2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)
August 28-29, 2018; Shahid Rajaee Teacher Training University - Tehran, Iran

A Novel Approach for Detecting DGA-based Ransomwares

Saeid Salehi
Computer Engineering & Information Technology Department
Amirkabir University of Technology
Tehran, Iran
saeid_salehi@aut.ac.ir

HamidReza Shahriari
Computer Engineering & Information Technology Department
Amirkabir University of Technology
Tehran, Iran
Shahriari@aut.ac.ir

Mohammad Mehdi Ahmadian
Computer Engineering & Information Technology Department
Amirkabir University of Technology
Tehran, Iran
mm.ahmadian@aut.ac.ir

Ladan Tazik
Computer Engineering & Information Technology Department
Amirkabir University of Technology
Tehran, Iran
ladan.tazik @aut.ac.ir

***Abstract*— Nowadays, hybrid cryptosystem ransomware, as well as botnets, utilize domain-generation algorithms to communicate with the command and control (C&C) server to exchange public key and perform their malicious actions. We present an approach for detecting domain-generation-algorithm-based ransomware for the first time. By running instances of this type of ransomware in a test environment, we analyze their behavior, especially in the DNS traffic segment, which leads us to derive several behavioral characteristics. Among these features, we can point to "random and gibberish characters" in the requested domains; But using this feature is not easy as it can yield a lot of false positives. Our new and innovative approach to solving this challenge is to measure "Frequency of Different Domains Generation" and "Repetition of Same Domains in a Time Interval". With the help of these criteria, we show that our method is more effective. The proposed approach can be used to detect botnets and other DGA-based malwares. Moreover, our approach detects ransomwares in their early phase of activity (i.e. before encrypting user data). Ultimately, we propose these features as a framework for identifying these ransomwares with high detection accuracy and low false positives rate.**

Keywords*— *Ransomware, Malware, domain generation algorithm, malware detection, malware analysis, behavioral analysis.

I. INTRODUCTION

This This Today's explosive growth of various types of malware and the development of advanced computing and communication technologies and the necessity for more and more security levels have created a major challenge in the anti-malware industry. One of the reasons for the growth of security threats in cyberspace is not only implementing new and unknown patterns in the development of malware but also the interval between the time of the release of new malware and the time of detection and reporting it by anti-malware companies. In the past few years, a particular kind of malware, called ransomware, has been a growing trend, although the concept of ransomware is not new (an example of such attacks was recorded at the end of 1980) [1].

Since 2009, cyber attacks against organizations have increased, and in 2013 nearly 91% of all organizations were targeted by cybercrime attacks. Due to losses and reports made by Malware Bytes[1] at the end of 2013, ransomware was the number one security threat this year. The ransomware was simple at first, but rather than just locking the user's screen, ransomware gradually began to encrypt whole or some parts of user's information system. In this case, if the user has not backed up her files, she has to pay the ransom to decrypt her system information [2].

Many victims who feel that their data is very important to be ignored and do not have a backup of their files are forced to pay the ransom to the attacker. For example, when in 2012 Symantec was able to dismantle a C&C network used by the CryptoDefense ransomware family Subsequent studies showed that 2.9% of the victims of the 68,000 who were infected were forced to pay the ransom [3].

In April 2016, statisticians from users who paid extortion to CryptoLocker were provided by researchers at the University of Kent, as shown in Figure 1, which percentage of the victims pay the ransom and what percentage of them did not pay the ransom [8].

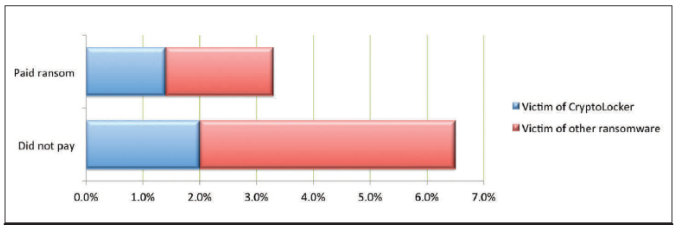

Figure 1. The Percentage of Victims Paid/did not Pay the Ransom [8]

The recent success of the ransomware has led to a large increase in the number of ransomware families in recent years, for example, CryptoWall 3.0 has been known to be the leading family of ransomware in the world, caused a loss of $ 325 million [3]. As another example, Sony's ransomware attack attracted the attention of big media, and the United States government even officially asserted that North Korea was behind the attack [4].

