Proceedings of 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), IEEE, Iran, Rasht, (2015) September 8-10 ,Ahmadian M M, Shahriari H R and Ghaffarian S M.(Link)
DOI: 10.1109/ISCISC.2015.7387902.

Connection-Monitor & Connection-Breaker: A Novel Approach for Prevention and Detection of High Survivable Ransomwares

Mohammad Mehdi Ahmadian, Hamid Reza Shahriari, Seyed Mohammad Ghaffarian
Department of Computer Engineering and Information Technology
Amirkabir University of technology
Tehran, Iran
{ mm.Ahmadian, Shahriari, s.m.Ghaffarian }@ aut.ac.ir

***Abstract*— Ransomwares have become a growing threat in recent years, and this situation continues to worsen. It rose awareness on a particular class of malwares which extort a ransom in exchange for a captive asset. Most widespread ransomwares make an intensive use of data encryption. Basically, they encrypt various files on victim's hard drives, removable drives and mapped network shares before asking for a ransom to get the files decrypted. In this paper, at first we propose a comprehensive ransomware taxonomy. Then, based on this taxonomy and according to a principal feature which we discovered in high survivable ransomwares (HSR) in the key exchange protocol step, we present a novel approach for detecting high survivable ransomwares and preventing them from encrypting victim's data. Experimental evaluation demonstrates that our framework can detect variants of recent dangerous ransomwares.**

Keywords-component; ransomware; cryptovirology; malare detection; malware prevention; high survivable ransomwares

I. INTRODUCTION

Cybercriminals and malware writers have diversified their efforts to make money from their victims, using methods that have been well-established on desktops, laptops, tablets and mobile devices, this includes ransomware. "Ransomware is the name of a so called phenomenon. It has been built upon the two words ransom and malware" [1]. To define this word, one may give the following general definition: "a ransomware is a type of malware which restricts access to the computer system that it infects, and demands a ransom paid to the creator(s) of the malware in order for the restriction to be removed". Some forms of ransomware encrypt files on the system's hard drive (cryptoviral extortion, a threat originally envisioned by Adam Young and Moti Yung [2]), while some may simply lock the system and display messages intended to force the user into payment .

Anandrao said in [3] that "It does not appear that a properly designed cryptoviral extortion attack has ever been carried out to date immensely." Also Gazet said in [1] that "No ransomware has reached a sufficient complexity level to successfully become a perfect extortion mean. None of the ransomwares we have studied, presents a reliable perfect extortion scheme. An explanation of this may be that ransomwares' writers have a limited knowledge of cryptography." These statements were valid before 2013. But the CryptoLocker ransomware in 2013 showed that the situation has changed and malware developers have increased their cryptology knowledge.

In June 2013, McAfee released data showing that it had collected over 250,000 unique samples of ransomware in the first quarter of 2013, more than double the number it had obtained in the first quarter of 2012[4]. CryptoLocker surfaced in late-2013, had procured an estimated US$3 million before it was taken down [5]. Based on Bitcoin transaction information ZDNet estimated that the operators of CryptoLocker had procured about US$27 million from infected users [6].

Ransomwares have become a growing threat in recent years, and the situation continues to worsen. As we see in Fig.1 based on [7, 8, 9] the number of new, unique samples in the 2nd quarter of 2013 is greater than 350,000.

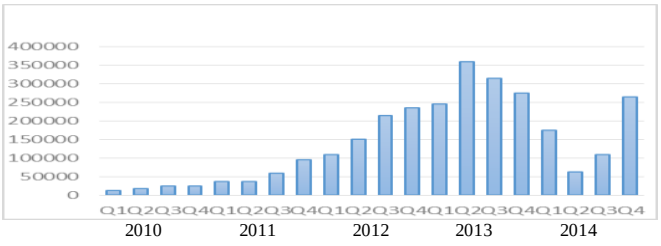

Figure 1. New Ransomware Samples

Ransomwares have been used for widespread mass extortion. Is the extortion scheme reliable? Are few resources and reverse-engineering able to break it? Do their writers make a thoughtful use of their creations and they don't have any weak point? These questions are some of the points that we discuss about ransomwares. "Battles of the future information warfare will be decided by the countries which have the leading edge in cryptovirological technologies and its countermeasures. This may be used to create panic by using methods such as rising a false nuclear alarm, may be used to block and encrypt military databases of the enemy nations. Also can be used to bring down communication in networks of

