International Journal of Applied Science and Engineering

State of the art: The monero cryptocurrency mining malware detection using supervised machine learning algorithms

Wilfridus Bambang Triadi Handaya[1*], Mohd Najwadi Yusoff[2], Aman Jantan[2]

[1] *Department of Informatics, Universitas Atma Jaya Yogyakarta, Daerah Istimewa Yogyakarta, Indonesia*
[2] *School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia*

ABSTRACT

Today's evolving information technology trend is fuelling increasingly various cybercrime. Various cybercrime types, such as malware attacks and data breach activities, organizations, and countries, occur every day. Malware is one of the biggest cyber threats on the Internet today. Every year, the number of data breaches continues to increase.

Mining is a complete cryptocurrency network's computing process to verify transaction records, called blockchains, and receive digital coins in return. This mining process requires severe hardware and significant CPU resources to create cryptocurrencies. A statistic published by Statista in mid-2020 about the most detected crypto-mining malware types influencing corporate networks global from January to June 2020, it can be seen that cryptocurrency mining activity leading to the pool of Monero amounts to 46% derived from the use of XMRig.

Malware detection is like an endless war between malware authors and malware prevention vendors. This trend change also makes the procedures and forms of analysis must adapt. From previously done manually with various tools for static analysis, to be subsequently replaced with automatic analysis through the application of machine learning algorithms.

Keywords: Malware detection, XMRig, Monero, Cryptocurrencies, Mining.

OPEN ACCESS

Received: October 9, 2020
Revised: December 20, 2020
Accepted: January 6, 2021

Corresponding Author:
Wilfridus Bambang Triadi Handaya
wilfridus.bambang@uajy.ac.id

Publisher:
Chaoyang University of Technology
ISSN: 1727-2394 (Print)
ISSN: 1727-7841 (Online)

1. INTRODUCTION

Cybercrime is the uppermost threat to persons and organizations in the world. Cybercrime that occurs is dangerous to the sovereignty of information from the company's system and can also lead to targets specifically for the state's network. The trend of cyber warfare, becoming a new threat and challenge. That can be addressed with full vigilance and anticipation early to damage the system and cause future vulnerabilities. With the cumulative combination of software and hardware facilities into each characteristic of human lives, continuing data security is fetching more monotonous. However, it has made criminals the perpetrators of threats skilled in distributing and penetrating target defenses using incredibly crafted and never-before-seen malware. The collection of tools owned by cyber threat actor today has elevated some concerns for security companies. The researcher must adopt innovative ways of dealing by leveraging machine learning algorithms' general competencies by building an approach to enhance the fileless cryptocurrency malware detection and classification system (Handaya et al., 2020).

According to published research, more than one million malware attacks hit the Internet network every day in Q3 2017 (Chen et al., 2018). This notice has not yet broadly enclosed the number of malware attacks in the first half of 2019, with over

430,000 unique users attacked by financial threats. The number of economic aggression in the first half of 2019 was 10,493,792.

Cryptocurrency is a digital currency used for online virtual transactions. Complex secret passwords are used to protect and secure digital currency. In comparison to conventional, centralized currencies, digital currencies are decentralized. No person shall be present and serve as an agent during a transaction. Payment in the digital currency is made from sender to receiver or peer-to-peer. However, all such transactions are still documented and monitored in the cryptocurrency system of the network.

Some parties use cryptocurrencies with guaranteed anonymity to secure established blockchain systems (Handaya et al., 2020). Also, there is a blockchain system to protect these digital currency transactions. Blockchains are like ledgers that record any transaction process in a system that runs decentralized. This scheme's consequence is that digital currency transactions are more manageable, safer, and more practical than conventional banking systems.

Furthermore, the digital currency is decentralized. In other words, no party shall become an intermediary in the transaction. Payments made using digital currencies shall be made from peer to peer, i.e., from sender to recipient. However, all transactions are still recorded in the existing configuration of the cryptocurrency network. The recording is made by cryptocurrency miners and will be paid for the digital money used. Meanwhile, cryptocurrencies are decentralized, computers with specific and advanced requirements are required. Usually, use a blockchain network such that digital currencies can be used for transactions.

1.1 Research Problem

The Interpol report with the title "ASEAN Cyberthreat Assessment 2020 Key Insights From The ASEAN Cybercrime Operations Desk" includes ASEAN countries while cryptojacking appeared a new threat. With the growing use of cryptocurrencies and the potential to leverage user systems' computational capacity to conduct illegal cryptocurrency mining, corporations and individual users worldwide are at risk. Cybercriminals have been shown to exploit computer security vulnerabilities to conduct cryptojacking operations globally with substantial effective performance (Interpol, 2020).

Cryptocurrency has become the latest trend in the cyber world, and no time is wasted in exploiting its features to receive a fast income. The operations designate that cybercriminals target various sources, from personal devices, an organization device (Soviany et al., 2018), IoT devices, to industrial machine devices. They are continued, as any vulnerable device that can provide CPU cycles. These users get infected by a crypto mining malware program or visit websites that stealthily run crypto mining software in the background and have increased significantly. A new kind of threat has become mainstream.

This advanced endpoint protection delivers prevention levels than ever before, moved to the latest hacker-active mitigation capabilities, advanced application locking, and enhanced malware protection. This technology enables accurate and massive detection modeling to study the entire landscape of monitored security threats. With the capability to process large numbers of samples, machine learning can make more accurate and faster predictions with far fewer false-positive ones than traditional machine learning.

As previously known, the use of traditional machine learning models relies on cyber threat expert analysts to select attributes used to train models while adding subjective human elements. Machines are also becoming more complex and slower as more data is added to the model, reaching gigabyte sizes. This model may also have a higher false-positive rate and may reduce information technology productivity as it always distinguishes malware and benign software (Kasperksy, 2020).

The concept of using machine learning algorithms in recent years has become more realistic to implement in malware detection. With an increasing number of generated malware incidents every day, the need for more automated and intelligent methods to learn, adapt, and capture malware is critical. The central premise behind the use of machine learning is pattern matching. When introducing an instance of malware, it may not be immediately noticeable. However, when looking at and studying the many examples from a source database or a dataset, it is possible to get a relatively easy pattern to distinguish between benign software and malicious software. Using models with machine learning algorithms is that one of them is to find these patterns.

The use of machine learning from this study is designed to learn based on experience, creating a correlation between observed behavior and malware. This correlation results in a higher accuracy between existing and zero-day malware and has a lower false-positive rate.

2. CRYPTOCURRENCY MINING MALWARE

Cryptojacking malware has become one of the most dangerous malicious software forms distributed by cybercriminals in recent years (Interpol, 2020). The malware gave how some see it as relatively benign compared to other more destructive attacks such as ransomware and trojans, and sometimes ignored as threats. Cryptocurrency mining gives cybercriminals an exploited foothold to deliver destructive attacks in the future. On the other hand, placing anticipatory actions against cryptojacking attacks at a lower level places the organization at high risk.