[1] www.malwarebytes.org

978-1-5386-7582-3/18/$31.00 ©2018 IEEE

Ransomware is continuing their growing trend, and it's safe to say that is one of the most dangerous malware types today, and cybercriminals have a lot of interest in publishing such malware because of the high income and extortion of victims. For example, the WannaCry ransomware, which was released in May 2017 has infected more than 300,000 machines in 150 countries and with total damages ranging from hundreds of millions to billions of dollars. The growing trend for new ransomware from 2010 to the end of 2017 [9-10-11-12] is sketched in Figure 2.

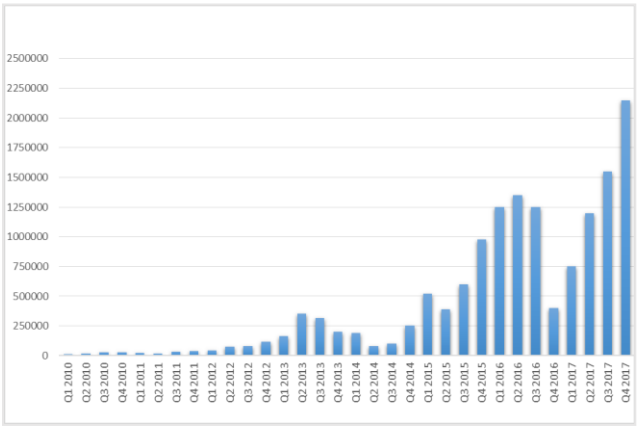

Figure 2. New Ransomware Samples

In 2016, Ahmadian et al. [2] presented 2entFOX framework for detecting survivable ransomware, based on extracting 20 features from the static and dynamic analysis of Windows ransomware. This study was conducted on 20 samples of ransomware, and the 2entFOX framework detect survivable ransomware with high detection rate and low false positive rate.

Another different work [6] was done in 2016, which their detection approach was based on Honeypot techniques. The authors of the paper try to determine a baseline for normal activities by analyzing the behavior of users and used honeypots as a prey to identify suspicious traffic. If an abnormal behavior is reported, the approach for detecting the ransomware is applied

In 2017, Kharraz et al. [1] presents a method for detecting Crypto and Locker ransomware. This detection system, called UNVEIL, was a dynamic analysis system; the important point of this analysis was that an attacker must manipulate files or the desktop of a user's system to perform a successful attack. UNVEIL automatically generates an artificial user environment and recognizes when the ransomware interacts with user's data. At the same time, this approach follows user's desktop changes that indicate ransomware-like behavior. The results of this study showed that UNVEIL could significantly detect unknown ransomware that many anti-malware could not detect.

Another work with a different approach to detecting ransomware is [7]. The authors suggested that detection can be done in the early stages of attack through the delivery channels of ransomware, such as Exploit Kit. They analyze crawling patterns (such as the listing of file path, dropped files, network activities, ransom request notes, etc.). These patterns were used to extract features for malware classification. In this research, they used machine learning algorithms (random forest, Bayesian and J48 decision tree). Experimental were performed in tightly bound and moderate bound modes, the best detection rate was achieved for random forest (in the tightly bound mode with an accuracy of 94% and in the moderate bound mode with an accuracy of 91%).

