Detecting Cryptojacking using ML

Rohan Sharma
Department of Cyber Security and Networks
Amrita Vishwa Vidyapeetham, Amritapuri
Kollam, India
amenp2csn21022@am.students.amrita.edu

***Abstract*—Cryptojacking means when an external entity is using your system's computational power for their benefit to mine cryptocurrency using your system resources. This attack is modeled in such a way that an attacker executes a script into your system and once the script is successfully executed it is using your system's computational power to mine cryptocurrency for the attacker. The proposed solutions to this problem are not suitable to fully protect the individual or corporates from this attack. In this paper, we propose a model to detect if the system is infected or not. We build the model using Machine Learning by understanding the behaviour of the malware on an infected machine. And according to this, we performed the experiment, and our model achieved a striking 0.99 F1 score. With this approach, we will be able to detect the malware more efficiently. Given the scope and originality of the issue under consideration, we hope that our methodology, backed by strong results, will pave the way for more research in this area.**

***Index Terms*—Cryptocurrency, Hijack, Malware, Processing power, Mining.**

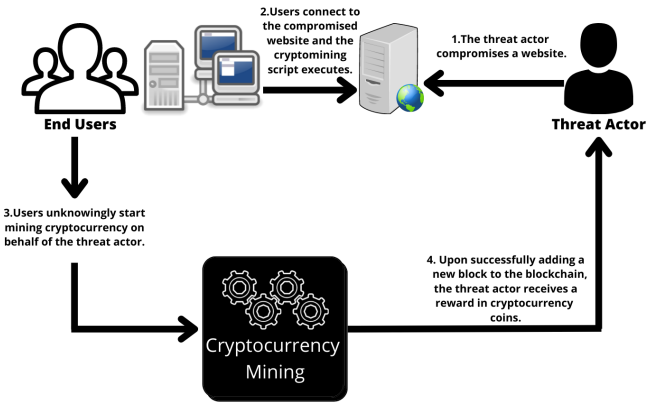

Fig. 1. Schematic Working of Cryptomalware.

I. Introduction

Blockchain is a new technology, and Bitcoin has piqued interest since its launch in 2009. The popularity of Bitcoin has grown to the point where certain financial organizations and governments have already included it in their monetary systems. There are almost 10,000 currency tokens available now, with around 2000 stable cryptocurrencies. According to a recent Kaspersky survey, approximately 20 percent of the world's population has purchased cryptocurrency and holds it in their crypto wallets. However, purchasing cryptocurrency is not the only option to invest in it, we can also mine it by dedicating our computing power to the blockchain of the relevant cryptocurrency and becoming a miner. Here comes the term "cryptojacking" this is used to describe a cyber-attack in which an attacker implants a malicious script on the target's machine and leverages their computational capacity to mine cryptocurrency for him. Cryptojacking is of two types one is an In-browser cryptojacking attack and another is the Script-based cryptojacking. In In-browser cryptojacking the attacker embeds the malicious script with their website using JavaScript and when the target visits the website the malicious code starts mining. And in scriptbased cryptojacking, the malware is sent to the target via any program, email, etc. and once it's downloaded onto your system it starts executing and starting the mining process this type of malware is much more difficult to detect because they are not making any harm to the targets system, they are just using their computational power to mine cryptocurrency and remain anonymous in the targets system. Just because of this reason only the crypto-malware are difficult to detect as traditional malware these are not harming or control the target system they just consume the computational power and send the calculated hash values back to the attacker. This decreases the efficiency and working of the electronic device as the malware is using the hardware power of the device to mine cryptocurrency and the owner may not be able to use the device as usual. This issue is that much severe that this practice is among the top 5 cybersecurity attacks in the world these days. We can check ***here*** for daily analysis of the cyberattacks in the world, Cryptojacking is amongst the top 5 cybersecurity attacks even if we take an example of India itself it is at the 3rd position in the different types of cyberattacks vectors. This threat's attack surface ranges from an individual to an organization, and none of us are secure. Anybody can be affected by this problem as this attack can be done on anybody starting from the common man to a multinational company. As this crypto malware can be added to any system or device in the world. This can be installed on a smartphone or up to a server or workstation of a company. Even the tesla servers are also hacked and have been served for the same purpose in the past. The attacker just needs the computational power from the hardware, this is how this attack is different from a ransomware attack as in this the attacker is only using the victim's hardware to generate money instead of in a ransomware attack the attacker has to demand money as a ransom. The victim who is using