The most crucial feature of malware is its ability to persist on the infected system for as long as possible. In this case, the malware downloaded is a banking trojan that uses the cryptocurrency to steal access data and carry out fraudulent transactions like a typical banker. The cryptocurrency

mining malware steals the infected machine's resources, which significantly impairs its performance and increases its wear and tear. The threat infects devices and machines and turns them into a monero-mining botnet. For example, WannaMine cryptojacking uses the same techniques as the WannaCry Ransomware to unfold destructive effects, such as CPUMiner and EternalM Miner.

Cryptojacking is an unlawful appearance of crypto mining. Cybercriminals use similar malware techniques to snitch into endpoints: drive-by downloads from the outside network, phishing campaigns, vulnerabilities in the browsers used, and specific plugins. There are two main working approaches for cryptojacking: the first is to infect the browser with a plugin-based that devours some computing power from the CPU. At the same time, attack an unaware user's online activities. The second form is similar to standard malware: installed directly to the machine. It runs on local machines and uses the target Internet connection in mining cryptocurrency activities

In 2018, based on the Cryptocurrency Mining Malware Trends and Threat Predictions from Sucuri, there is a shift toward cryptojacking attacks, especially with the total cryptocurrency market capitalization rose 3224% during 2017 from $17 billion to $565 billion, with daily trade volumes surpassing USD 50 billion. In Fig. 1, we can see the report from The European Union Agency for Cybersecurity, titled ENISA Threat Landscape 2020 – Cryptojacking. Based on Enisa (2020), the top cryptomining malware globally statistics describes and analyzes the domain and lists relevant recent incidents. Based on the graph, XMRig, a high-performance Monero (XMR) miner, counted 21% of all malware cryptomining activities from January 2019 to April 2020.

Due to the absence of cybersecurity defense and vulnerabilities, it was noted that thousands of devices were compromised, such as MikroTik routers with the known vulnerability CVE-2018-14847; cybercriminals had launched their cryptojacking campaigns in the ASEAN area. The value of cryptocurrencies like Bitcoin soared over several months as demand surpassed supply, peaking at USD 12,000 at the end of June 2018, tripling its value compared to the beginning of the year (Interpol, 2020). The prevalence of crypto mining in the area occurred with this rise and choose the monero (XMR), with activity detections in some targeted attacked countries like Indonesia, Philippines, Thailand, and Vietnam. Meanwhile, it is a reason for attackers to make money compared to other cyberthreat activities. It takes more attention from all parties to minimize the organization's risk, falling prey to cryptojacking and other types of malware attacks.

2.1 How Cryptocurrency Mining Works

Cryptocurrency systems generally claim to provide anonymous and decentralized transaction processing. This anonymity may be an additional precaution for user confidentiality and privacy, as written from research conducted by Zhang et al. (2019). Acceptance and demand for cryptocurrencies have increased a hundredfold over the past few years. The industry has evolved since the beginning and is associated with the growing trading and acceptance of cryptocurrencies. Currently, cryptocurrencies are already available on hundreds of exchanges around the world against fiat currencies. The continuousness and constancy of malware attacks, especially in cryptocurrency mining, make this research's main background—the technique changes, which use predictable procedures to enter and control the target system. However, using various combinations (synergistic) of attacks provides the highest success factor to take over the network and perform multiple unlawful events. There has been a compromise in cybersecurity and law enforcement groups over the period that crypto piracy is weakening. The consequence of numerous enormous removals of mining malware or the

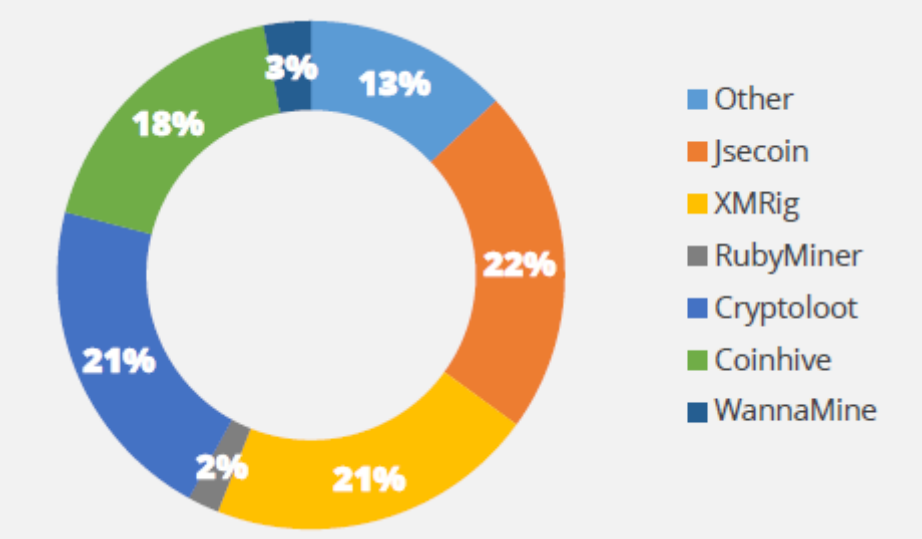

Fig. 1. Top cryptomining malware globally

```
$cmdmon="powershell -NoP -NonI -W Hidden `"`$mon = ([WmiClass]
'root\default:Win32_Services').Properties['mon'].Value;`$funs = ([WmiClass]
'root\default:Win32_Services').Properties['funs'].Value ;iex
([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String(`$funs)));Invoke-Command  -ScriptBlock
`$RemoteScriptBlock -ArgumentList @(`$mon, `$mon, 'Void', 0, '', '')`""

$vbs = New-Object -ComObject WScript.Shell

$vbs.run($cmdmon,0)
```

Fig. 2. Command to run scripts that call miner executable

WannaMine campaign that plowed half a million personal computer into mining cryptocurrencies.

An additional factor is the closure of coinhive, a leading site that handles crypto miners. Coinhive provides a JavaScript script that sites can combine to make user resources mine cryptocurrency, Monero. This cryptocurrency appeals to cybercriminals because it is hard to track. Azizah et al. (2020) convey that the coinhive script is speedily misused: mining scripts inserted into sites without the website owner's permission with more processing power is used in mine cryptocurrencies. The coinhive site closed in March 2019, and with it, the number of site infections dropped sharply. The other factor shows cryptocurrency valuations were collapsing - saw its destruction as a cryptojacking-related game changer. In Fig. 2, we can understand the function to command to run scripts that call miner executable using PowerShell.

Microsoft PowerShell is part of the Microsoft Windows operating system, which is active and ready for use when the operating system is installed. Applications are built using the .NET framework and recognize instructions in the Common Language Routine (CLR). The instructions known in the PowerShell scripting will direct the operating system to run a specific activity. On the other hand, it offers full-access to critical Windows system functions such as the Windows Management Instrumentation (WMI) and the Component Object Model (COM) objects, as mentioned before by Hendler et al. (2018). Microsoft PowerShell includes some attributes that also contain commands in built-in type, with every command using a consistent format, to have the ability to execute or perform commands given by users. PowerShell also consists of an extensive collection of built-in controls, with each having a consistent interface, and these can work with user-written commands. Cybercriminals can rely on the fact that it is already there without installing anything else, and this malware aims to be more active regarding infecting machines and avoiding detection.