Unfortunately, little research has been done in the area of detecting DGAs in ransomware, but the opposite in the botnets, DGAs detection methods have reached maturity [14]. This was one of the motivations that made us study and evaluate the DGA-based ransomware (DGR). The authors of the paper [5] presented a method for detecting domain-generation algorithms (DGA), which the DGA detecting module extends Rob Renaud gibberish detector implementation based on Markov Chain, but using this method yields to high false positive error rate, Because, there are too many legitimate domains that have gibberish letters in the requested domain, and moreover, there are much ransomware that their requested domains do not have this feature, and the domains are quite normal, therefore, this algorithm will not be able to detect these kinds of ransomware.

In this paper, we will use other behavioral features along with this feature to solve the mentioned challenge and increase the detection rate and reduce the false positives; one of the other benefits of our proposed method is that it can be used to detect other malware, such as botnets and other DGA-based malware. Also our approach can detect ransomware in the early phase activity like before exchanging public key.

In the remainder of this paper, in Section 2, we will describe the definitions and concepts in this field; then, in Section 3, we will propose a new approach and then introduce the proposed architecture of the approach and ultimately evaluate the results of this research. Ultimately, implementation and evaluation of our method is given in Section 4.

II. DEFINITIONS AND BASIC CONCEPTS

In this section, we try to summarize the introduction of ransomware and their types, as well as explaining the concept of domain generation algorithms and their categorization.

A. Ransomware

Generally, ransomware is a kind of malware that restricts access to the system or its resources after infecting a computer system, and then the ransomware designer fixes the constraint in exchange for ransom from the victim. Various types of categories have been provided for the ransomware, but the most complete categorization is provided in [5], which is given below.

1) Categorization of Ransomware

1. **Non-Cryptographic Ransomware (NCR):** Some ransomware never uses cryptographic methods to encrypt their files in their operating process. These ransomware extort the user in some way such as locking the user's screen or changing the master boot record or partition table**.**
2. **Cryptographic Ransomware (CGR):** Such ransomware uses cryptographic algorithms to capture and seize access to targeted resources in the victim's system. In the general scenario, these ransomware secretly encrypt

victim files, and after completing the infection, they will inform the user about ransomware and start to extort payment from the victim. These types of ransomware are divided into three categories, depending on the type of exploitation of the cryptographic system:

2.1. **Private-key Cryptosystem Ransomware (PrCR):** This type of ransomware encrypts the victim's assets using private key cryptosystems such as classic cryptosystems, the DES cryptosystem, or even modern private key cryptosystems.

2.2. **Public-key Cryptosystem Ransomware (PuCR):** These types of ransomware use public key cryptosystems such as RSA for their encryption operations. They are more dangerous and more complex than the PrCR ransomware because the user's data is encrypted with the public key and the private key remains with the malware designer. In PrCR, the private key is on the victim's system and the malware analyst could access it and analyze the cryptosystem, but in the PuCR ransomware, the private key is in the designer's hands, and the victim needs to access the private key to retrieve their data, thus has to pay the ransom.

2.3. **Hybrid Cryptosystem Ransomware (HCR):** PrCR ransomware is much faster in cryptographic operations than PuCR but is detected by malware analysts because of a private key on the victim's system. In order to benefit from the advantages and reduce the weaknesses of aforementioned cryptosystems, ransomware developers use the PuCR and PuCR methods both. We can point out to high survivable ransomware (HSR) instance of HCR category.

3. **High Survivable Ransomware (HSR)**: A ransomware has a high survivability property if it gains control over its critical host assets and captures host resources exclusively, and if the ransomware is erased or altered by someone other than the attacker (designer of the malware), access to resources disappears for good. On the other hand, the encryption process for this kind of ransomware should be done only after the ransom has been paid through a solution that the developer of the malware has already set up, such as the C & C server. In other words, survival of the valuable resources of the victim who is infected with HSR depends on the survival of the HSR on the victim's system. In the victim's system infected with HSR, retrieving resources and clearing the system without the help of an attacker is not possible, and therefore the victim has to pay the money in exchange for the recovery of resources and system cleanup [5].

