume that the same concept applies to CapJack, it would be helpful to do a quantitative research to fully grasp the scheme's stability. To do so, it tests the miner detection rate by changing the maximum allowed number of hybrid apps. The suggested framework through technique scales smoothly with the increasing complexity of hybrid software. When it gathers the testing data samples, many other processes (including those that are part of the operating system) operate in the background concurrently. The suggested method seems to be very

resistant to such confounding variables. The similar trend is seen across device types, however the total detection rate drops (from 90% to 81%). The miner identification accuracy follows a similar pattern to that seen on PCs, illustrating the generalizability of the suggested approach. In addition, it can reach 100% detection accuracy when it uses a window-based technique. However, spotting cryptomining malware is not always easy. Numerous methods of detecting cryptojacking rely on excessive CPU use and overheating as their key detection triggers [23].

The following techniques and characteristics of applications may help them remain undetected:

The use of individual system components is managed by both native apps and browser-based miners. For instance, they could decide on a maximum allowable percentage of CPU use. They may avoid detection by system safeguards by limiting their CPU utilisation.

4.2 Functionality observation

They may hide their actions by beginning the mining process, while seeming like harmless programmes, such as compression or encoding programmes that are active. Compression and encoding apps are computationally complex procedures that might result in significant CPU utilisation, much as crypto mining malware, since they execute bitwise and encryption operations and cryptographic calculations. The number of false positives or false negatives produced by this depends on the capabilities of the system's protection measures.

Some browser-based miners engage in a practice known as "drive-by mining," in which they remain in the system for a very little period of time before moving on to the next victim. Interval mining is also available to miners based on native applications; in this mode, miners are dormant for a period of time and then activate for a brief mining and communication period with the major mining pool. During these brief mining sessions, anti-virus and other security measures have little chance of detecting the infection. Crypto mining malware is difficult to categorize and detect since it does not damage a system's functionality or show any obvious symptoms.

5. Conclusion

This research presents a novel method, called CapJack, for detecting illegal cryptocurrency mining operations by using cutting-edge CapsNet technology. This deep

learning framework has to specifically use CapsNet for malware identification. It's a useful tool for malevolent miners who sometimes have many mining programmes running at the same time. Following the success of the CapsNet-based approach, two-tiered classification system capable of quickly modifying a trained model to detect miners using unproven gear has been developed. To be useful in real life, a system must be able to adapt to a wide variety of possible victim devices. A well-engineered and working prototype is the product of the team's efforts.

References

[1] H. Geng, Y. Zhemin, Y. Sen, Z. Lei, N. Yuhong, Z. Zhibo, Y. Min, Z. Yuan, Q. Zhiyun, and D. Haixin, "How you get shot in the back: A systematical study about cryptojacking in the real world," in Proceedings of the ACM Conference on Computer and Communications Security (CCS), 2018.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Proceedings of Advances in Neural Information Processing Systems (NIPS), pp. 1097–1105, 2012.

[3] S. Sara, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules," in Proceedings of Advances in Neural Information Processing Systems (NIPS), pp. 3856–3866, 2017.

[4] R. Recabarren and B. Carbunar, "Hardening stratum, the bitcoin pool mining protocol," Proceedings on Privacy Enhancing Technologies Symposium (PETS), vol. 2017, no. 3, pp. 57–74, 2017.

[5] R. Tahir, S. Durrani, F. Ahmed, H. Saeed, F. Zaffar, and S. Ilyas, "The browsers strike back: countering cryptojacking and parasitic miners on the web," in IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019, pp. 703–711.

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

[7] G. Mani, V. Pasumarti, B. Bhargava, F. T. Vora, J. MacDonald, J. King, and J. Kobes, "Decrypto pro: Deep learning based cryptomining malware detection using performance counters," in IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS). IEEE, 2020, pp. 109–118.

[8] H. Darabian, S. Homayounoot, A. Dehghantanha, S. Hashemi, H. Karimipour, R. M. Parizi, and K.-K. R. Choo, "Detecting cryptomining malware: a deep learning approach for static and dynamic analysis," Journal of Grid Computing, pp. 1–11, 2020.

[9] H. N. C. Neto, M. A. Lopez, N. C. Fernandes, and D. M. Mattos, "Minecap: super incremental learning for detecting and blocking cryptocurrency mining on software-defined networking," Annals of Telecommunications, pp. 1–11, 2020.

[10] G. Hong, Z. Yang, S. Yang, L. Zhang, Y. Nan, Z. Zhang, M. Yang, Y. Zhang, Z. Qian, and H. Duan, "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, 2018, pp. 1701–1713.

[11] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "Web-based cryptojacking in the wild," arXiv preprint arXiv:1808.09474, 2018.

[12] W. Wang, B. Ferrell, X. Xu, K. W. Hamlen, and S. Hao, "Seismic: Secure in-lined script monitors for interrupting cryptojacks," in European Symposium on Research in Computer Security, 2018, pp. 122–142.

[13] Pau Rodríguez, Miguel A Bautista, Jordi Gonzalez, and Sergio Escalera. 2018. Beyond one-hot encoding: Lower dimensional target embedding. Image and Vision Computing 75 (2018), 21–31.

[14] Jan Rüth, Torsten Zimmermann, Konrad Wolsing, and Oliver Hohlfeld. 2018. Digging into browser-based crypto mining. In Proceedings of the Internet Measurement Conference 2018. 70–76.

[15] Joshua Saxe and Konstantin Berlin. 2017. eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys. arXiv preprint arXiv:1702.08568 (2017).

[16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). 265– 283.

[17] Sergei Arnautov, Bohdan Trach, Franz Gregor, Thomas Knauth, Andre Martin, Christian Priebe, Joshua Lind, Divya Muthukumaran, Dan O'keeffe, Mark L Stillwell, et al. 2016. {SCONE}: Secure Linux Containers with Intel {SGX}. In 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). 689–703.

[18] Yara Awad, Mohamed Nassar, and Haidar Safa. 2018. Modeling malware as a language. In 2018 IEEE International Conference on Communications (ICC). IEEE, 1–6.

[19] J.-Y. Kim, S.-J. Bu, and S.-B. Cho, "Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders," Information Sciences, vol. 460- 461, pp. 83–102, 2018.

[20] S. Bose, T. Barao, and X. Liu, "Explaining ai for malware detection: analysis of mechanisms of malconv," in Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8, IEEE, Glasgow, UK, July 2020.

[21] E. Raffff, W. Fleshman, R. Zak, H. S. Anderson, B. Filar, and M. McLean, "Classifying sequences of extreme length with constant memory applied to malware detection," 2020, https://arxiv.org/abs/2012.09390.

[22] R. Vyas, X. Luo, N. McFarland, and C. Justice, "Investigation of malicious portable executable fifile detection on the network using supervised learning techniques." in Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 941–946, IEEE, Lisbon, Portugal, May 2017.

[23] S. Jeon and J. Moon, "Malware-detection method with a convolutional recurrent neural network using opcode sequences," Information Sciences, vol. 535, pp. 1–15, 2020.

Author's biography

S. S. Sivaraju is currently working as a Professor in the Department of Electrical and Electronics Engineering, R V S College of Engineering and Technology, Coimbatore, Tamil Nadu, India. His area of research electrical drives, renewable energy and deep learning algorithms.