The default software from the Microsoft Windows operating system, such as Powershell and Windows Management Instrumentation (WMI), is one of the favorite applications used by malicious software in carrying out its activities to extract resources from the user's computer (Bulazel and Yener, 2017). This technique is found in various malware analyses carried out and are a significant concern in preventive measures. The new technique that malware makers later developed is to use attacks without leaving a trace of files through the script access mechanism or using vulnerabilities found in operating systems and applications.

The infection chain in fileless cryptocurrency-mining malware involves loading the malicious code to the system's memory. The physical footprint is only stored as an indicating an infection is a malicious batch file and installed WMI service, and with some various using PowerShell features. The high value of crypto-currencies has attracted malicious actors that use hijacked resources to mining cryptocurrency (Pastrana, 2019).

2.2 Why Monero

In August 2017, researchers exposed a type of malware merely designed to mine cryptocurrency, called Monero (XMR). Moreover, as of this writing, XMR is the cryptocurrency affording its users the highest anonymity quantity. Fig. 3 shows the statistic published by Statista in mid-2020 about the most commonly detected crypto-mining malware families affecting corporate networks worldwide. The time range started from January to June 2020, and it can be seen that cryptocurrency mining activity leading to the pool of monero amounts to 46% derived from the use of XMRig. XMRig is an application that is malicious software categorized as a trojan that specifically conducts monero cryptocurrency mining activities using resources from the victim's computer without the system owner's permission.

Monero, created in April 2014, is a variant of cryptocurrency with a focus on private and censorship-resistant transactions. Monero is an open-source project that develops a network that allows parties to communicate without disclosing the number of the sender, receiver, or transaction using robust encryption technology. Monero has a decentralized ledger, like other cryptocurrencies, that all participants can download and check independently. Monero uses an obscure public directory, which means that anyone can broadcast transaction history, while others cannot extract transaction details. Monero implements a proof-of-work mechanism to issue new coins and encourage miners to protect the network and verify transactions.

Monero is based on the CryptoNight hash proof-of-work algorithm, which comes from the CryptoNote protocol (Möser et al., 2018). The CryptoNote protocol has significant algorithmic differences in blockchain confusion. The ring signature, which mixes the signer input with each of the other ring member's input, connects each subsequent transaction more complicated.

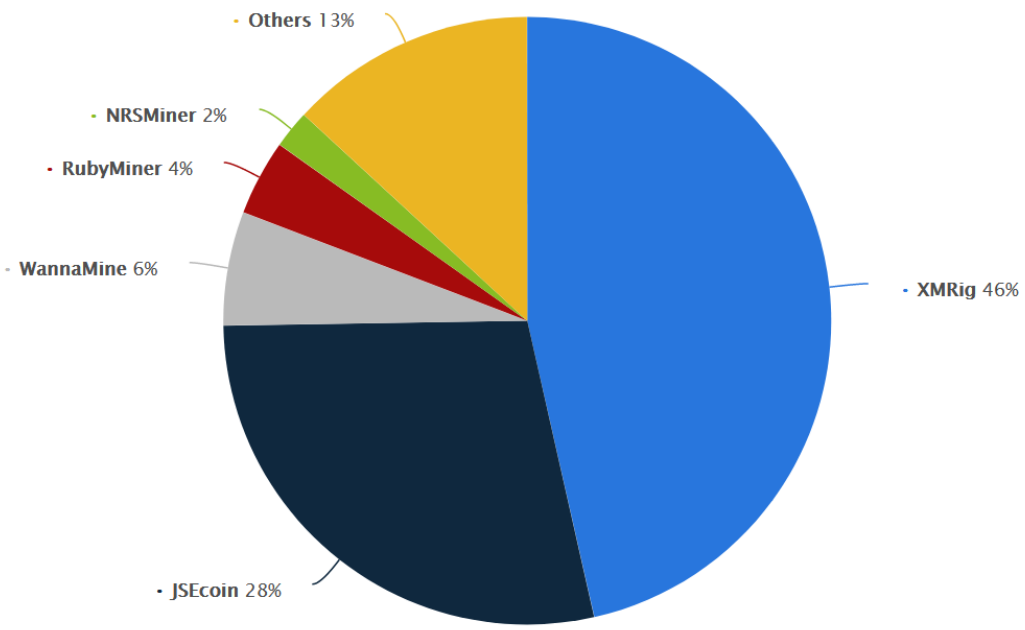

Fig. 3. Most detected crypto-mining malware families affecting corporate networks worldwide from January to June 2020

Moreover, the stealth address generated for each transaction prevents anyone other than the sender and receiver from discovering the actual destination transaction address. Ultimately, this is the amount of transfer hidden by the confidential ring transaction mechanism. Monero is designed to resist integrated circuit mining (ASIC), commonly used to mine other cryptocurrencies, such as Bitcoin. It can mine efficiently on consumer hardware such as x86, x86-64, ARM, and GPU.

The newly discovered form of cryptocurrency mining malware was able to remain well hidden, so the researchers who investigated it found that it had spread to almost every computer in the company that had been infected. Cryptojacking malware exploits infected computers' processing power to mine cryptocurrencies, causing the system to slow down, even to the fore. The Monero crypto-mining campaign came to light after several security platforms spotted suspicious network alerts and abnormal file activity on systems within the organization that had reported unstable applications and network slowdowns. Precisely, for crypto miners, organizations must monitor CPU activity on the computer. With mining doing its job by exploiting processing power, organizations should pay attention to any noticeable decrease in processing speed.

Malware has been made to be highly persistent and stay in regular contact with command and control servers, which, if necessary, can provide new instructions or stop malware. However, researchers note that during analysis, no new commands are received. The researchers found several variants of crypto mining malware installed on almost every server and workstation. It was victimized, and that some machines had even infected password thieves - likely used as a means to add more machines to the mining botnet. It is not known how the initial infection occurred, but in some cases, malware has been around for years.

In incidents such as the WannaCry Ransomware Attack or the purchase of illegal substances in the deep network for illegal use, privacy-by-design from monero cryptocurrency (Saxena et al., 2017) has attracted relevant parties to evade law enforcement. Also, since payments and balance sheets are still under a veil, all those seeking financial privacy are encouraged to use Monero, which is not the standard for most cryptocurrency variants.

Kumar and Vaishakh (2016), suggested to turn out the detecting obfuscation before doing the malware reverse engineering is critical. They show that obfuscation is a type of malware-Java malware-which can be detected by extracting important static features. Alazab et al. (2014) proposed the current hybrid MR-ANNIGMA is superior to the independent wrapper and filter method and can produce 97.53% accuracy. This research's main contribution is developing a fully automatic method that does not require signatures to decompress, deobfuscate, and reverse engineer binary executable files without the need to check the assembly code manually. Carlin et al. (2018) proposed that dynamic opcode tracking helps detect the encryption mining behavior in the browser's sample set. We can also differentiate between HTML files executed in the browser with encrypted mining enabled and the duplicate files with encrypted mining disabled; speed and accuracy are too high.