2) Domain Generation Algorithms(DGA)

Recently, ransomware, as well as botnets, use domain-generation algorithms (DGAs) to perform their malicious behaviors. DGA is code piece that is used to produce a large number of Internet domain names periodically. The main purpose of using these algorithms in malware is that the malware overcomes the methods of identifying the hardcoded list, which is easily identified by black/ whitelist security mechanisms. Since the domains produced by these algorithms are generally pseudo-random and the domain-generation frequency is very high, these algorithms are not detected by the current black/whitelist filtering mechanisms, which has led much ransomware to use these domains in different ways.

Taxonomy of DGAs

In [13], domain-generation algorithms are categorized into binary-based and scripted-based DGAs, which we briefly describe below:

- **Binary-based DGA:** they are embedded as a binary code in the malware itself; therefore, they are activated when the malware is successfully installed on the user's system. After installation, the DGA module is launched, which generates a number of pseudo-random domains based on the predetermined value of seed. The main purpose of this category is to start a C & C server channel with the malware programmer. Also, this type of DGAs can help in HTTP-based phishing attacks. Examples include malware like GameOver Zeus, Cryptolocker, PushDo, and Ramdo.

- **Script-based DGA:** Script-based DGAs are embedded in a JavaScript code loaded in the browser. When the user opens a malicious web page in a browser, JavaScript runs and activates the DGA to generate links. Attackers use these types of DGAs to generate HTTP URLs containing random domain names (hostnames) for drive-by download attacks. Script-based domain generation algorithms are commonly used for early infections. The attacker in the first step deceives the victim to visit a malicious website (receiving a phishing email that contains an illegitimate URL). Then, when users click on the embedded URL, the browser is forced to visit the domain at the URL. When the browser opens the malicious domain, a JavaScript is rendered in the browser, performing two primary functions. First, it generates a domain: the JavaScript contains DGA logic, which it executes to output the pseudorandom domain on the fly. Then, it dynamically generates an iframe, which loads the URL built using the generated domain name through DGA and appends the server path. The server path is predefined in the JavaScript. When the iframe is generated and the URL is loaded, the browser implicitly sends the HTTP request to the destination server that hosts the resource specified in the URL.

III. PROPOSED DETECTION APPROACH

After examining different instances of various ransomwares, especially HSR ransomware, we found that, based on the current state of the ransomware and the upcoming process, malware tends to use DGA more than the embedded static list to store C&C server addresses. What is evident is that a static list has a number of limitations, including the need for additional space on the malware file, the limitation of the number of addresses, the identification of addresses after the first infections. Therefore, if appropriate algorithms are used to identify the addresses produced by DGA algorithms, these malware can be detected before the public key exchange process so their malicious process can be stopped. By analyzing the DGA-based ransomware (DGR) in the test environment, we tried to

analyze their behavior, especially in their DNS traffic section. According to the DNS traffic analysis of ransomware, we have achieved three features of "random and gibberish characters in the requested domains," and measuring “Frequency of Different Domains Generation," and " Repetition of Same Domains in a Time Interval”, which are simple and effective. Although the use of random and gibberish letters in the requested domains yield to false positive rate and reduces the detection accuracy, the idea of using other features introduced will solve the challenge. The architecture of the proposed method is further elaborated.

A. The Architecture of Proposed Approach

In this section, we intend to propose a detection framework to detect DGA-based ransomware (DGR). We focus on HSR ransomware that is considered to be HCR malware, the most dangerous and complex ransomware up to now. Figure 4 shows the proposed architecture of our method.

1) Data Provider: This section of the architecture, in accordance with the class of features introduced, has the task of providing different data corresponding to the proposed approach properties. In this section, with the help of tools, the monitoring process of the evaluated program and its behavioral and dynamic analysis is done under the supervision of an expert.

2) Preprocessor: This section is designed with the goal to preprocess the data from different classes of extracted features that have high detail and volume, and, by deleting the data and additional details and reducing unrelated features, try to increase speed, integrity, and simplicity in the detection engine. Of course, what data is removed by the preprocessor is determined by the prior knowledge of the ransomware analyst.