Proceedings of 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), IEEE, Iran, Rasht, (2015) September 8-10 ,Ahmadian M M, Shahriari H R and Ghaffarian S M.(Link)
DOI: 10.1109/ISCISC.2015.7387902.

enemies nation by causing denial-of-service attacks in large scale" [3], Hence attention to ransomwares and other types of cryptovirological attacks is crucial.

In this paper, we present a novel approach for the most dangerous ransomwares to detect their malicious activity and abort their encryption process before it starts. In summary, we make the following contributions:

- At first, in section 2, we present a novel ransomware taxonomy based on cryptovirological attacks and our studies on ransomwares, which we believe is enough comprehensive to cover all known types of ransomwares.
- Then in section 3, we present a novel approach for detecting HSRs that use domain generation algorithm (DGA).
- Finally we suggest a novel prevention and detection approach called "Connection-Monitor & Connection Breaker" (CM&CB) for the strongest type of ransomware, the HSR. Our experimental results based on a proof of concept implementation demonstrate the efficacy of the proposed approach to thwart the threat of the most dangerous types of ransomwares.

II. Proposed Ransomwar Taxonomy

In this section we describe our proposed comprehensive taxonomy.

A. *Non-Cryptographic Ransomware (NCR)*

Some ransomware payloads do not use encryption. In these cases, the payload is simply an application designed to restrict interaction with the system by locking the screen or even modifying the master boot record and/or partition table. Because of the weak techniques used by this type of ransomwares, their damages can be reversed without paying the ransom.

B. *CryptoGraphic Ransomware (CGR)*

Cryptographic Ransomware (CGR) use cryptographic algorithms to captivate valuable assets in exchange for a ransom. A typical scenario in this regard is that the malware will start encrypting the user data (including documents, images and etc.) silently. After encryption is complete the victim user is informed that all of his/her data are encrypted and can only be decrypted if he/she pays the ransom. We divide these ransomwares into three subtypes based on the cryptosystem they use.

B-1)Private-key cryptosystem ransomware (PrCR)

Some ransomwares use private-key cryptosystems such as classical ciphers, DES family or modern private-key cryptosystems to captivate victim assets. For example CryptorBit ransomware which was discovered at December 2013 used a self-designed classical cipher cryptosystems similar to polyalphabetic substitution ciphers in just first 512 bytes of target files. As Young and Yung described in [10] about the one-half virus the vulnerability of PrCR lies in the fact that they are inherently scrutable once found. When a malware analyst gets hold of a ransomware, the analyst learns the operations that the malware is programmed to perform. The attack carried out by the PrCR is repairable for the simple reason that the view of the ransomware writer and the view of the malware analyst are symmetric (*Fig. 2*).

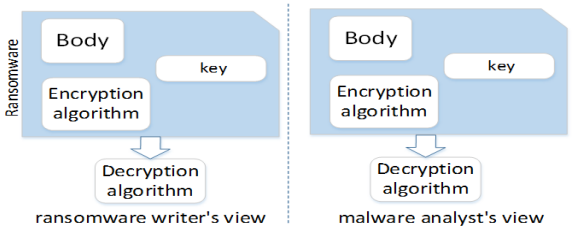

Figure 2. Symmetric views of the ransomware

"The key needed to be removed from the malware analyst's view but not from the ransomware writer's view. This would make the views asymmetric" [10]. it is computationally possible to recover the secret key (or Session Key) used for symmetric encryption by reverse engineering or sometimes with brute forcing (According to Gazet tests in [1], in some ransomware infection such as Gpcode.ab it is possible to smartly bruteforce a file within fifteen minutes and find the key). Some ransomwares such as Trojan.Win32.Filecode don't need to store a key; part of target file is used as the key then we put this ransomware in PrCR subtype same as CryptorBit. Another popular encryption algorithm is the XOR cipher, which of course is trivial for the analysts to break it.