Part of the problem is that the threat posed by cryptojacking is usually a function of the value of the cryptocurrency itself. In short, as the value of cryptocurrencies increases, when market conditions are

worthwhile, hackers will respond by preparing their activities accordingly. The valuation of cryptocurrency itself is still unstable in the global industry continues to be carried out systematically and on a large scale. However, given the need to set up hundreds, thousands, or even thousands of computers to be linked together to generate the necessary calculations to obtain enough other cryptocurrencies such as monero, it must be easy to prepare for fraudulent cryptojacking hackers use.

However, this contradicts company data showing that cryptocurrency mining malware is the most detectable threat in the first semester of 2019 in terms of file-based threat components. Unlike many other malware forms, individual victims may not notice it in a password hijacking case because the malware is likely to dig secretly in the background. However, in the corporate environment, some computers eventually penetrated, and the story may be different. Therefore, an expert continuously needs to look for signs in their networks, such as abnormal power consumption peaks and system performance degradation. Although attacks in browsers are sometimes challenging to detect, many preventive measures can be taken after the threat has been isolated and resolved. Of course, it is always better if there is action to prevent this possibility.

On the other hand, the first exact route for communication through the call port is mine-defense browser extensions such as MinerBlock, NoScript, or No Coin. The browser extensions depend on the browser used because some browsers are more comprehensive than others because they may have blocked known mining domains. Similarly, anti-virus, anti-malware, and ad blocking programs also need to be updated and customized. On the other side, cross-border investigations help took down and limited the cyber perpetrators moving forward.

Although attacks in browsers are sometimes challenging to detect, several preventive measures can be taken after the threat has been isolated and resolved. Of course, it is always better if there are measures to prevent this possibility. The miner is not often malicious. It does borrow the user's system resources without the user's permission to mine Monero. Based on this analysis, researchers speculate that the Monero wallet address and mining pool are expected to be hard-coded into PE-based coin miners. The infected traffic may flow to the mining pool address.

3. MACHINE LEARNING IN CRYPTOCURRENCY MINING MALWARE DETECTION

Malware analysis is a method in which the goals are to analyze and determine the indicated malicious software sample file. Specific information generated will provide knowledge to choose or develop effective detection techniques for applications considered to have malicious code in them. The malware analysis process is also a crucial aspect of research and development methods for efficiently removing infected systems or applications. An expert in the anti-virus software requires significant resources if manually investigating files infected by malware.

The quantity of malware that experts need to examine remains to grow every day. The trend will increase, supported by better knowledge of malware programming and the extensive use of computer equipment today. Malware detection is like an endless war between malware makers and malware prevention vendors. This trend change also makes the procedures and forms of analysis have to adapt. From previously done manually with various tools for static analysis, to be subsequently replaced with automatic analysis through sandboxes, open-source projects, application of machine learning algorithms, or other related solutions.

The analysis and extraction approaches from malware can frequently describe into two types: (i) created on features strained from an unpacked static version of the executable file without execution of the analyzed executable files (Gandotra et al., 2014). Furthermore, (ii) formed on dynamic features or behavior features discovered during the executable files' execution. Previous research by Le et al. (2018) points out that the data set can be trained to analyze malware attacks' micro behavior. The machine learning algorithm can detect a new malware machine learning algorithm that aims to develop a framework to analyze the scripts at the highest stage of a network security solution to achieve zero days of the attack. In Fig. 4, we can take a look at the detection algorithm life cycle released by the Kaspersky report in 2020, in which anti-malware companies have turned to the machine science field of machine learning. It has been effectively used for image recognition, search, and decision-making to extend its malware detection and classification based on the Kaspersky Report (Kasperksy, 2020).

Zhang et al. (2017) proposed a general framework based on independent component analysis and a semi-supervised algorithm named ColSVM (collaborative support vector machine). Taran et al. (2018) proved a new defense mechanism based on cryptographic principles, which can be applied to many existing deep neural network classifiers to defend against gradient-based adversarial attacks. Simple and effective mechanisms based on practical cryptography principles prove the proposed ideas' great potential. Souri and Hosseini (2018) visualized a malware detection taxonomy based on machine learning approaches and show in Fig. 5.