3) Extracted Features: We found three classes of extractive features by analyzing the DNS traffic behavior of the samples:

3.1. *Random and Gibberish Characters in the Domains:* One of the obvious features of DGR requested domains is the randomness or meaningless nature of the letters. In other words, legitimate registered domains have a meaning, but domain-generation algorithms, because of the use of pseudo-random algorithms in the DGA module, produce the domains that are completely random and nonsense, but this feature can not distinguish between malicious domain and legitimate one by itself, since in some cases there are legitimate users who have registered legitimate domain that uses letters in a very different languages or in a completely obscure domain. On the other hand, some ransomware uses a number of domains to communicate with the C &C server, which lacks the inherent feature of gibberish of the letters, so this feature cannot detect all the desired ransomware and also yields to a lot of false positives. To solve this challenge, we'll use the other features next to it, which will be introduced in the following features. We show gibberish characters in domain generated by PWS-Zbot.gen.xd ransomware in Figure 3.

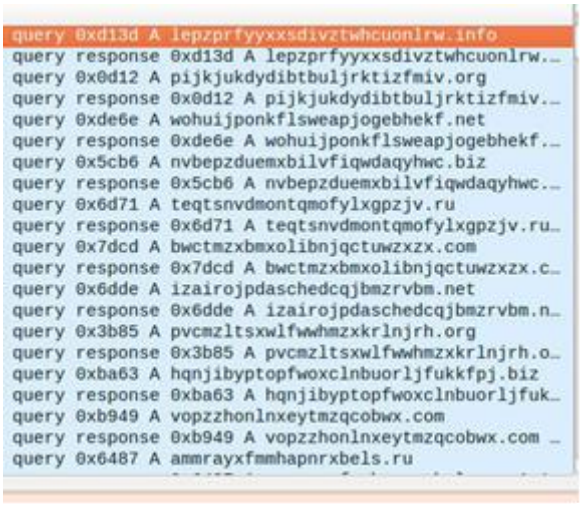

Figure 3. Gibberish Characters in Ransomware

3.2 The Frequency of Different Domains Generation: the feature extracted from the DNS traffic analyzing of ransomware is the frequency of generating different domains. DGR generates a large number of different domains over a period of time in order to prevent their domains from being detected by security mechanisms such as the black/white list, and since its designer is aware of the output of the algorithm, then his/her certain domains generated by the algorithm are ready to serve ransomware with the help of domain registration methods. If the malware is successfully installed and launched on the victim's system, after the network connection, the DGA module in the ransomware is triggered and communicates with the C & C server. However, the ransomware may not receive a response from C & C server after producing the requested domain, so after a short time, it will request another domain until it receives the response with one of the domains registered by the designer, So the frequency of generation of domains increases. As an example, the Cryptowall requests a different domain every 10 seconds once, and it generates 47 different domains within a time interval of 7 minutes and 22 seconds or the pws-zbot.gen.xd requests a different domain every one and a half seconds and it produces 1,000 different domains in a period of time.

3.3 The Replication of the Same Domains in a Time Interval: Since DGRs generate domains periodically if any of the domains do not succeed in communicating with the C & C server in a period of time, after a time lapse, they begin to reproduce the domain. As an example, Cryptowall re-generates a domain after a 48-minute and 21-second interruption. Therefore, in the next period, most of the production domains are repetitive compared to the domains produced in the previous period, and we can use this feature for detection. Of course, in some malware, different domains are produced at any time, and in each period, domains are produced in a different order, which again we can use the duplicate feature of the same domains to detect.