The fact that the whole encryption process (including the secret key) is visible to the analyst, is a major flaw in the ransomware that use symmetric private-key encryption. Other examples of ransomware in this type are LZR, AIDS, and KOH [2].

B-2) Public-key cryptosystem ransomware (PuCR)

The notion of using public key cryptosystem for such attacks was introduced in 1996 by Young and Yung [2]. The two believed the AIDS ransomware was not effective due to its use of symmetric cryptography. In May 2005 examples of these ransomwares appeared. By mid-2006, some ransomwares such as Gpcode, TROJ.RANSOM.A, Archiveus, Krotten, Cryzip and MayArchive began utilizing more sophisticated RSA encryption schemes, with ever-increasing key-sizes. Gpcode.AG, which was detected in June 2006, performed encryption with a 660-bit RSA public key. In June 2008, a variant known as Gpcode.AK was detected that used a 1024-bit RSA key, which was believed to be large enough to be computationally infeasible to break without a concerted distributed effort.

In these ransomware, a pair of keys known as the public-key and private-key (or encryption-key and decryption-key) is generated using an asymmetric cryptosystem (such as the RSA), and the public-key is safely placed in the payload of the

Proceedings of 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), IEEE, Iran, Rasht, (2015) September 8-10 ,Ahmadian M M, Shahriari H R and Ghaffarian S M.(Link)
DOI: 10.1109/ISCISC.2015.7387902.

PuCR, which is used for encrypting data on the victim machine; while the private-key is kept secret by the writer. In this manner, the encrypted data can only be recovered by the writer of the PuCR who holds the private-key (or decryption-key)(assuming no fresh backup or volume shadow copies exists). This key pair is generated only once, before the PuCR is deployed. As a result, malware analysts will not be able to determine the private-key from monitoring the ransomware's operations (see Fig. 3).

Following encryption of the victim's data, the PuCR notifies the user and provides contact information of the PuCR writer. Once contacted, the PuCR writer demands a ransom in exchange for the private-key. Once the private-key is obtained, the user is able to recover all encrypted data.

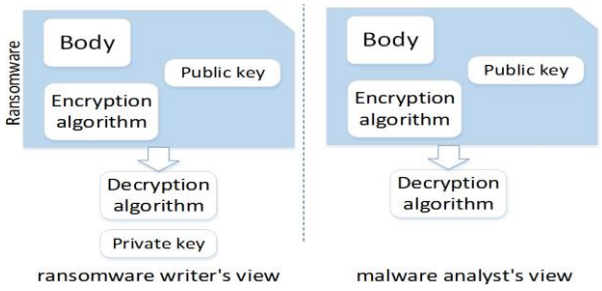

Figure 3. Asymmetric views of the ransomware

A drawback from the perspective of the PuCR writer is that he cannot free one victim without potentially freeing all the victims, because the freed victim could publish the private-key. This drawback can somewhat be solved if the PuCR generates multiple key pairs. The PuCR could randomly (or otherwise) choose a key from the set of keys, thereby allowing the PuCR writer to free some victims without freeing all the rest. This gives the PuCR writer more control over who he can selectively free. However carrying many keys is expensive. Also, note that this is only a partial solution, since users may cooperate with each other, and eventually all private-keys will be published. Another drawback is the fact that asymmetric encryption is much slower than symmetric encryption schemes.

B-3)Hybrid cryptosystem ransomware(HCR)

To solve the aforementioned problems, some ransomware writers employ hybrid cryptosystems. In this case, again a pair of asymmetric keys are generated and the public-key is place in the malware payload. But, for the data encryption process, a random secret-key is generated on each victim machine, and the captive data are encrypted using this key and a fast symmetric cipher. The random generated secret-key is encrypted using the public-key and only stored in this way. In this case the adversary is not required to disclose his private-key. The malware writer demands the ransom and for decryption, the cipher text of the random secret-key is sufficient. He then decrypts the secret-key using the private-key and sends it back to the victim. In this method, with a high probability each victim has a unique key, and so publishing of the decryption key is of no help to other victims.