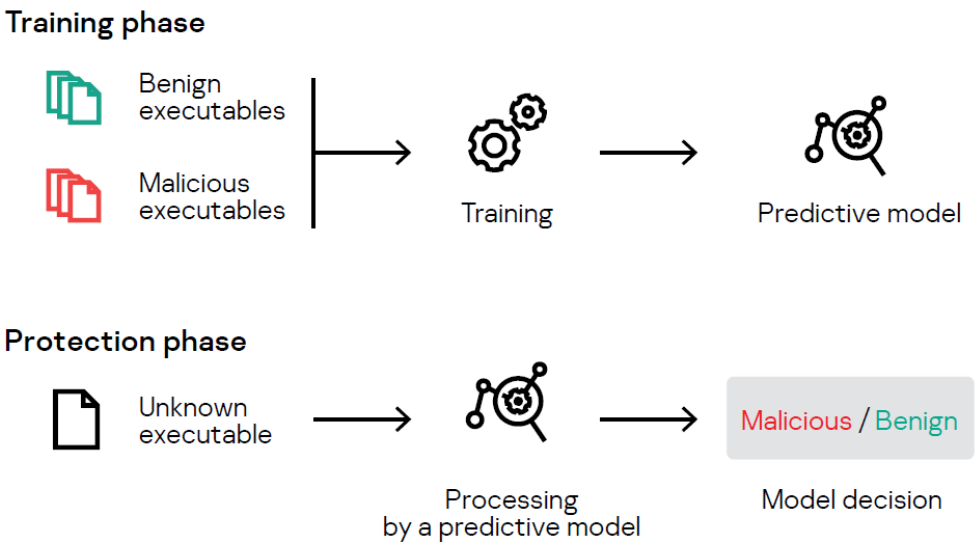

Fig. 4. Machine learning detection algorithm life cycle

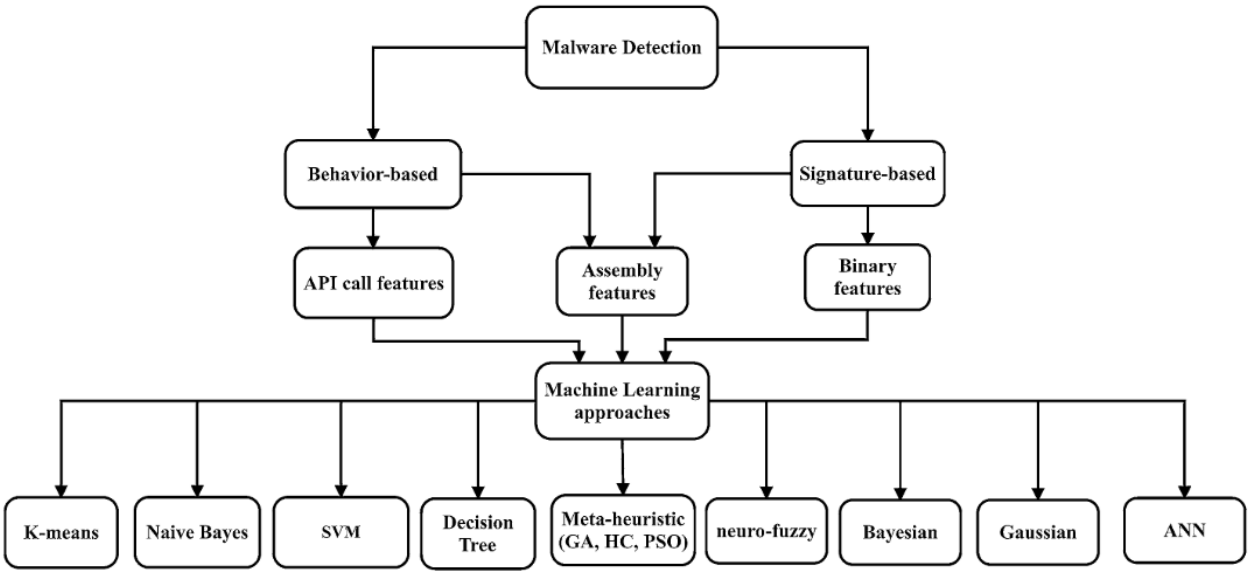

Fig. 5. Taxonomy of malware detection techniques

3.1 Static Analysis

Static analysis, also called static code analysis, is the procedure of debugging the application without executing the full function inside; at this stage, malware testing is done without inspection of the code or running an indicated executable file. The information produced concerning functions and additional technical indicators while support the generate digital signatures from malicious code. When the researcher is doing a static analysis step, several tools and practices are used to gather as much malware information as possible. The first step is to scan the file using specialized software from an offline vendor or insert it into an online scanner website such as VirusTotal. Alam et al. (2015) use static analysis to detect malicious activity in executables to detect obfuscation trends in malware that demonstrate a better accuracy rate.

Dynamic analysis is an effective method to detect zero-day attack threats. This analysis allows the malware to check behavior, learn features, and try to identify technical indicators. After obtaining all the detailed information, use it to detect the parameters. Technical indicators can include IP addresses, domain names, file path locations, other files, and registry entries, which can be found on the network or a computer while visualized in Fig. 6. The next step is to identify and find communications with external servers controlled by the attacker. The working principle of the dynamic analysis process is to run the malware correctly in a virtual machine that has been created or prepared.

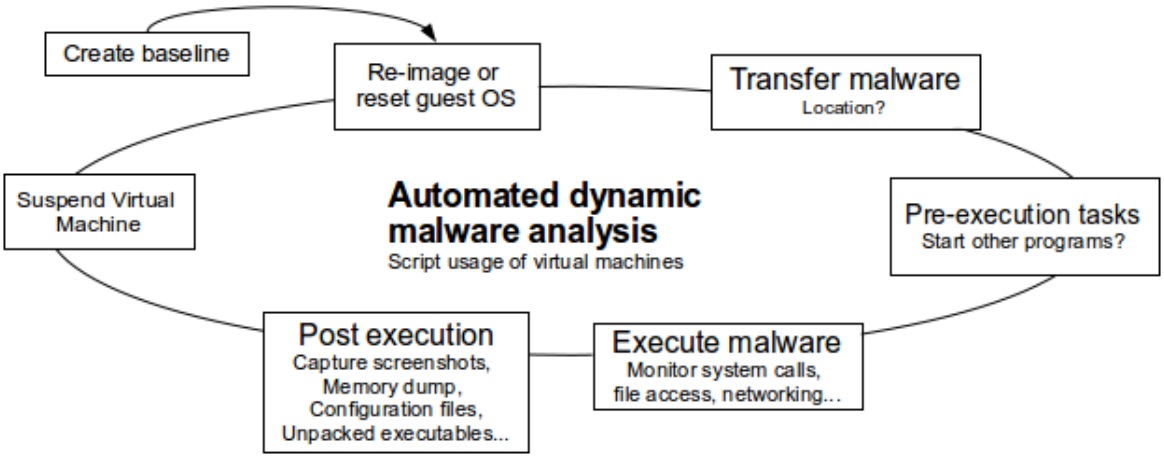

Fig. 6. Sample of the automated dynamic malware analysis phase

3.2 Dynamic Analysis

Dynamic analysis as an answer to static analysis countermeasures and analysis gained popularity is an analytical technique performed to detect malicious software, based on the previous research by St'Astna and Tomasek (2018). Shiva, in another research, notes that some malware detection techniques such as API Calls, Instruction tracing, System Calls (Darshan and Jaidhar, 2018), Registry changes, and unknown access to memory write. Canzanese and Kam (2015) running a study using system call traces using a custom host-agent known as the System Call Service (SCS), obtained from production hosts. SCS is a service application in recent versions of Microsoft Windows and logs process-level system call traces. Stiborek et al. (2018) show malware detection technology using dynamic analysis. System call information is used to build fine-grained models to capture the behavior of malware. A scanner is used to match new programs' activity with these models to classify them as benign or malicious software.

Damodaran et al. (2015) stated the malware detection technology with the trained Hidden Markov Model (HMM) on static and dynamic feature sets. The detection rate of a large number of malware families is compared. The results show that the detection rate of dynamic analysis is the highest. Choudhary and Vidyarthi (2015) proved that the dynamic analysis indicates that the executable file's execution can understand its behavior. The obtained results show high accuracy, representing that this method can be further improved using a larger sample space. Other research projects have proposed a malware detection tool based on runtime monitoring, which can extract the statistical structure of malware from the headers of all essential parts of the PE file, decrease the dimensionality and increase the compactness of the function. Bai et al. (2014) stated by mining the format information of PE files, a malware detection method is proposed, and experiments on recent Win32 malware are introduced.

Dai et al. (2018) stated that dynamic analysis has efficiently overcome the shortcomings of static analysis technology, but it is vulnerable to malware evasion. This analysis enables malware to check behavior, learn features, and try to identify technical indicators. After obtaining all the detailed information, use it to detect the parameters. Ceschin et al. (2019) stated that the technical indicators could include IP address, domain name, file path location, other files, and registry entries found on the network or computer. Besides, the process continues to identify and find communications with external servers controlled by the attacker. The working principle of the dynamic analysis process is to run the malware correctly in a specific virtual machine that has been created or prepared.

3.3 Malware Dataset

Azab et al. (2014) proposed that applying the machine learning algorithms is necessary to access a dataset containing many samples to be analyzed. The data set is mainly composed of malware and benign features. Data collection aims to obtain fundamental data sets in the data set, representing the most common behavior patterns and use similarity parameters of data objects. The research started using samples of the malware dataset, including UserId, UUID, Details, Actions, ActionType, SessionType, Version, and SessionID.

Researchers can interact directly with datasets from previous research established and grouped previously by third parties such as vendors, universities, research institutes, or individuals. Mohaisen et al. (2015) even created a new dataset named as automal as the study's result. For examples of datasets used in various researches, it can be seen in Table 1.

