International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 08 | Aug - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

A Review on Data Driven Models for Identifying Cryptojacking Malwares

Rajesh Chouhan[1], Prof. Sanmati Jain[2]
VITM, Indore, India.[1,2]

ABSTRACT: With the rise of popularity of cryptocurrencies such as Bitcoin, Libra, Ripple, Ethereum etc, more attacks on crypto-currencies have been seen. Crypto-jacking is the unauthorized use of someone else's computer to mine cryptocurrency. Hackers do this by either getting the victim to click on a malicious link in an email that loads crypto-mining code on the computer, or by infecting a website or online ad with JavaScript code that auto-executes once loaded in the victim's browser. Apart from the crypto-jackers, web-crypto-mining library providers also enabled benign publishers to use this mechanism as an alternative monetization scheme for extracting revenue. The cryptojacking attacks have risen significantly with respect to conventional malwares and phishing attacks due to the relatively low risk and high reward nature of cryptojacking attacks. The nature of the attack also makes it extremely complex to detect such attacks. Hence, machine learning based techniques have been at the forefront of detecting cryptojacking. This paper presents the related work in the domain with its salient points and also lays down a methodology for the implementation of a proposed approach for detection of possible cryptojacking attacks.

Keywords: Data-mining, Darkweb, Crypto Mining Crypto currency: Bitcoin; Crypto-jacking; Machine Learning

I. INTRODUCTION

Crypto currency is an internet-based medium of exchange which uses crypto graphical functions to conduct financial transactions. Crypto currencies can be sent directly between two parties via the use of private and public keys. These transfers can be done with minimal processing fees, allowing users to avoid the steep fees charged by traditional financial institutions.

Examples of crypto currencies: Bitcoin, Libra, Ripple, Ethereum etc. The most important feature of a crypto currency is that it is not controlled by any central authority: the decentralized nature of the block chain makes cryptocurrencies theoretically immune to the old ways of government control and interference. Cryptocurrencies can be sent directly between two parties via the use of private and public keys. These transfers can be done with minimal processing fees, allowing users to avoid the steep fees charged by traditional financial institutions. Some of the common crypto-currencies which are popular worldwide are shown in the figure below with their symbols.

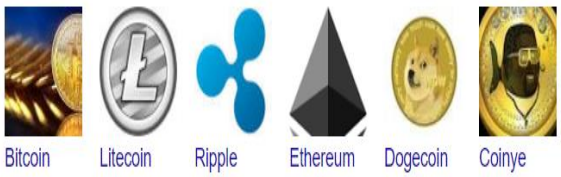

Fig.1 Common Cryptocurrencies

The cryptocurrencies have the following attributes:

1) They are not legalized by several governments.
2) They however are used as proxy currencies in dark-web transactions.
3) They can be used by users to remain anonymous and yet use the currency.
4) They can be earned by solving complex problems available on the dark web.
5) Transactions comprising of cryptocurrencies are generally carried out based on peer-to-peer networks.

The architecture for cryptocurrency based transaction is shown in the figure below:

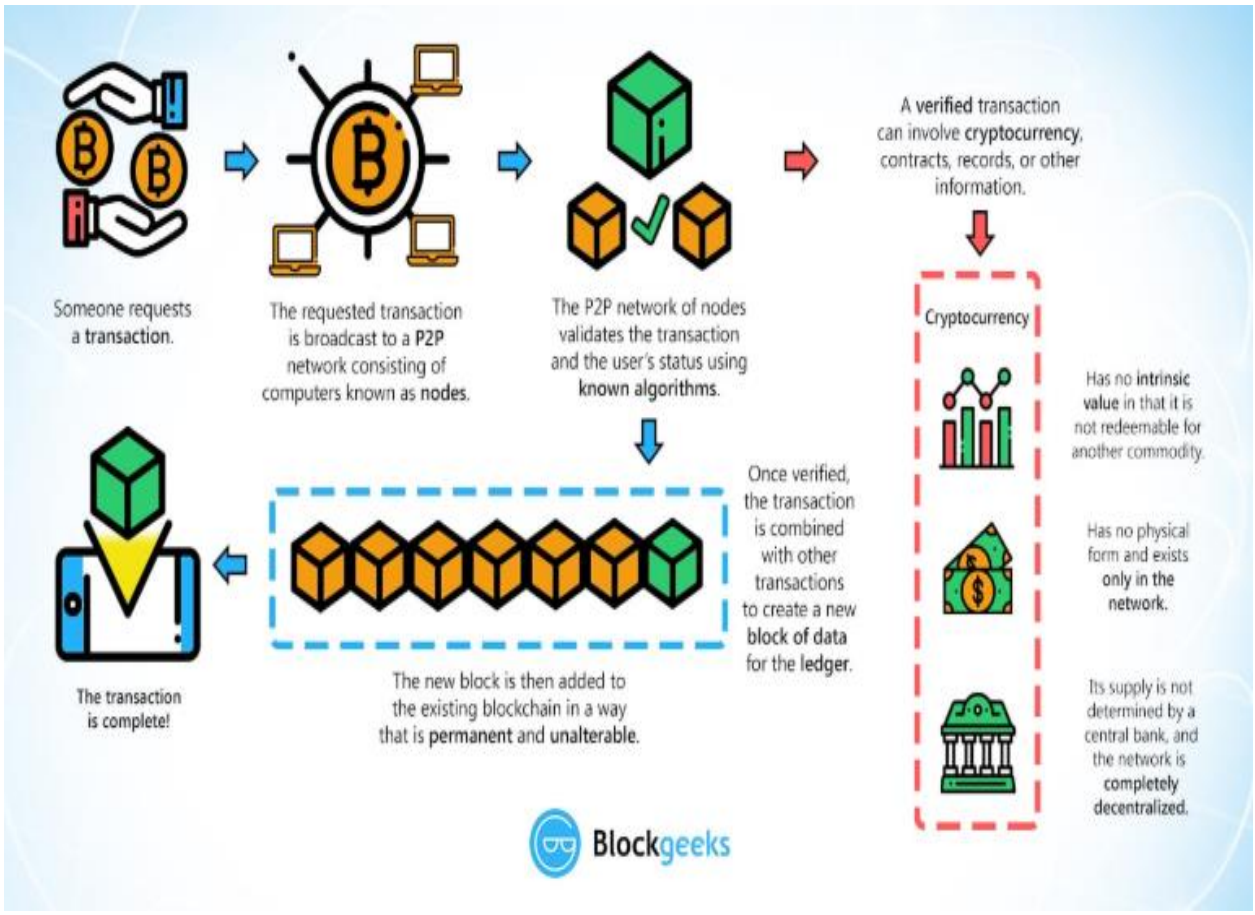

Fig.2 Cryptocurrency based per-to-peer transaction architecture

II. CRYTOMINING AND CRYTPOJACKING

Cryptocurrency mining, or crypto mining is generally a subset of web mining. It is a process in which transactions for various forms of cryptocurrency are verified and added to the block chain digital ledger. Also known as crypto coin mining, altcoin mining, or Bitcoin mining (for the most popular form of cryptocurrency, Bitcoin), cryptocurrency mining has increased both as a topic and activity as cryptocurrency usage itself has grown exponentially in the last few years. Each time a cryptocurrency transaction is made, a cryptocurrency miner is responsible for ensuring the authenticity of information and updating the block chain with the transaction. The mining process itself involves competing with other crypto miners to solve complicated mathematical problems with cryptographic hash functions that are associated with a block containing the transaction data. Cryptojacking is utilizing a host system to illegally mine cryptocurrencies without the knowledge of the host system. Web-based mining is a method of cryptocurrency mining that happens inside a browser, using a script delivered from a website. The first attempts of in-browser such as Bitcoin mining failed due to the increased mining difficulty. However, the rise of alternative crypto-coins (altcoins) that provide distributed mining, increased mining speed and ASICs, has again made it viable to implement crypto mining and cryptojacking. The three facets of web mining are shown in the figure below for purpose of cryptojacking:

Fig.3 Subsets of Web Mining

Web content and web usage mining generally render information regarding novice and gullible web users.

III. PREVIOUS WORK

The previous work entails the work done previously for the detection of crypto mining.