III. CONNECTION-MONITOR& CONNECTION-BREAKER APPROACH

After describing the taxonomy of ransomware, it is clear that the most dangerous ransomware are hybrid cryptosystem ransomwares or HCRs. In this paper we propose CM&CB, a new framework to detect the most dangerous types of ransomware and prevent them from encrypting the victim's files. In this section, first we define the targeted types of ransomware, and then describe our proposed framework.

A. High survivable ransomwares (HSR)

Below we describe the requirements that need to be fulfilled for an effective mass extortion means:

− The ransomware infects users' computer, thus it should be considered as compromised and harmful.

− Ransomware writer should be the only one able to reverse the infection. In order to claim a ransom, the malware need to possess a reliable extortion means. If a victim can get rid of the infection herself, she will not pay the ransom [1]. For a perfect extortion, the decryption-key must never be stored on the victim's machine, because advanced users or malware analysts, with a least of reverse-engineering skills will be able to restore the system into a clean state.

In some cases, we analyzed some ransomwares (such as CryptoDefense) which after generating the secret key on the victim's machine, they send it to the malware command & control servers (C&Cs) which is a major design flaw that allows the private-key to be recovered. Thus based on our definition these are not perfect extortion mechanisms and moreover, not a strong HCR. As we described it is possible to extract the secret-key (one sample is described in [1]) from the victim's host before it could send it to the C&Cs and delete it from the victim's machine. As described in earlier sections, the PrCR are also not perfect extortion mechanisms.

− Freeing one victim should not help other victims to get rid of the infection. We are dealing with mass infection, so if one victim accepts to pay a ransom and receives a decryption tool or key, passing it to other victims should be of no help to them [1]. Writing ransomwares which have a strong unique encryption key or module in every different victim is very cumbersome and so hard for PrCR writers. As we described if a ransomware use same encryption keys in its victims then it is not a perfect extortion.

According to the three properties for perfect extortion, and based on [2], survivability is a common issue for all ransomwares. The following is the definition of ransomware with "high survivability property".

Definition 1: A ransomware has the "high survivability" property if it can maintain control over a critical host resource RC such that it grants access to RC solely when it is needed, and such that if the ransomware is modified or removed, RC is rendered permanently inaccessible and the decryption process can be completed only by Command & Control server (C&C) key while the ransom is paid.

Proceedings of 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), IEEE, Iran, Rasht, (2015) September 8-10 ,Ahmadian M M, Shahriari H R and Ghaffarian S M.(Link)
DOI: 10.1109/ISCISC.2015.7387902.

Among publicized ransomwares existing in malware databases in the last decade, the high survivable ransomwares are in the HCR subtype. In this work, we propose a detection and prevention framework to counter the high survivable ransomware (HSR). Moreover, our framework can detect every ransomware which uses a key exchange procedure for its operation.

B. Overview of CM&CB approach

Adleman has shown that detecting viruses is an intractable problem, and that it seems unlikely that protection-systems predicated on virus detection will be successful [11]. Young and Yung have shown that even if a HCR is detected in a given system, it may be a computationally intractable problem to reverse its effect on the host system (assuming the asymmetric cryptography is strong) [2]. Our proposed framework is able to detect all current HSRs (which are published so far) before the encryption process starts, thus thwarting the operation completely. In an earlier section, we described the general operation of HCRs. During our analysis, we have dissected the HCR attack protocol which lead us to find an effective feature in HCRs, based on this feature we have designed and implemented our framework. To understand this feature we review the HCR attack process in more detail.

Step 1 (Seek for victim): At first the HCR is propagated via mail spams or other way of propagation. For example the CryptoLocker is typically spread through emails sent to company email addresses that pretend to be customer support related issues from Fedex, UPS, DHS. These emails would contain a zip attachment that when opened would infect the computer.

Step 2 (execution): In this step the HCR is executed by an unaware user by social engineering methods. For example the CryptoLocker zip files contain executables that are disguised as PDF files as they have a PDF icon and are typically named something like FORM_101513.pdf.exe. Since Microsoft does not show extensions by default, they look like normal PDF files and people open them. After execution the HCR mostly generates a large random symmetric session key K_s and initial vector IV. In some sophisticated HCR like Cryptolocker in this step the HCR tries to delete the victim's volume shadow copies, so the restoration will be disabled.

Step 3 (public key exchange): According to our description in PuCR there are a lot of limitations for ransomware writers to put pair key in ransomwares in order that, in this step the HCR will then attempt to find a live C&C or his public directory to receive unique public key K_{pu}. For example Cryptolocker attempts to find a live C&C by connecting to domains generated by a DGA. Some examples of domain names that the DGA will generate are kjqwymybbdrew.biz and jkaeaxjmnxvpv.ru. Once a live C&C server is discovered it will communicate with it and receive a public encryption key that will be used to encrypt data files.

Step 4 (Encryption): As soon as the infection specific public key has been obtained, the victim's data will be encrypted using Ks and later get chained using a chaining mode as CBC. The actual information is then may be deleted or overwritten. As in (1), The Initialization Vector is appended with the symmetric key and encrypted using the virus writer's public key.

$$M' = E_{Kpu}(\{IV, Ks\}) \quad (1)$$

The encrypted plaintext (M') is then held for ransom [12].

Step 5 (Display message): After infection the M' and anonymous ways to contact the HCR writer are displayed on the victim's screen.

Step 6 (Decryption): If the victim agrees on the condition to pay demanded ransom, he should transmits M' to the HCR writer. HCR writer then decrypts the pair by using the corresponding private key K_{pr} and sends back the pair to the victim. In some samples we see the HCR writer use an executable application for decryption instead of sending the {IV, Ks}.

After we analyzing more than 40 recent ransomwares and considering the current anti-malware technologies and according to malwares and anti-malwares futurology, first version of our framework is designed based on an idea which is related to the public key exchange stage in above protocol. In this stage most advanced and evasive ransomwares use DGA, because embedding a static list of C&C candidates within ransomwares poses problems for cybercriminals should the malicious binary eventually be captured and analyzed by security vendors and analysts. To overcome this frailty, the majority of modern ransomwares have turned away from hard-coded lists and is designed to use DGAs. In our first framework we designed a connection monitor which can detect DNS domain request which generated by DGAs. *Fig. 4* shows the pseudo-code of the sample routines used as DGA.

```
suffix = ["anj", "ebf", "arm", "pra", "aym", "unj","ulj",
"uag", "esp", "kot", "onv", "edc"];

def generate_daily_domain():
    t = GetLocalTime();
    p = 8;
    return generate_domain(t, p);
def scramble_date(t, p):
    return (((t.month ^ t.day) + t.day)*p) +t.day + t.year;
def generate_domain(t, p):
    if t.year < 2014:
        t.year = 2014;
    s = scramble_date(t, p);
    c1 = (((t.year >> 2) & 0x3fc0) + s) % 25 + 'a';c2 =
(t.month + s) % 10 + 'a';c3 = ((t.year & 0xff) + s) % 25 +
'a'
    if t.day*2 < '0' || t.day*2 > '9':
        c4 = (t.day*2) % 25 + 'a'
    else:
        c4 = t.day % 10 + '1'
    return c1 +'h'+c2+c3+'x'+c4+suffix[t.month - 1]
```

Figure 4. DGA pseudo-code example[13]

We used Markov chains and a model of character to character transitions from English and Persian texts which writes with English alphabet for model training phase. For example, we find out how common it is for there to be a 'h'

Proceedings of 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), IEEE, Iran, Rasht, (2015) September 8-10 ,Ahmadian M M, Shahriari H R and Ghaffarian S M.(Link)
DOI: 10.1109/ISCISC.2015.7387902.

after a 't' (pretty common). In English, we expect that after a 'q', we will get a 'u'. If we get a 'q' followed by something other than a 'u', this will happen with very low probability, and therefore it should be pretty alarming. Normalize the counts in tables so that we have a probability. Then for a request, walk through the matrix and compute the product of the transitions we take. Then normalize by the length of the domain request. When the number is low, we likely have a gibberish query. We extend Rob Renaud gibberish detector implementation for our simple framework to detect DGAs DNS request with high detection rate (this idea can be improved by botnet papers such as [14]). It's clear that no benign software needs to connect with its server with DGA domains. Based on this idea we could notify the users DGA DNS requests and thwart the HSRs which designed based on DGAs. With this approach we could detect all the HSRs which they request for domain based on DGAs.

As you see in *Fig. 5* when a ransomware wants to connect to his C&C with DGA the connection monitor verifier (CMV) with the help of DGA detector detects suspicious connection. One another feature which is important in DGA malicious requests is these algorithms generate and request many domains connection in a short time. Then the suspicious connection notifier (SCN) will show the user a suspicious connection and all the user can break the connection and report this suspicious connection address to experts. DGA detector framework beside all the benefits needs a high precision for decreasing false positives. In the other hand some domains are not gibberish sometimes they are in a different language then this framework needs more effort to be a real acceptable popular framework.

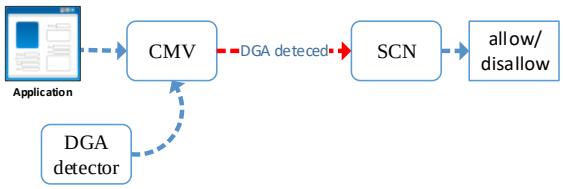

Figure 5. Architecture of DGA detector framework

In the next step we suggest an extended framework based on a same idea which is related to the public key exchange stage in HCR protocol. With this simple but novel idea which is never be proposed for ransomware detection and also malware detection we could successfully detect the current dangerous HCRs such as Cryptolocker, Cryptolocker2, Cryptowall, Cryptowall2 and thwart their encryption process before it started.

In this framework we have implemented a connection-monitor which check all the applications especially the new or untrusted executable files in Windows (as with the majority of ransomware is targeted at Microsoft Windows operating system) (*Fig. 7*). In a simple definition connection monitor approach check all the out coming traffic of the executables and ask the user for accept the connection or ignore it. In the advanced mode of our framework we designed an augmented code signing certificate. Beside the common code signing for integrity checking we suggest to the developers send their application required connection addresses to their certificate authority (CA). The CA after a simple address checking will verify the list as verified required connection addresses (VRCA) (*Fig. 6*).

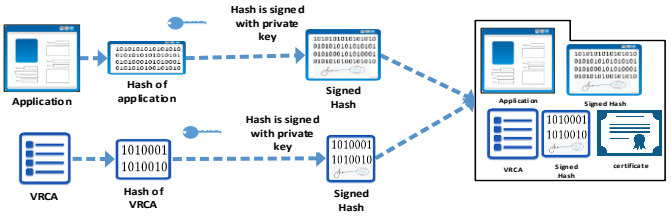

Figure 6. Augmented code signing certificate

For example Microsoft Office productions connect to www.microsoft.com and its subdomains for their updates then it will be verified in the VRCA. When Microsoft OneNote wants to update itself because of connection monitor verifies the valid connect address via its VRCA transparently this application will connect without any connection breaking. But in the other hand for example when a ransomware like Cryptolocker wants to connect to his C&C as it has no valid VRCA then the SCN will show the user a suspicious connection and all the user can break the connection and report this suspicious connection address to experts.

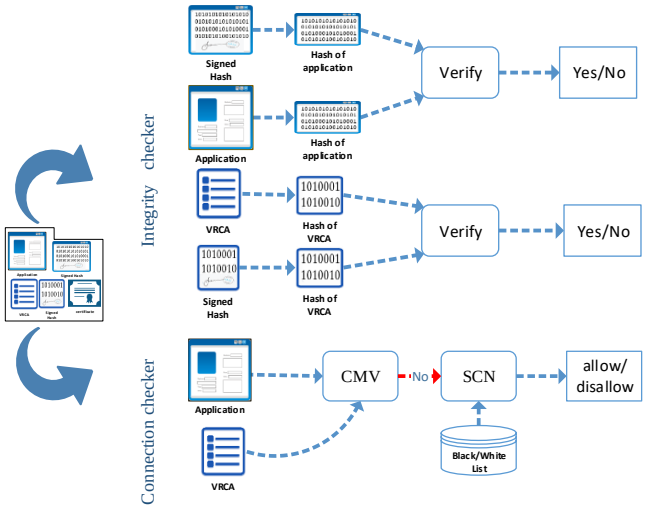

Figure 7. Architecture of preposed framework

If the HCR at first starts encrypting data and then wants to exchange the Kpu with connection breaking this process will be broken and the IV, Ks will be reminded within every instance of the HCR then a malware analyst can always find the contents of the main HCR body, simply by the decryption directly, using the stored keys.

Proceedings of 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), IEEE, Iran, Rasht, (2015) September 8-10 ,Ahmadian M M, Shahriari H R and Ghaffarian S M.(Link)
DOI: 10.1109/ISCISC.2015.7387902.

IV. EVALUATION

In this section, we present experimental results and discuss our experiences with our novel approach. In particular, we assess how effective the proposed framework is detecting and preventing HCRs from encrypting user's data with hybrid cryptosystems. There are some ransomware detector such as Hitman pro kickstart, HitmanPro CryptoGuard, BitDefender AntiCryptoWall but all of them are signature-based and cannot detect new or unknown ransomwares. According to this fact there is no ransomware detector which can detect new or unknown ransomwares we couldn't compare our framework detection result with other tools or frameworks. The only tool which currently can prevent new ransomwares or other types of malwares is CryptoPrevent which actually it is not a ransomware detector, it is only a software-restriction policy tool for expert users so working with it for normal users is so complicated. But over framework is so simple and the only windows which needs user interaction is SCN.

To demonstrate that our system is effective in detecting HSRs, we test our idea with more than 20 new common ransomware samples. This framework achieved to detect all of the HSRs before the public key exchange is completed and thwart their encrypting. The proposed approach detection rate is 100% with 0% false negative in HSR detection. An overview of these tests is provided in Table 1. These samples were selected (from malwaretips.com and BleepingComputer.com) because they are very popular and so complicated. In Table 1 detection means detecting before public key exchange and thwarting the encryption.

TABLE I. PROPOSED FRAMEWORK EXPERIMENTAL RESULTS

Ransomware Name	Ransomware Type				HSR	Detection
	HCR	*PuCR*	*PrCR*	*NCR*		
Cryptolocker	√	×	×	×	√	√
Cryptolocker 2	√	×	×	×	√	√
Cryptolocker 3	√	×	×	×	√	√
Cryptowall	√	×	×	×	×	√
Cryptowall 2	√	×	×	×	√	√
Cryptowall 3	√	×	×	×	√	√
CoinVault	√	×	×	×	√	√
CryptoGraphic Locker	√	×	×	×	×	√
CryptoDefense	√	×	×	×	×	×
CryptoDefense 2	√	×	×	×	√	√
CryptorBit	×	×	√	×	×	×
TorrentLocker (original)	×	×	√	×	×	×
TorrentLocker	√	×	×	×	√	√
ACCDFISA	×	×	√	×	×	×
BuyUnlockCode	√	×	×	×	×	×
CryptoFortress	√	×	×	×	×	×
PClock2	×	×	√	×	×	×
Critroni(CTB Locker)	√	×	×	×	×	×
ComputerCrime&IntellectualProperty Section	×	×	×	√	×	×
Harasom	×	×	√	×	×	×

V. DISCUSSION

Because of the background operation of the CMV in the proposed framework, this process is transparent from users. Also the SCN won't be much of bother for users because the domains are first checked with a black-list and a white-list, and if the domain is missing in both, then the user is prompted. Of course the proposed approach has one limitation: The current design can only prevent HSRs from captivating data, but non-HSRs like CryptoDefense can accomplish their attack. Fortunately, as discussed earlier, the damages caused by non-HSRs are reversible and anti-malware products are able to fix the problem and decrypt the captivated data without paying the ransom. For example the malware writer had a flaw in the CryptoDefense program that left the K_{pr} on the victim's host. By tools such as Emsisoft Decryptor it's possible to extract the K_{pr}.

"This is evidence that the malware problem is fundamentally a cultural problem. Even though people are educated and understand well concepts such as the physical security and the necessary maintenance of a car, they do not understand the consequences of irresponsible behavior when using a computer" [13]. In these days the user awareness is important to counter malware infections. Frameworks like the CM&CB act in this regard.

VI. CONCLUSION

In this paper we presented several techniques to counter the threat caused by dangerous ransomware. The proposed techniques include a DGA-detector, and a novel monitoring framework called CM&CB to detect and prevent damage by the most dangerous ransomware. The key observation for the success of this approach is that the operation of HSRs relies on a key-exchange step. By monitoring and blocking this step, the whole operation of the HSR is thwarted.

The main advantages of the proposed idea can be summarized as following. First, this framework is the first framework which is designed focused on the issue of ransomwares, by monitoring suspicious connections and preventing them from encrypting the victim's data. The experimental evaluations show that the proposed framework can successfully thwart the most dangerous HSRs, which was an open problem in the field of malware mitigation. Currently, further research is taking place on developing and more extensive evaluation. In addition, this framework is also useful in the detection of other types of malicious software, such as Bitcoin-mining malwares, botnets, drive-by download malwares and etc. Because the idea of issuing this kind of augmented certificate is not specific to HSRs, it can be a useful defensive mechanism against many other threats. Our long term objective is to extend this framework by adding another 17 HSR features to detect new and unknown sophisticated HSRs which they will not detect with only key-exchange step feature.

REFERENCES

Proceedings of 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), IEEE, Iran, Rasht, (2015) September 8-10 ,Ahmadian M M, Shahriari H R and Ghaffarian S M.(Link)
DOI: 10.1109/ISCISC.2015.7387902.

[1] Gazet, Alexandre. "Comparative analysis of various ransomware virii." Journal in computer virology 6.1 (2010): 77-90.

[2] Young, Adam, and Moti Yung. "Cryptovirology: Extortion-based security threats and countermeasures." Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on. IEEE, 1996.

[3] Shivale, Saurabh Anandrao. "Cryptovirology: Virus Approach." arXiv preprint arXiv:1108.2482 (2011).

[4] "Update: McAfee: Cyber criminals using Android malware and ransomware the most". InfoWorld. Retrieved 16 September 2013.

[5] "Cryptolocker victims to get files back for free". BBC News. 6 August 2014. Retrieved 18 August 2014.

[6] Violet Blue (December 22, 2013). "CryptoLocker's crimewave: A trail of millions in laundered Bitcoin". ZDNet. Retrieved 2013-12-23.

[7] McAfee Threats Report: February 2015, By McAfee Labs,Page 38,2015.

[8] McAfee Threats Report: Third Quarter 2013, By McAfee Labs,Page 19,2013.

[9] McAfee Threats Report: Second Quarter 2014, By McAfee Labs,Page 21 ,2014

[10] Young, Adam, and Moti Yung. Malicious cryptography: Exposing cryptovirology. John Wiley & Sons, 2004.

[11] Adleman, Leonard M. "An abstract theory of computer viruses." Proceedings on Advances in cryptology. Springer-Verlag New York, Inc., 1990.

[12] Abidin, Shafiqul, Rajeev Kumar, and Varun Tiwari. "A Review Report on Cryptovirology and Cryptography." International Journal of Scientific & Engineering Research 3.11 (2012): 1.

[13] Stone-Gross, Brett, et al. "Your botnet is my botnet: analysis of a botnet takeover." Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009.

[14] Yadav, S., Ashwath K. K. R., and Supranamaya R. . "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis." Networking, IEEE/ACM Transactions on 20.5 (2012): 1663-1677.