This study will also use the sample classification of malware from the EMBER dataset, based on Vinayakumar et al. (2019) research. The author will be compensated by the cryptocurrency mining malware dataset collected by other researchers as a reference for the primary dataset. An extraction will then be performed to create a new data set containing malware features in the cryptocurrency mining category.

Table 1. An example of a dataset used in malware detection

No	Dataset	Vendor/Author	Some researcher used
1	NSL-KDD	https://www.unb.ca/cic/datasets/nsl.html	(Sabar et al., 2018)
2	VXHeavens	https://archive.org/details/vxheaven-windows-virus-collection	(Bai et al., 2014) (Alazab et al., 2014) (Gupta and Rani, 2018)
3	Ember	https://github.com/endgameinc/ember	(Vinayakumar et al., 2019; Alazab, 2014) (Zhang et al., 2019) (Ceschin et al., 2019) (Vinayakumar and Soman, 2018) (Raff et al., 2016)
4	Virustotal	https://virustotal.com	(Canzanese and Kam, 2015; Rhode et al., 2018) (Gupta and Rani, 2018) (Xiaofeng et al., 2018) (Ceschin et al., 2019) (Pastrana, 2019)
5	KDDCUP'99	http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html	(Li and Jiao, 2015)
6	Honeynet project	https://www.honeynetproject.com/dataset.html	(Alazab et al., 2014)
7	Virusshare	https://virusshare.com/	(Shijo and Salim, 2015) (Carlin et al., 2018) (Gupta and Rani, 2018) (Xiaofeng et al., 2018) (Pastrana, 2019)
8	nothink	http://www.nothink.org/	(Gupta and Rani, 2018)
9	automal		(Mohaisen et al., 2015)
10	Zeus dataset	Trendmicro	(Azab et al., 2014)
11	Microsoft Malware Classification Challenge (BIG 2015)	https://www.kaggle.com/c/malware-classification	(Le et al., 2018)
12	Avg	Avg company	(Pluskal, 2015)
13	AMP ThreatGrid	Cisco	(Stiborek et al., 2018)
14	MNIST, Fashion MNIST		(Taran et al., 2018)

3.4 Supervised Learning

The learning in supervised learning is based on using the initial data set's labeled data, where the data samples are mapped to the correct results. On this data set, the model is trained to know the location of the correct result. Li et al. (2015) propose a novel hybrid methodology to detect malicious code based on deep learning, which combines the advantages of AutoEncoder and DBN to improve detection accuracy while reducing the model's time complexity. Rhode et al. (2018) stated that a new malware prediction model based on Recurrent Neural Network (RNN) was developed, significantly reducing the dynamic detection time to less than 5 seconds while retaining the dynamic's advantages model. This mechanism uses machine activity data to predict malicious behavior and proves that its function is superior to other machine learning solutions previously used for malware detection.

In the previous research, Xiaofeng et al. (2018) proposed a combination classification architecture that combined machine learning and deep learning and described a new API sequence process algorithm AMHARA. This research shows that integrated classifier has a better performance than separate machine learning or deep learning. Gupta and Rani (2018) proposed scalable architecture to detect malware binaries on MLIB Apache Spark and HDFS using three supervised algorithms (NB, SVM, and RF) and works on the Apache Spark's scalable machine learning library. It turns out that the random forest algorithm provides the best accuracy for malware detection.

Also, classification accuracy stated by Banin and Dyrkolbotn (2018) shows that the memory access mode can

be effectively used for malware detection. Without loss of classification performance, feature selection's implementation advantageously reduces the data dimension by three orders of magnitude.

3.4.1 Supervised Learning Algorithm used in Research

The various research publications and experimental data activities that have been done previously in implementing machine learning algorithms in introducing malware are also used in the current research, shown in Table 2.

The machine learning method presently suits the most practical approach, start from the design and implementation of various cybersecurity solutions for an endlessly increasing application area, especially in malware detection. To calculate the efficiency of a malware detection algorithm, researchers can use Large O notation, commonly

Table 2. The conclusions of the various machine learning algorithms used in malware detection

No	Researcher	Algorithms	Accurate	Remarks
1	Anderson (2018)	LightGBM, malconv	92.2% at a fpr less than 0.1%, or a 97.3% at a less than 1% fpr	ember
2	Azab et al. (2014)	k-NN, combined with TLSH SSDEEP, SDHASH, and NILSIMSA methods	0.989 and 0.999	Zeus datasets
3	Bai et al. (2014)	Random forest, the ensemble with Adaboost (J48) and Bagging (J48)	99.1%	vxheaven
4	Banin and Dyrkolbotn (2018)	kNN	78.4%	Not listed
5	Canzanese and Kam (2015)	logistic regression (LR) and support vector machines (SVMs) using stochastic gradient descent (SGD)	92%	virustotal
6	Carlin et al. (2018)	SVM	99.9%	VirusShare
7	Ceschin et al. (2019)	Malconv, Non-Neg.MalConv LightGBM	Malware 91.69% benign 80.95% benign 93.28%	Ember, virustotal
8	Choudhary and Vidyarthi (2015)	Support Vector Machine	97.8%	Not listed
9	Dai et al. (2018)	SPAM-GIST, HPC, k-NN(k = 3), k-NN(k = 5), RF MLP	multilayer perceptron (MLP) is the best with max precision up to 95%.	Not listed
10	Gupta and Rani (2018)	Random Forest	98.88%	Not listed
11	Zhang et al. (2017)	Collaborative SVM, Combined with Independent Component Analysis	> 90%	Extracted manually from 2045 executables with CRC64 unified coding
12	Kruczkowski (2014)	SVM, Naive Bayes, kNN	C.A. more than 80%	Not listed
13	Kumar and Vaishakh (2016)	Random forest	99%	dataset of 375 malware samples containing 182927 strings and 12721 Java classes

No	Researcher	Algorithms	Accurate	Remarks
14	Lysenko et al. (2019)	SVM	90.28% to 98.21%	Not listed
15	Mohaisen et al. (2015)	SVM	For SVM Poly. Kernal, 99.22%	Build a new dataset, named AutoMal.
16	Pluskal (2015)	SVM	FPR on a subset of practices lower than 0.05%.	AVG Dataset
17	Zhang et al. (2019)	Linear SVC, Logistic Regression, LightGBM, Random Forest	micro avg f1-score 0.96 and macro avg f1-score 0.89	ember
18	Sabar et al. (2018)	Hyper Heuristic SVM	85.69%	NSL-KDD
19	Sari and Maarof (2017)	Decision Tree	93.3% for multiclass and 94.6% for binary classification	Not listed
20	Shijo and Salim (2015)	Random Forest and SVM, combined with extended features by concatenating the static and dynamic feature vector	97.68 % and 98.71%	VirusShare
21	Vinayakumar et al. (2019)	The baseline method with gradient boosted decision tree (GBDT) model using LightGBM	99.911%	ember
22	Wang et al. (2018)	Support Vector Machine (SVM), BayesianNetwork (BN), Logistic Regression (LR), and Multilayer Perceptron (MLP)	96.52%, 93.48%, 88.99%, and 86.16% in validation set	Not listed
23	Xiaofeng et al. (2018)	Random Forest	98.3%	Virus Share, VirusTotal

referred to as Big-O Notation, which is a way to analyze a programming algorithm against execution time and how efficient and complex the code lines are in the time dimension. The notation conveyed the upper limit of a function's growth rate and commonly described asymptotic output (Dalatu, 2016). By understanding Big-O Notation, it will be easier to solve the problems at hand and give insight into a function's expected performance, especially in finding the better machine learning algorithms to detect cryptocurrency mining malware. There are two dimensions in calculating the complexity of the code. The first is the complexity of space complexity related to how much space is used, such as memory or computer hard disks. The second is the complexity of time or time complexity relating to how long lines of code are executed.