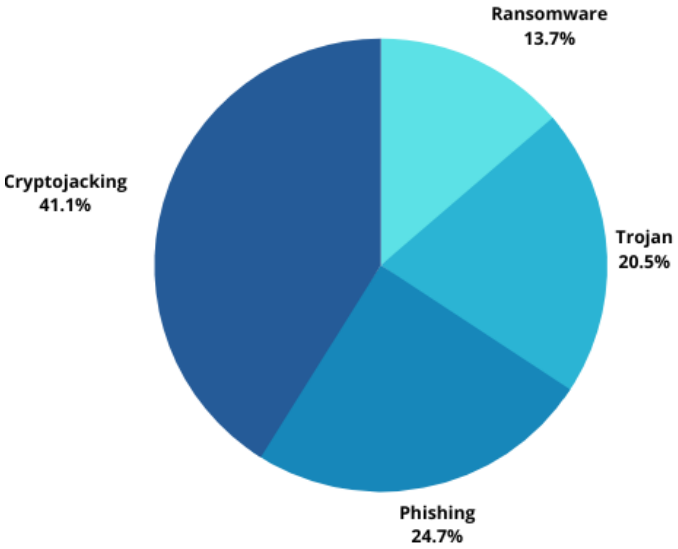

Fig. 2. Increase in the number of cryptojacking attacks in the recent past

the devices will experience a slower performance or lags in execution in their device, as its processing power is being used somewhere else so they will not be happy with the service of the device. This will impact the security firms, browser companies, and even the device manufacturers, so there are looking for solutions to this problem. Due to the seriousness of this emerging threat, we are proposing a solution to this threat as we will be building the model using machine learning and detecting the infected machine, and try to achieve the optimal scores from our experiment so that we can detect this malware more efficiently and faster. And further can be able to take required steps for mitigation.

II. Related work

The research conducted in this domain because of the rise in crypto jacking attacks has become a key research area for researchers, and several studies are currently underway to detect crypto-malware. Many methods are available, such as installing antivirus, updating the system, knowing about the current cryptojacking trends, installing browser extensions, setting up ad blockers, disabling JavaScript, etc. But what happens if a system becomes infected? How can you get to know? What should be the appropriate mitigation measures to be taken? Many studies are carried out for the analytical examination of crypto-malware,

- *In the "Cryptojacking Detection with CPU Usage Metrics" research paper, researchers examine the CPU metrics as to how a machine behaves when it got infected via crypto-malware.*
- *In "MINOS: A Lightweight Real-Time Cryptojacking Detection System" this research paper the researchers have proposed MINOS a novel cryptojacking malware detection method that uses image-based identification methods to differentiate between benign and malicious crypto-malware libraries.*
- *In "How You Get Shot in the Back: A Systematical Study about Cryptojacking in the RealWorld" this research paper researchers have designed "CMTracker" to recognize the malicious activities in the system automatically.*
- *In "MineThrottle: Defending against Wasm In-Browser Cryptojacking" in this research paper the researchers have introduced technique "Minethrottle" to detect illegal cryptomining through our web browsers.*
- *In "Cryptomining makes noise: Detecting cryptojacking via Machine Learning, " the researchers are working on building a robust model to detect and identify the crypto mining activities in the system. They have achieved great results an F1 score of 0.96 and an AUC for ROC greater than 0.99.*

We can see that a lot of different approaches are being introduced in identifying and detecting the cryptojacking activities in the system, and different models are developed on the basis of different metrics. We are also working on the different CPU metrics and analyzing the infected machine and normal machine and examining the different performance vectors and developing the model on that through our experiment trying to achieve the optimal scores for the cryptojacking malware detection with much high identification rate.

III. Methodology

A. Dataset

To analyze this issue, we found the data set at ***this*** location.

For training the model we need to trust the authenticity and integrity of the dataset, so that there will be no manipulation in the dataset, this data is presented from a website, and all the features were extracted by the author, and while analyzing these features we got to know that the dataset is dependent mainly on these benchmarks i.e., CPU, Per Core CPU usage, disk, File-system, Memory, Networking, Processes, System Identifiers, and Timestamp.

We are having two datasets; one is an infected dataset and another is a normal dataset. The infected dataset contains the 82 extracted features and 14460 samples and in the normal dataset, we have 82 features and 80800 samples. All the features belong to these categories:

CPU:

- *cpu_idle: when there is nothing to do, then the kernel just wastes this slice of time and remains idle to save power.*
- *cpu_steal: when we are in a virtualized environment then the hypervisor "steals" some of the CPU cycles, and gives them to some other programs within that virtual system.*
- *cpu_IO_wait: this is the time when CPU has nothing to do just has to wait for the result of a disk/ read/ write process to complete.*

• *cpu_nice: this means the priority of the execution, like whether it is executing in normal priority, below priority, or with higher priority.*
• *cpu_user: this means whether the cpu is running in user mode or application mode.*
• *cpu_system: what CPU is running - a kernel code, device driver, or other kernel modules.*
• *cpu_total: like how much core cpu we are having, how much total cpu processing power we have this is calculated by no. of cores × clock speed per core.*

Disk:
• *diskio_sda_disk_name: this means the name of the partition on which the processing data is executed.*
• *diskio_sda_key: this means on which partition the processing is being done if there are multiple partitions.*
• *diskio_sda_read_bytes: this is the data that is being read from the partition*
• *diskio_sda_write_bytes: this is the data that is written to the partition.*
• *diskio_sda_time_since_update: this is the time since the last reading was taken.*

File System(fs):
• *fs_/_devicename: this means the device model name and manufacturer.*
• *fs_/_free: this means how much free space is available.*
• *fs_/_fs_type: this means that which are the available file system types ext1, ext2, ext3, hpfs, jfs, ntfs, xfs.*
• *fs_/_key: where the file system has to be mounted.*
• *fs_/_mnt_points: pointer to the file system.*
• *fs_/_size: this means that what is the size of the file system.*
• *fs_/_used: this means that how much file system is used.*

CPU usage:
• *load_cpucore: load on each logical processor cores*
• *load_min1: this is the load decay exponentially after 1 min i.e., 60 seconds.*
• *load_min5: this is the load decay exponentially after 5 min*
• *load_min15: this is the load decay exponentially after 15 min*

Per Core CPU Usage:
• *percpu_0_IOwait: this is the time when cpu core has nothing to do just has to wait for the result of a disk/ read/ write process to complete*
• *percpu_0_nice: this means the priority of the execution of a core if it is executing in normal, below, or higher priority.*
• *percpu_0_user: this means whether the CPU core is running in user mode or application mode.*
• *percpu_0_system: if a core is executing a kernel instruction or application instruction.*
• *percpu_0_steal: when we are on a virtualized environment then the hypervisor "steals" some of the CPU core's cycles, and gives them to some other programs within that virtual system.*
• *percpu_0_total: maximum clock speed of a core.*

Memory:
• *mem_active: active memory means the memory used by a particular process.*
• *mem_inactive: inactive memory means the memory allocated to a process that's no longer running.*
• *mem_available: available memory means an estimate of how much memory is available for starting new applications without swapping.*
• *mem_buffers: it's the region of the memory used to temporarily store data while some data is being moved from one place to another*
• *mem_cached: it's the memory in which the computer application temporarily store data in the computer main memory to enable fast revivals of the data.,*
• *mem_free: it's the memory that contains no useful data and is just waiting to be used.*
• *mem_shared: memory that is shared between many programs to provide communication among them.*
• *mem_total: total installed memory in a system.*
• *mem_used: how much memory is currently being used.*
• *memswap_free: available free swap memory*
• *memswap_sin: how much memory swap is created.*
• *memswap_sout: how much memory is again converted back to normal memory*
• *memswap_total: total swap memory*
• *memswap_used: used swap memory*

Network :
• *network_IO_cumulative_cx: this is the total number of transmitted and received packets*
• *network_IO_cumulative_tx: this is the total number of transmitted packets*
• *network_IO_cumulative_rx: this is the total number of received packets.*
• *network_IO_interface_name: defines if the network interface is physical or virtual.*
• *network_IO_key: Network interface name.*
• *network_IO_time_since_update: this is the time since the last network log occurred.*

Process :
• *proccesscount_runing: it's the count of running processes in the system at a given time,*
• *proccesscount_sleeping: number of sleeping processes in the system at a given time*
• *proccesscount_thread: it's the number of threads per process in the system*
• *proccesscount_total: total number of processes in a system.*

System Identifiers :

• *system_hostname: what a device is called on the network*
• *system_hr_name: which system OS we are using*
• *system_linux_distro: name of the Linux distribution of the system*
• *system_OS_name: Linux OS name of the system*
• *system_OS_version: Linux OS Version of the system*
• *system_platform: x86 or x86_64 architecture of the system*

Timestamp :

• *This data is collected in real-time so this dataset includes a feature based on timestamp.*

This feature we are neglecting because time series analysis is useful for short-term forecasting, but it could sometimes lead to wrong predictions. This is because it requires historical data to construct the models, which means that if some significant changes occurred over time, then those changes will not be included within the forecasted periods. Another major reason for dropping this feature is that because it acts as an anomaly, it does not give the exact project value rather than a probability distribution on possible future outcomes. This means that we can forecast how probable specific values will be reached but not what they are exactly going to be.

B. Experiments Run

We were having two datasets, i.e normal and anormal dataset. Normal dataset compromises of the CPU metrics of a normal, non-infected machine and anormal dataset compromises of the CPU metrics when a machine got infected with crypto malware. We merged both the datasets and created our final dataset as "Cryptojacking_.csv"

1) We have extracted our crypto final dataset.
2) Panda library is used to load the dataset.
3) We are dropping the irrelevant features.
4) We are checking the dataset for missing values.
5) We type casted our target feature as int. i.e. "infected_"
6) We found 30 missing values and we imputed them by taking mean of those columns.
7) We are splitting the dataset into feature and target set.
8) We dropped the "timestamp" feature.
9) We have done the splitting of dataset into X_, X_, y_, y_.
10) We have created the logistic regression model for predicting values on the basis of our training dataset.
11) We have used the confusion matrix and generated the classification report for predicted values in logistic regression.
12) We have evaluated the accuracy, f1-score, precision, recall score for logistic regression model.
13) We have done the AUC plotting for logistic regression model.
14) We have created the decision tree classifier for predicting values on the basis of our training dataset.
15) We have used the confusion matrix and generated the classification report for predicted values for the decision tree classifier.
16) We have evaluated the accuracy, f1-score, precision, recall score for decision tree classifier.
17) We have done the AUC, heatmap plotting for decision tree classifier.
18) We have created the KNN classifier for predicting values on the basis of our training dataset.
19) We have used the confusion matrix and generated the classification report for predicted values for the KNN classifier.
20) We have evaluated the accuracy, f1-score, precision, recall score for KNN classifier.
21) We have done the AUC plotting for KNN classifier.

We have taken the F1-score as our primary metric. F1-score is used as it the combination of recall and precision. It is used to distinguish between the two classifiers.
Precision is used as a secondary metric as it will reduce the count of false positive, malware justified as not a malware.

IV. Results and Analysis

We have analysed the metrics by using numerous classifiers, models, and confusion metrics and have predicted values for precision, accuracy, recall, and F1 score. We are comparing the results of the models with each other.

After getting the predicted scores from the models we concluded that our model is able to achieve a good result when compared with the existing solutions. In my perspective, no model will be able to achieve 100 percent of the outcome since it will overfit the model we can say that our model predicted

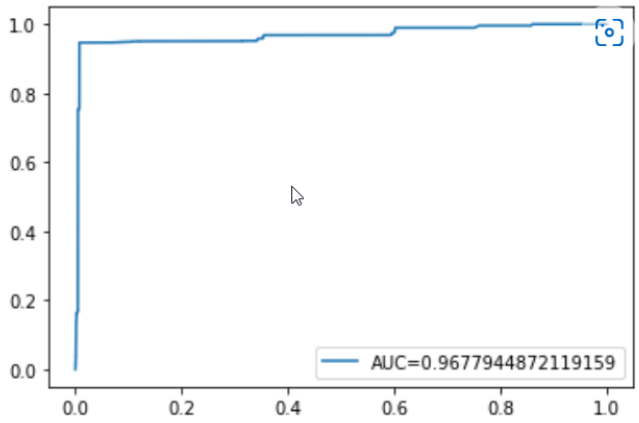

Fig. 3. AUC for Confusion matrix

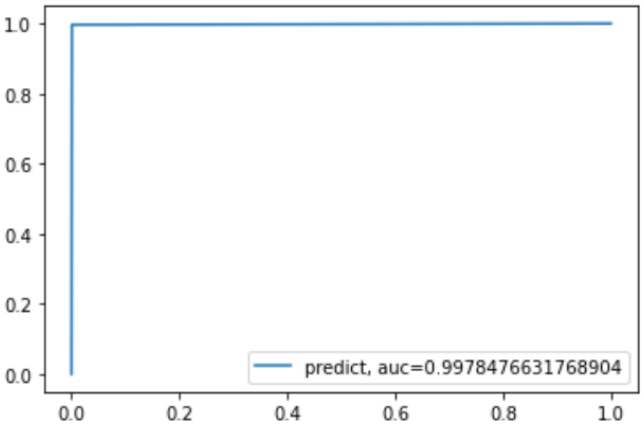

Fig. 4. AUC for Decision Tree Classifier

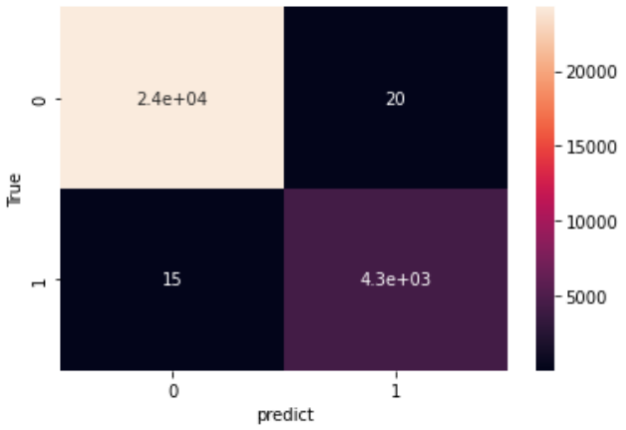

Fig. 5. Heatmap for Decision Tree Classifier

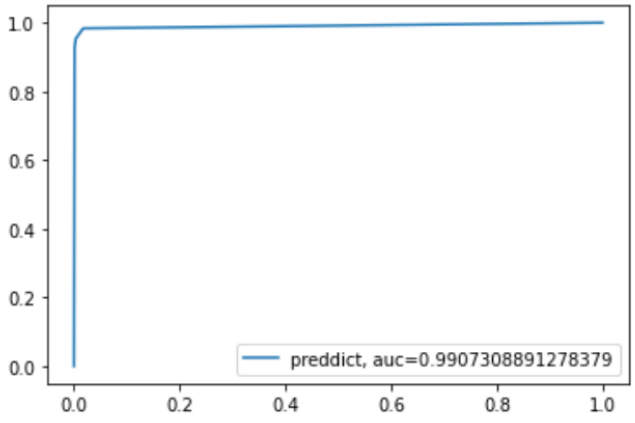

Fig. 6. AUC for Decision KNN Classifier

TABLE I
COMPARISON TABLE SCORE FOR LOGISTIC REGRESSION, DECISION TREE CLASSIFIER AND KNN .

Scores	Logistic Regression	DTC	KNN
Recall	46.76 %	99.58%	95.35%
Precision	94.37%	99.44%	97.99%
Accuracy	91.55%	99.85%	99.00%
f1-score	62.53%	99.51%	96.65%

good value and detecting and identifying the crypto mining activities in the system.

V. CONCLUSION

Due the severity of this emerging threat and the challenges described above, many cryptojacking detection models have been built. In this paper, we worked with dataset merged from two datasets, as we know that running cryptojacking shows a discernible pattern on CPU metrics. So the dataset has been made according to the CPU metrics and changes. The first dataset is based on the CPU metrics when the system is infected with crypto-malware and another dataset from a normal machine. The model performed well which is built on these datasets, and we achieve a 99.51 percent f1-score from decision tree classifier for detecting the crypto-malware in the system. The evaluation of our approach using static analysis of cryptojacking malware, on the other hand, will be left for future study.

REFERENCES

[1] Naseem, F., Aris, A., Babun, L., Tekiner, E., Uluagac, S. (2021, February). MINOS: A lightweight real-time cryptojacking detection system. In 28th Annual Network and Distributed System Security Symposium, NDSS.
[2] Xavier, S. N. (2020). Machine Learning Approaches to Detect Browser-Based Cryptomining (Doctoral dissertation, Dublin, National College of Ireland).
[3] Hayes, C. (2021). The Evolution of Cryptojacking (Doctoral dissertation, Utica College).
[4] Hong, G., Yang, Z., Yang, S., Zhang, L., Nan, Y., Zhang, Z., ... Duan, H. (2018, October). 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 (pp. 1701-1713).

[5] Caprolu, M., Raponi, S., Oligeri, G., Di Pietro, R. (2021). Cryptomining makes noise: Detecting cryptojacking via Machine Learning. Computer Communications, 171,126-139.
[6] Bian, W., Meng, W., Zhang, M. (2020, April). Minethrottle: Defending against wasm in-browser cryptojacking. In Proceedings of The Web Conference 2020 (pp. 3112-3118).
[7] Hernandez-Suarez, Aldo Sanchez-Perez, Gabriel Toscano, Karina Olivares Mercado, Jesus Portillo-Portilo, Jose Avalos, Juan-Gerardo García Villalba, Luis. (2022). Detecting Cryptojacking Web Threats: An Approach with Autoencoders and Deep Dense Neural Networks. Applied Sciences. 12. 3234. 10.3390/app12073234.
[8] Maurantonio Caprolu, Simone Raponi, Gabriele Oligeri, Roberto Di Pietro, Cryptomining makes noise: Detecting cryptojacking via Machine Learning,(2021) Computer Communications, Volume 171, ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2021.02.016.
[9] Sachan, Rohit Agarwal, Rachit Shukla, Sandeep. (2022)." DNS based In-Browser Cryptojacking Detection".
[10] Petrov, I., Invernizzi, L., Bursztein, E. (2020). Coinpolice: "Detecting hidden cryptojacking attacks with neural networks". arXiv preprint arXiv:2006.10861.