Sanda et al. proposed Truth in Web Mining: Measuring the Probability and the Imposed Overheads of Cryptojacking. — Given the lower-risk nature of cryptojacking, the number of such incidents were nearly double of those of ransomware attacks. Apart from the cryptojackers, web-crypto mining library providers also enabled benign publishers to use this mechanism as an alternative monetization schema for web in the era of declined ad revenues. In spite of the buzz raised around web-cryptomining, it is not yet known what the profitability of web-cryptomining is and what is the actual cost it imposes on the user side. In this paper, the overhead imposed to the user is analysed with regards to power consumption, resources utilization, network traffic, and device temperature and user experience. Those overheads along with the profitability of web-cryptomining to the ones imposed are compared by advertising to examine if web-cryptomining can become a viable alternative revenue stream for websites.

Gupta et al. have provided a Automated Discovery of JavaScript Code Injection Attacks in PHP Web Applications. In this paper they discussed some of the performance issues in the existing defensive solutions of Java Script injection attacks (e.g. Cross-Site Scripting (XSS) attacks). Moreover, a high level of comparison for such existing solutions has been done based on some useful metrics. Based on the identified performance issues, this paper proposed an automated detection system, which scans the numerous possible locations of web sites for JavaScript injection vulnerabilities. Their detection system, firstly, scans the web site for discovering the injection locations. Secondly, it injects the malicious XSS attack vectors

in such injection points. Lastly, it takes an input as the list of different XSS attacks exploited in the second step and scan for these attacks in the vulnerable web application. Detection capability of our automated system is evaluated on a real world PHP web application i.e. BlogIt and results obtained are very promising.

Duesse et al. have provided a Randomized Instruction Set Emulation to Disrupt Binary Code Injection Attacks. In this paper it is shown that Binary code injection into an executing program is a common form of attack. Most current defences against this form of attack use a 'guard all doors' strategy, trying to block the avenues by which execution can be diverted. We describe a complementary method of protection, which disrupts foreign code execution regardless of how the code is injected. A unique and private machine instruction set for each executing program would make it difficult for an outsider to design binary attack code against that program and impossible to use the same binary attack code against multiple machines. As a proof of concept, authors describe a randomized instruction set emulator (RISE), based on the open-source Valgrind x86-to-x86 binary translator. The prototype disrupts binary code injection attacks against a program without requiring its recompilation, linking, or access to source code. The paper describes the RISE implementation and its limitations, gives evidence demonstrating that RISE defeats common attacks, considers how the dense x86 instruction set affects the method, and discusses potential extensions of the idea.

Papadopoulo et al. have provided DroidCIA: A Novel Detection Method of Code Injection Attacks on HTML5-Based Mobile Apps. In this paper Smartphones have become more and more popular recently. There are many different smartphone systems, such as Android, iOS, etc. Based on HTML5, now developers can have a convenient framework to develop cross-platform HTML5- based mobile apps. Unfortunately, HTML5-based apps are also susceptible to cross-site scripting attacks like most web applications. Attackers can inject malicious scripts from many different injection channels. This new text box injection channel was not detected by other researchers so far because they only analysed JavaScript APIs, but overlooked HTML files which captured text box input type information. Later, they applied this new method to a vulnerable app set with 8303 cases obtained from Google Play. They detected a total of 351 vulnerable apps with accuracy 99%, which included 347 detected also by other researchers as well as 4 extra vulnerable apps that belonged to this text box injection channel. they also implemented a Code Injection Attack detection tool named DroidCIA that automated the drawing of JavaScript API call graph and the combination of API with HTML information.

Kara et al. analyzed Code Injection Attacks on HTML5-based Mobile Apps: Characterization, Detection and Mitigation. They mainly considered that due to the portability advantage, HTML5-based mobile apps are getting more and more popular. Unfortunately, the web technology used by HTML5-based mobile apps has a dangerous feature, which allows data and code to be mixed together, making code injection attacks possible. In this paper, they have conducted a systematic study on this risk in HTML5-based mobile apps. They found a new form of code injection attack, which inherits the fundamental cause of Cross-Site Scripting attack (XSS), but it uses many more channels to inject code than XSS. These channels, unique to mobile devices, include Contact, SMS, Barcode, MP3, etc. To assess the prevalence of the code injection vulnerability in HTML5- based mobile apps, we have developed a vulnerability detection tool to analyze 15,510 PhoneGap apps collected from Google Play. 478 apps are flagged as vulnerable, with only 2.30% false-positive rate. We have also implemented a prototype called NoInjection as a Patch to PhoneGap in Android to defend against the attack.

IV. PROBLEMS STATEMENT AND PROPOSED ARCHITECTURE

Typical crypto-attackers try to use the computational power of innocent web users to mine cryptocurrencies illegally (called crypto-jacking). However, it is extremely difficult to detect crypto-jacking attacks because of the following reasons:

1) The attack code is generally very small (compact)
2) The attack doesn't produce any significant effect.
3) Only some common issues like the machine slowing down, more CPU utilization or mild heating can be observed.
4) Hence most attacks go unnoticed.

Typical signs of a crypto-mining operation include increased CPU usage, degraded system performance, and sluggish application responsiveness. Demands imposed by crypto-mining can have serious consequences. "In one instance, crypto-mining software was known to destroy the device that hosted it. Even if signs of crypto-jacking appear on a system, finding the malware can be challenging.
System defenses that depend on software signatures and anomalies, such as modified files or system data, can struggle to identify crypto-mining malware when it lands on a network. Crypto-miners do not modify files, and their anomalous behavior is limited to increased CPU usage or power consumption that can be hard to attribute specifically to a crypto-miner, since there can be other applications—games, for instance—that tend to over-consume the processing capabilities of a system. Increased CPU usage is easier for an individual to recognize than it is for a typical enterprise

The approach would use standard benchmark data sets for testing.
The features are to be selected which are:

1) Temperature
2) Power
3) Network Speed
4) CPU Utilization
5) Memory Utilization
6) Interfering applications (interference)

The features are to be fed to a machine learning based tool to classify detect the attacks. The architecture of the machine learning to be used for the proposed system is shown in the figure below with the data collected from the benchmark datasets. The configuration of the machine learning can be either of the standard configurations but in cryojacking based scenarios, highly supervised machine learning algorithms may not work. The machine learning flow is shown in the figure below:

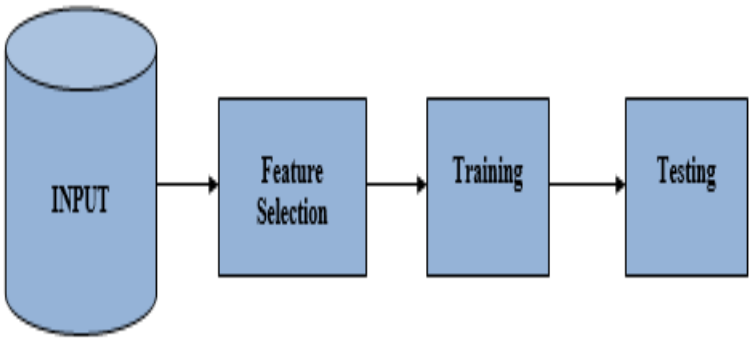

Fig.4 Machine learning based classification architecture

After the classification stage is over, the efficacy of the systems need to be evaluated based on certain parameters called features.

Sensitivity (Se): It indicates the algorithms ability to correctly detect the samples which test affirmative and belong to a particular category specified by feature values.

Accuracy (Ac): It indicates the algorithms ability to detect correctly whether a sample belongs to a particular data set category or not out of all the possible classification outcomes.
Thus, the above-mentioned parameters help in the evaluation of any proposed algorithm which deals with statistical data and statistical modelling. It is desirable to attain high values of sensitivity and accuracy. The performance metrics are defined as:

1. **True Positive (TP):** It is the categorization of a data sample into positive with correct prediction
2. **True Negative (TN):** It is the categorization of a data sample into negative with correct prediction.
3. **False Positive (FP):** It is the categorization of a data sample into positive with incorrect prediction
4. **False Negative (FN):** It is the categorization of a data sample into negative with incorrect prediction

Sensitivity (S_e): It is the comparative positive marker in the data set as how many samples are marked positive. Mathematically:

$$Se = \frac{TP}{TP + FN}$$

Accuracy (Ac): It is a measure of the correctness of classification prediction. It is the ratio of correct classifications to all classifications. Mathematically:

$$Ac = \frac{TP + TN}{TP + TN + FP + FN}$$

The aim would be to hit higher accuracy compared to previously existing systems.

CONCLUSION: It can be concluded from previous discussions that with the rise of popularity of cryptocurrencies such as Bitcoin etc, more attacks on crypto-currencies have been seen. Despite of the reduction in cryptocurrency prices, cryptojacking continues to be prevalent on the web due to the minimal effort it requires from the attackers. Malicious miners have shown up in mobile devices, cloud infrastructure, or even critical infrastructure. The proposed approach aims at using machine learning based techniques for detecting crypto-jacking attacks. A summary of the relevant concepts and previous work aids the formulation of the approach.

References

[1] O Sanda, M Pavlidis, N Polatidis, "A deep learning approach for host-based cryptojacking malware detection", Evolving Systems, Springer, 2023, pp.1-16
[2] Shashank GuptaB. B. Gupta, "XSS-secure as a service for the platforms of online social network-based multimedia web applications in cloud", Springer 2022
[3] Patrick Duesse, Christian Gehl ,Ulrich Flegel Sven Dietrich ,Michael Meier, "Detecting zero-day attacks using context-aware anomaly detection at the application-layer", IEEE 2021.
[4] Panagiotis Papadopoulo, Panagiotis Ilia, and Evangelos Markatos, "Truth in Web Mining: Measuring the Probability and the Imposed Overheads of Cryptojacking, Springer 2019
[5] Debabrata Kara, Suvasini, Panigrahib, Srikanth Sundararajan, "SQLiGoT: Detecting SQL injection attacks using graph of tokens and SVM", Elsevier 2016
[6] Shashank Gupta B. B. Gupta, "JS-SAN: defense mechanism for HTML5-based web applications against javascript code injection vulnerabilities", Wiley Online Library 2016.
[7] Yen-Lin Chen ; Hahn-Ming Lee ; Albert B. Jeng ; Te-En Wei, "DroidCIA: A Novel Detection Method of Code Injection Attacks on HTML5-Based Mobile Apps", IEEE,2015.
[8] Panagiotis Papadopoulo, Panagiotis Ilia, and Evangelos Markatos, "Truth in Web Mining: Measuring the Probability and the Imposed Overheads of Cryptojacking, Springer 2019
[9] Shashank GuptaB. B. Gupta, "XSS-secure as a service for the platforms of online social network-

based multimedia web applications in cloud", Springer 2018

[10] Patrick Duesse,Christian Gehl ,Ulrich Flegel Sven Dietrich ,Michael Meier, "Detecting zero-day attacks using context-aware anomaly detection at the application-layer", IEEE 2017

[11] [Roberta PiscitelliEmail authorShivam BhasinFrancesco Regazzoni, Fault Attacks, Injection Techniques and Tools for Simulation, Springer 2017

[12] Debabrata Kara,Suvasini, Panigrahib, Srikanth Sundararajan, "SQLiGoT: Detecting SQL injection attacks using graph of tokens and SVM", Elsevier 2016

[13] Shashank Gupta B. B. Gupta, "JS-SAN: defense mechanism for HTML5-based web applications against javascript code injection vulnerabilities", Wiley Online Library 2016.

[14] Yen-Lin Chen ; Hahn-Ming Lee ; Albert B. Jeng ; Te-En Wei, "DroidCIA: A Novel Detection Method of Code Injection Attacks on HTML5-Based Mobile Apps", IEEE,2015.

[15] Danda B. Rawat ; Chandra Bajracharya, "Detection of False Data Injection Attacks in Smart Grid Communication Systems", IEEE 2015.

[16] Xing Jin, Xunchao Hu, Kailiang Ying, Wenliang Du, Heng Yin and Gautam Nagesh Peri, "Code Injection Attacks on HTML5-based Mobile Apps: Characterization, Detection and Mitigation",IEEE, 2014.