From the proposed various studies before, we specifically look at three supervised learning algorithms' performance, namely SVM, Random Forest, and J48 Decision Tree, in providing high accuracy to malware detection from the previous research in the summarized table. Furthermore, the classification knowledge will be used to find the best accuracy in detecting cryptocurrency mining malware.

4. CONCLUSION

This technical note is intended to provide implementation instructions for related work in information technology to support malware issues. Malicious software can interfere with a computer's operation, collect sensitive information, or access computer systems. This form of malware can appear in executable code (EXE), scripts, active content, and other software. Malware is a general term used to refer to various forms of malicious or intrusive software. Malware is usually disguised as regular files or embedded in harmless files. The rapid development of malware requires vigilant computer system users, especially those using Windows and MAC platforms.

Crypto-jacking is a malicious activity that makes infected devices secretly mine digital currencies. To do so, attackers must use the victim's processing power and network quota (usually unknown and without permission). In general, mining malware designed for malicious activity is designed to use minimal resources to avoid victims' attention. Knowing that mining requires a lot of computing power, attackers can try and attack multiple devices. In this way, they can get enough computing resources to conduct mining activities at low risk and cost.

To accelerate the malware detection, the researcher with background knowledge of machine learning is essential for understanding the research's actual implementation. The concepts of feature sets, feature extraction, and selection methods are proven together with the machine learning algorithms used in the actual part. The selected algorithms are support vector machine, decision tree, and random forest.

ACKNOWLEDGMENT

This research is supported by the Ministry of Higher Education Malaysia under the Fundamental Research Grant Scheme (FRGS) No. 203/PKOMP/6711799 and USM Short-term Grant No. 304/PKOMP/6315237. The author also acknowledges the Universitas Atma Jaya Yogyakarta and Yayasan Slamet Rijadi Yogyakarta for supporting the author's study at Universiti Sains Malaysia.

REFERENCES

Alam, S., Horspool, R.N., Traore, I., Sogukpinar, I. 2015. A framework for metamorphic malware analysis and real-time detection. Computers and Security, 48, 212–233. doi:10.1016/j.cose.2014.10.011

Alazab, M., Huda, S., Abawajy, J., Islam, R., Yearwood, J., Venkatraman, S., Broadhurst, R. 2014. A hybrid wrapper-filter approach for malware detection. Journal of networks, 9, 2878–2891. doi:10.4304/jnw.9.11.2878-2891.

Anderson, H.S.R., Phil. 2018. EMBER: An open dataset for training static PE malware machine learning models. arXiv e-prints. Retrieved from https://arxiv.org/pdf/1804.04637.pdf

Azab, A., Layton, R., Alazab, M., Oliver, J. 2014. Mining malware to detect variants. Fifth Cybercrime and Trustworthy Computing Conference, 44–53. doi:10.1109/CTC.2014.11

Aziz, B.A.A., Ngah, S.B., Dun, Y.T., Bee, T.F. 2020. Coinhive's monero drive-by crypto-jacking. IOP Conference Series: Materials Science and Engineering, 769. doi:10.1088/1757-899x/769/1/012065

Bai, J., Wang, J., Zou, G. 2014. A malware detection scheme based on mining format information. The Scientific World Journal. doi:10.1155/2014/260905

Banin, S., Dyrkolbotn, G.O. 2018. Multinomial malware classification via low-level features. Digital Investigation, 26, S107–S117. doi:10.1016/j.diin.2018.04.019

Bulazel, A., Yener, B. 2017. A survey on automated dynamic malware analysis evasion and Counter-evasion. Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium on - ROOTS, 1–21. doi:10.1145/3150376.3150378

Canzanese, R.M., Spiros, Kam, M. 2015. System Call-based detection of malicious processes. IEEE International Conference on Software Quality, Reliability and Security, Vancouver, BC, Canada.

Carlin, D., O'Kane, P., Sezer, S., Burgess, J. 2018. Detecting cryptomining using dynamic analysis. 16th Annual Conference on Privacy, Security and Trust (PST), Belfast, Ireland

Ceschin, F., Botacin, M., Gomes, H.M., Oliveira, L.S., Grégio, A. 2019. Shallow security: on the creation of adversarial variants to evade machine Learning-based malware detectors. Reversing and Offensive-oriented Trends Symposium (ROOTS), Vienna.

Chen, L., Sultana, S., Sahita, R. 2018. HeNet: A deep learning approach on intel processor trace for effective exploit detection. IEEE Security and Privacy Workshops (SPW), 109–115. doi:10.1109/SPW. 2018.00025

Choudhary, S.P., Vidyarthi, M.D. 2015. A simple method for detection of metamorphic malware using dynamic analysis and text mining. Procedia Computer Science, 54, 265–270. doi:10.1016/j.procs.2015.06.031

Dai, Y., Li, H., Qian, Y., Lu, X. 2018. A malware classification method based on memory dump grayscale image. Digital Investigation, 27, 30–37. doi:10.1016/j.diin.2018.09.006

Dalatu, P.I. 2016. Time complexity of K-means and K-medians clustering algorithms in outliers detection. Global Journal of Pure and Applied Mathematics, 12, 4405–4418.

Damodaran, A., Troia, F.D., Visaggio, C.A., Austin, T.H., Stamp, M. 2015. A comparison of static, dynamic, and hybrid analysis for malware detection. Journal of Computer Virology and Hacking Techniques, 13, 1–12. doi:10.1007/s11416-015-0261-z

Darshan, S.L.S., Jaidhar, C.D. 2018. Performance evaluation of Filter-based feature selection techniques in classifying portable executable files. Procedia Computer Science, 125, 346–356. doi:10.1016/j.procs.2017.12.046

ENISA, 2020. ENISA Threat landscape 2020 - cryptojacking. Retrieved from https://www.enisa.europa.eu/publications/enisa-threat-landscape-2020-cryptojacking/at_download/fullReport

Gandotra, E., Bansal, D., Sofat, S. 2014. Malware analysis and classification: A survey. Journal of Information Security, 5, 56–64. doi:10.4236/jis.2014.52006

Gupta, D., Rani, R. 2018. Big data framework for Zero-day malware detection. Cybernetics and Systems, 49, 103–121. doi:10.1080/01969722.2018.1429835

Handaya, W.B.T., Yusoff, M.N., Jantan, A. 2020. Machine learning approach for detection of fileless cryptocurrency mining malware. Journal of Physics: Conference Series, 1450. doi:10.1088/1742-6596/1450/1/012075

Hendler, D., Kels, S., Rubin, A. 2018. Detecting malicious PowerShell commands using deep neural networks. 13th ACM ASIA Conference on Computer and Communications Security, Incheon, Republic of Korea.