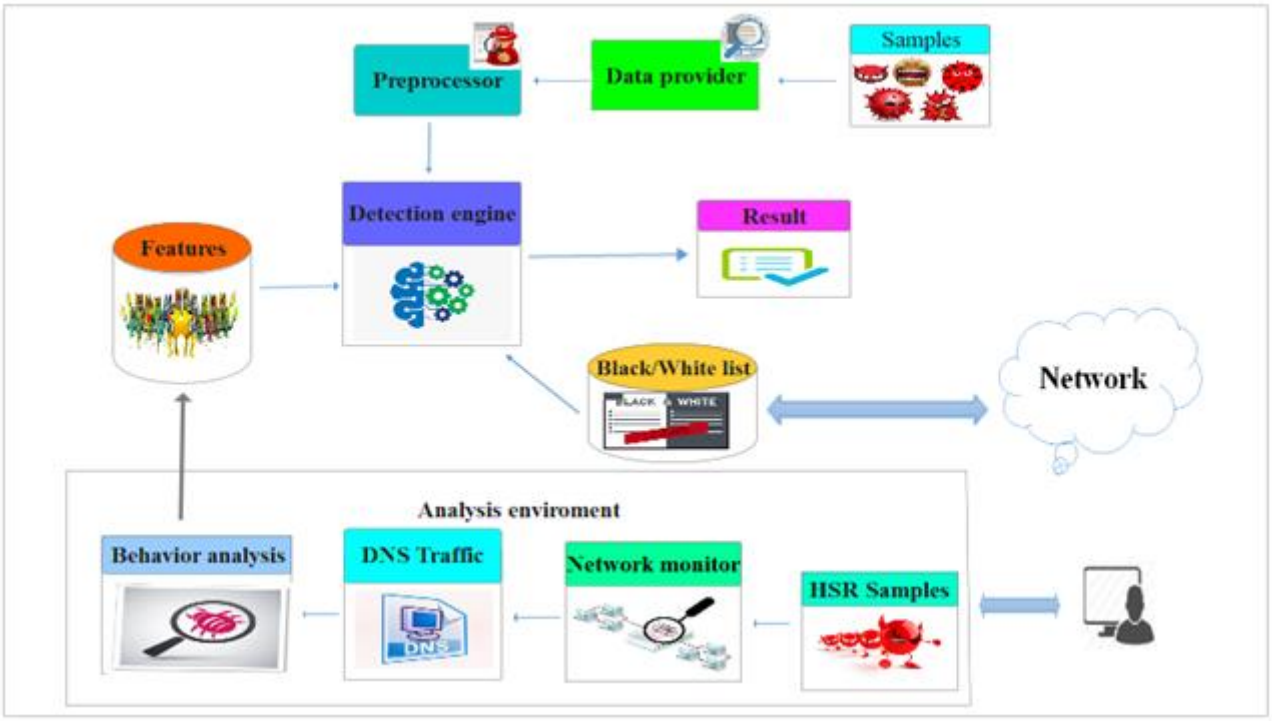

Figure 4. Architecture of Detection Approach

4) Detection Engine: The detection engine decides according to the extracted features. To detect the random and gibberish letters, the detection engine uses algorithms such as the Gibberish Detector which is based on Markov chain [5]. But as we mentioned before it is not enough because there are many legitimate domains that have this feature, which can cause the engine to detect a high false positive rate. On the other hand, the detection engine cannot detect all DGA-based malware because many of requested domains do not have the randomness of characters, and are quite similar to the regular domain. So, in order to solve this challenge, we will use two characteristics of " Frequency of Different Domains Generation,"" and " Repetition of Same Domains in a Time Interval ". Our detection engine interacts with the domain black/ whitelist module that is connected to the global network. This increases the power of the detection engine and also reduces the false positive rate.

5) Black/white List: When the ransomware runs on the victim's system and attempts to generate a domain, the victim system itself asks for a range of legitimate domains, so this module is used to enhance the detection efficiency and reduce the false positive rate of the engine.

6) Monitoring the Network Traffic: In order to analyze DNS traffic, we need to monitor and record network traffic, but we can not run the malware while we are connected to the Internet because it will infect other users on the network, so by using the tool. INetSIM, we try to create a simulated network that returned the response to the ransomware requests. We also used tools such as Wireshark to capture network traffic, especially DNS traffic.

IV. IMPLEMENTATION AND EVALUATION

In this section, we present and discuss the implementation and the results of the experiments. We will show how much the proposed approach is effective in detecting and preventing DGA-based ransomware (DGR) that encrypts the user's data with hybrid cryptosystem. In order to implement and test our proposed approach, we first needed a virtual machine to simulate the victim machine running instances of the malware.

So we created an ubuntu16.04 virtual machine in Virtual box to simulate the network and capture DNS traffic. Since if a ransomware connects to the Internet, it causes other network users to be infected. We needed to create a simulated network, which return any ransomware's request back to it. To do this, we use the INetSIM tool to simulate a network. We also configured the network so that any request from the victim machine connects to the simulated network and receive the answer from there.

We also used the Wireshark tool to monitor the network and capture the DNS traffic. Because some of the ransomwares are smart and detect the monitoring tools of victim's system, we run the Wireshark in a simulated network (Ubuntu 16.04). The DNS traffic output after the pre-processing steps is fed to detection engine, and the detection engine according to the extracted features and the domain black/whitelist modules recognize the ransomware Figure 5 shows the test environment.

There are a number of anti-ransomware tools, such as Hitman pro kickstart, HitmanPro, CryptoGuard, BitDefender AntiCryptoWall. All of which are signature-based and cannot detect new and unknown ransomware such as HSRs. Although the 2enFOX framework [2] is provided to detect HSRs, since this framework is not based on the characteristics of network traffic, then the proposed method cannot be compared with this existing detection system. But if we want to compare our proposed method with the only similar work in the field of DGA-based anti-ransomware, since [5] only considers the randomness or gibberish nature of the letters, it yields to the high false positive rate. As we

discussed before our approach uses other features to detect DGA-based malware more accurately. Another advantage of our approach is its simplicity and the use of multiple simple and efficient features.

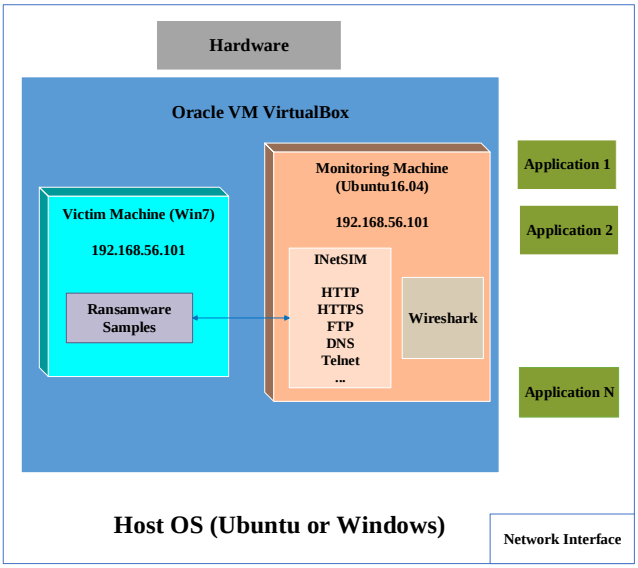

Figure 5. Test Environment

To demonstrate the effectiveness of our approach in detecting DGRs, we tested it on more than 20 samples. Our approach detects all HSR samples before completing the public key exchange process. The detection rate of 100% and the false positive and false negative rates of zero are results of our approach in detecting DGRs. All of these samples (collected from VirusShare.com, malwaretips.com, andBleepingComputer.com) have been selected because of their popularity and complexity. Of course, we predict that by increasing the number of HSR and non-HSR samples, the false positive rate may naturally increase, which we are going to consider it in future works. Naturally, the weakness of the proposed approach is that it does not detect non-DGR ransomware and if it is to be used to detect all malware, additional features should be added to our approach.
We present an overview of our tests in Table 1.

V. Conclusion

In this article, we presented a new approach to detect DGA-based ransomwares (DGR) that is one of the most dangerous ransomware (HSR) based on hybrid cryptosystem. The remarkable point of this new approach is to use the behavioral characteristics of DNS traffic. By detection the ransomware before it can evolve the public key exchange process and disconnect it from the command and control server we can prevent its progress.

The main advantages of our proposed approach can be summarized as follows: First, this approach is the first approach that focuses on DGRs. As our approach can detect the ransomware in its early phase of activity (before public key exchange and starting encryption of user data). It could be used as a proactive mechanism to prevent encryption of data by ransomware. Our tests have shown that this approach can successfully detect most DGRs. Second, it can also be used to detect other malwares such as botnets and other malware-based DGAs. Third, our proposed approach can be combined with other methods of detection that are not based on network traffic and detect a wide range of ransomwares.

Table 1. The Experimental Result of Our Approach

Name of ransomware	Type of Ransomware				HSR	DGR	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	×	×	✓	×	×	×	×
CryptDomaIQ	✓	×	×	×	✓	✓	✓
Pws-zbot.gen.xd	✓	×	×	×	✓	✓	✓
Winlock	✓	×	×	×	✓	✓	✓
Crypt.DB	✓	×	×	×	✓	✓	✓
Kryptic	✓	×	×	×	✓	✓	✓

References

[1] Kharraz, Amin, Sajjad Arshad, Collin Mulliner, William K. Robertson, and Engin Kirda. "UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware." In USENIX Security Symposium, pp. 757-772. 2016.

[2] Ahmadian, Mohammad Mehdi, and Hamid Reza Shahriari. "2entFOX: A framework for high survivable ransomwares detection." Information Security and Cryptology (ISCISC), 2016 13th International Iranian Society of Cryptology Conference on. IEEE, 2016, DOI: 10.1109/ISCISC.2016.7736455.

[3] Sgandurra, Daniele, Luis Muñoz-González, Rabih Mohsen, and Emil C. Lupu. "Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection." arXiv preprint arXiv:1609.03020 (2016).

[4] Krebs, Brian. "FBI: North Korea to blame for Sony hack." *Retrieved from KrebsOnSecurity: http://krebsonsecurity. com/2014/12/fbi-north-korea-to-blamefor-sony-hack* (2014).

[5] Ahmadian, Mohammad Mehdi, Hamid Reza Shahriari, and Seyed Mohammad Ghaffarian. "Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares." Information Security and Cryptology (ISCISC), 2015 12th International Iranian Society of Cryptology Conference on. IEEE, 2015, DOI: 10.1109/ISCISC.2015.7387902.

[6] Moore, Chris. "Detecting Ransomware with Honeypot Techniques." In Cybersecurity and Cyberforensics Conference (CCC), 2016, pp. 77-81. IEEE, 2016. DOI: 10.1109/CCC.2016.14, INSPEC Accession Number: 16397869.

[7] Gangwar, Keertika, Subhranshu Mohanty, and A. K. Mohapatra. "Analysis and Detection of Ransomware Through Its Delivery Methods." In *International Conference on Recent Developments in Science, Engineering and Technology*, pp. 353-362. Springer, Singapore, 2017.

[8] Everett, Cath. "Ransomware: to pay or not to pay?." Computer Fraud & Security (2016).Volume 2016, Issue 4, April 2016, Pages 8-12.DOI: 10.1016/S1361-3723(16)30036-7.

[9] McAfee Threats Report: First Quarter 2013, By McAfee Labs,Page 12,2013.

[10] McAfee Threats Report: Fourth Quarter 2013, By McAfee Labs,Page 16,2013.

[11] McAfee Threats Report: June 2016, By McAfee Labs,Page 46,2016.

[12] McAfee Threats Report: March 2018, By McAfee Labs,Page 8,2018.

[13] Sood, Aditya K. and Sherali Zeadally. "A Taxonomy of Domain-Generation Algorithms." IEEE Security & Privacy 14, no. 4 (2016): 46-53.

[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.