Interpol, 2020. Asean cyberthreat assessment 2020 key insights from the asean cybercrime operations desk. Retrieved from https://www.interpol.int/content/download/14922/file/ASEAN_CyberThreatAssessment_2020.pdf

Kasperksy, 2020. Machine learning methods for malware detection. Retrieved from https://media.kaspersky.com/en/enterprise-security/Kaspersky-Lab-Whitepaper-Machine-Learning.pdf

Kruczkowski, M.N.-S., Ewa. 2014. Comparative study of supervised learning methods for malware analysis. Journal of Telecommunications and Information Technology, 4, 24–33. doi:10.1109/CNSR.2007.22

Kumar, A., Fischer, C., Tople, S., Saxena, P. 2017. A traceability analysis of monero's blockchain. In: Foley S., Gollmann D., Snekkenes E. (eds) Computer Security – ESORICS 2017. ESORICS 2017. Lecture Notes in Computer Science, vol 10493. Springer, Cham. https://doi.org/10.1007/978-3-319-66399-9_9

Kumar, R., Vaishakh, A.R.E. 2016. Detection of obfuscation in java malware. Procedia Computer Science, 78, 521–529. doi:10.1016/j.procs.2016.02.097

Le, Q., Boydell, O., Mac Namee, B., Scanlon, M. 2018. Deep learning at the shallow end: Malware classification for non-domain experts. Digital Investigation, 26, S118–S126. doi:10.1016/j.diin.2018.04.024

Li, Y., Ma, R., Jiao, R. 2015. A hybrid malicious code detection method based on deep learning. International Journal of Security and Its Applications, 9, 205–216. doi:10.14257/ijsia.2015.9.5.21

Lysenko, S., Bobrovnikova, K., Nicheporuk, A., Shchuka, R. 2019. SVM-based technique for mobile malware detection. 2019 Computer Modeling and Intelligent Systems, Zaporizhzhia, Ukraine.

Mohaisen, A., Alrawi, O., Mohaisen, M. 2015. AMAL: High-fidelity, behavior-based automated malware analysis and classification. Computers and Security, 52, 251–266. doi:10.1016/j.cose.2015.04.001

Möser, M., Soska, K., Heilman, E., Lee, K., Heffan, H., Srivastava, S., Christin, N. 2018. An empirical analysis of traceability in the monero blockchain. Privacy Enhancing Technologies. 143–163. 10.1515/popets-2018-0025.

Pastrana, S.S.-T., Guillermo. 2019. A first look at the Crypto-Mining malware ecosystem: A decade of unrestricted wealth. IMC '19: Proceedings of the Internet Measurement Conference, Amsterdam, Netherlands.

Pluskal, O. 2015. Behavioural malware detection using efficient SVM implementation. Paper presented at the RACS, Conference on research in adaptive and convergent system, Prague, Czech Republic.

Raff, E., Zak, R., Cox, R., Sylvester, J., Yacci, P., Ward, R., Nicholas, C. 2016. An investigation of byte n-gram features for malware classification. Journal of Computer Virology and Hacking Techniques, 14, 1–20. doi:10.1007/s11416-016-0283-1

Rhode, M., Burnap, P., Jones, K. 2018. Early-stage malware prediction using recurrent neural networks. Computers and Security, 77, 578–594. doi:10.1016/j.cose.2018.05.010

Sabar, N.R., Yi, X., Song, A. 2018. A Bi-objective Hyper-heuristic support vector machines for big data Cyber-security. IEEE Access, 6, 10421–10431. doi:10.1109/access.2018.2801792

Sari, M.S.A.B.M., Maarof, M.A. 2017. Classification of malware family using decision tree algorithm. UTM Computing Proceedings: Innovations in Computing Technology and Applications.

Saxena, P., Fischer, C., Kumar, A., Tople, S. 2017. A traceability analysis of monero's blockchain. European Symposium on Research in Computer Security, 153–173.

Shijo, P.V., Salim, A. 2015. Integrated static and dynamic analysis for malware detection. Procedia Computer Science, 46, 804–811. doi:10.1016/j.procs.2015.02.149

Souri, A., Hosseini, R. 2018. A state-of-the-art survey of malware detection approaches using data mining techniques. Human-centric Computing and Information Sciences, 8. https://doi.org/10.1186/s13673-018-0125-x

Soviany, S., Scheianu, A., Suciu, G., Vulpe, A., Fratu, O., Istrate, C. 2018. Android malware detection and Crypto-mining recognition methodology with machine learning. IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC), Bucharest, Romania.

St'Astna, J., Tomasek, M. 2018. Assembling behavioral characteristics of malicious software. IEEE 14th International Scientific Conference on Informatics, INFORMATICS 2017-Proceedings, Poprad, Slovakia.

Stiborek, J., Pevný, T., Rehák, M. 2018. Probabilistic analysis of dynamic malware traces. Computers and Security, 74, 221–239. doi:10.1016/j.cose.2018.01.012

Taran, O., Rezaeifar, S., Voloshynovskiy, S. 2018. Bridging machine learning and cryptography in defense against adversarial attacks. Retrieved from https://arxiv.org/abs/1809.01715

Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Venkatraman, S. 2019. Robust intelligent malware detection using deep learning. IEEE Access, 7, 46717–46738. doi:10.1109/access.2019. 2906934

Vinayakumar, R., Soman, K.P. 2018. DeepMalNet: Evaluating shallow and deep networks for static PE malware detection. ICT Express, 4, 255–258. doi:10.1016/j.icte.2018.10.006

Wang, Y., Cai, W., Lyu, P., Shao, W. 2018. A combined static and dynamic analysis approach to detect malicious browser extensions. Security and Communication Networks, 1–16. doi:10.1155/2018/7087239

Xiaofeng, L., Xiao, Z., Fangshuo, J., Shengwei, Y., Jing, S. 2018. ASSCA: API based Sequence and Statistics features Combined malware detection Architecture. Procedia Computer Science, 129, 248–256. doi:10.1016/j.procs.2018.03.072

Zhang, K., Li, C., Wang, Y., Zhu, X., Wang, H. 2017. Collaborative support vector machine for malware detection. Procedia Computer Science, 108, 1682–1691. doi:10.1016/j.procs.2017.05.063

Zhang, R., Xue, R., Liu, L. 2019. Security and privacy on blockchain. ACM Computing Surveys, 52, 1–34. doi:https://doi.org/10.1145/3316481

Zhang, S.-H., Kuo, C.-C., Yang, C.-S. 2019. Static PE malware type classification using machine learning techniques. 2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA).