IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 2, 2020, accepted March 22, 2020, date of publication March 30, 2020, date of current version April 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984187

Open Repository for the Evaluation of Ransomware Detection Tools

EDUARDO BERRUETA [1], DANIEL MORATO [2], EDUARDO MAGAÑA [1], AND MIKEL IZAL [1]

[1] Department of Electrical, Electronic and Communications Engineering, Public University of Navarre at Arrosadia Campus, 31006 Pamplona, Spain
[2] Institute of Smart Cities, 31006 Pamplona, Spain

Corresponding author: Daniel Morato (daniel.morato@unavarra.es)

This work was supported by the Spanish MINECO under Project TEC2015-69417-C2-2-R and Project PID2019-104451RB-C22.

ABSTRACT Crypto-ransomware is a type of malware that encrypts user files, deletes the original data, and asks for ransom to recover the hijacked documents. Several articles have presented detection techniques for this type of malware; these techniques are applied before the ransomware encrypts files or during its action in an infected host. The evaluation of these proposals has always been accomplished using sets of ransomware samples that are prepared locally for the research article, without making the data available. Different studies use different sets of samples and different evaluation metrics, resulting in insufficient comparability. In this paper, we describe a public data repository containing the file access operations of more than 70 ransomware samples during the encryption of a large network shared directory. These data have already been used successfully in the evaluation of a network-based ransomware detection algorithm. Now, we are making these data available to the community and describing their details, how they were captured, and how they can be used in the evaluation and comparison of the results of most ransomware detection techniques.

INDEX TERMS Ransomware, open repository, traffic analysis.

I. INTRODUCTION

Ransomware is a type of malware that hijacks computers by locking them or by encrypting their files. The former is called lockscreen ransomware, while the latter is named crypto-ransomware or cryptoware. Cryptoware has become more important in recent years owing to its increased number of infections. In 2015 and 2016, the number of new ransomware samples detected increased rapidly, rising from 9296 to 32091 just in the last four months of 2016, according to Kaspersky Labs [1]. In the past two years (2018 and 2019), the number of new ransomware samples and the number of infections decreased (20% in 2018) [2]; however, this drop was caused by a change in the targets of the ransomware from the general population to specific companies. In 2019, Symantec reported that enterprise infections were up by 12% in 2018 and accounted for 81% of all ransomware infections in that year [2].

The importance of this type of malware encouraged the development of detection tools both in research and in cybersecurity enterprises. Among the research papers published in the past 4 or 5 years, the vast majority based detection on dynamic information obtained at the infected computer while the ransomware was running, such as the increase in file data entropy and the frequency of read and write operations or the system functions called. Other methods use information obtained from network traffic, such as the Domain Name System (DNS) requests [3] or general traffic statistics: Transmission Control Protocol (TCP) connections, Internet Protocol (IP) addresses, or TCP ports [4]. The tools developed for Android devices analyse the program binary by searching for specific function calls, text strings, or even some elements in the screen. The program is labelled as ransomware based on the combination of values of some of the parameters mentioned [5], [6] or, in some cases, by using them as input in a machine learning procedure [7], [8].

Although the detection accuracy of such tools is frequently reported in research papers, sometimes it is difficult to compare these results with each other because they are tested in different scenarios or because the evaluation parameters (the true positive rate, the false negative rate, or the detection accuracy, for example) are not the same for all the tools. Also some tools are tested only with one ransomware sample, so an accuracy of 100% does not mean the same as in other approaches that are tested using hundreds of samples [9].

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

The main problem in testing detection tools is not the lack of ransomware samples, but the lack of working samples, as they stop being functional when their control servers go down (when they are found out by cybersecurity agencies, for example). The ransomware samples must be run while the control servers are active, and all the activity information must be extracted to test any detection tool. There are some public websites where ransomware binaries are uploaded: malware-traffic-analysis [10] or hybrid-analysis [11], for example. These websites offer binaries of different types of malware (including ransomware), and they analyse the binary and some other aspects of the malware, such as the infection vector or the DNS queries. However, none of these repositories provides the information needed for detection tools based on the dynamic behaviour of the malware.

This paper describes a public repository, available at IEEE DataPort and our local servers,[1] containing dynamic information obtained from more than 70 ransomware samples in action. We have been collecting these data since 2015 and we keep updating the repository. The dynamic information is obtained by running each binary in a scenario with a shared directory between a server and a client. The ransomware encrypts all the files in this directory, and we capture the network traffic and store it in a trace file. The file also contains other traffic that the client generates during the encryption — for example, DNS requests and connections to Command-and-Control (C&C) servers. We have extracted all the input/output (I/O) operations from the file-sharing protocol, and we offer independent files containing this information, which can be used in detection methods based on file access operations.

The main contributions of this paper are:

- We offer an updated ransomware sample repository with the result of running more than 70 samples from 31 different ransomware strains. These samples contain more than 1 TB of data in more than 200 traces. The behaviour of the malware can be analysed, as each samples shows the ransomware in action.
- New tools can be tested with old samples that are now deactivated for several reasons but could be reactivated in the future.
- Tools based on analysing file access operations can be tested, as the repository offers all the I/O operations executed by the ransomware samples.
- We prove that the majority of existing tools can be tested with this repository.
- The repository is kept updated and contains ransomware binaries since 2015.

The paper is structured as follows: Section II presents a historical review of ransomware strain appearances and the most important outbreaks. In Section III, we explain the scenario and methodology used for capturing the traces of ransomware activity. Then, in Section IV, we present a brief description of the files in the repository and an analysis of two samples, and we extract important ransomware characteristics from the files in the repository as an example of what can be done with these data. In Section V, we explain why this repository is useful for ransomware research. Finally, in Section VI, we present the conclusions of this paper.

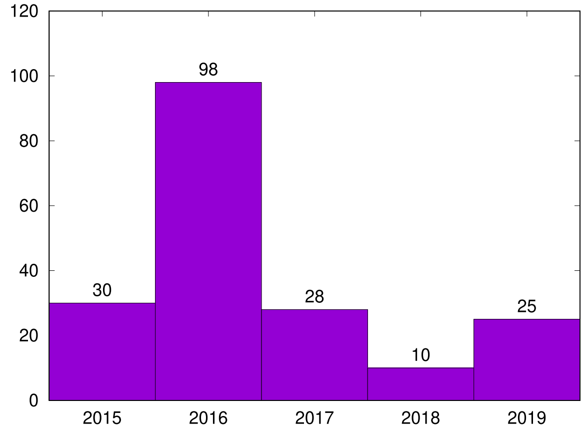

FIGURE 1. Number of new ransomware families appearing from 2015 to 2019 [2], [14].

II. HISTORY AND CLASSIFICATION OF RANSOMWARE

The first documented ransomware (the PC Cyborg locker ransomware) appeared in 1989, but it was not until 2013 that this type of malware was considered an important problem for home users and enterprises, with the appearances of the first crypto-ransomware (CryptoLocker [12]).

In 2016, Europol declared that cryptoware had become ‘‘the most prominent malware threat [...] for citizens and enterprises alike’’ [13]. Since then, the strategy used by ransomware developers has changed. Enterprise targets are a more profitable objective than individual users; therefore, hackers have focused on them. Consequently, although the total number of infections and the appearance of new ransomware strains has dropped, in 2019 ransomware was still considered as ‘‘the top cyber threat faced by European cybercrime investigator’’ by Europol [14].

Figure 1 shows the number of new ransomware strains from 2015 to 2019. In 2016, this number increased to 98, while in 2018, there were only 10 new ransomware strains. This has not caused a drop in the attackers' benefits, as they have changed the target of these attacks from users to enterprises (in 2018, the number of enterprise infections was four times higher than in the general population) [2].

In addition to the appearance of new ransomware strains, each strain has multiple variants or versions with different behaviours, C&C servers, and names. Sometimes, the ransomware authors change the behaviour of the malware to make detection more difficult or because the filtration of private keys enables file decryption.

Figure 2 shows the number of new ransomware samples detected between the end of 2016 and the beginning of 2019 [15], [16]. Although there was an increase in the number of new ransomware strains in 2016, it was not

[1] http://dataset.tlm.unavarra.es/ransomware

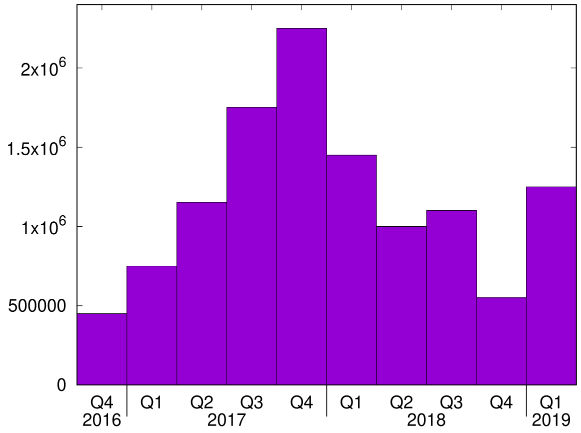

FIGURE 2. Number of new ransomware samples appearing from 2016 to 2019 [15], [16].

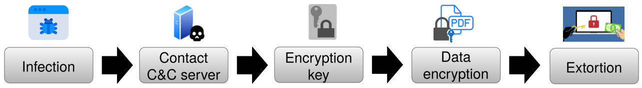

FIGURE 3. Ransomware infection steps.

reflected significantly in the number of samples. However, in 2017, the number of new ransomware samples increased more then three-fold compared with the last quarter of 2016. This increase could be caused by new strategies in malware development in the samples discovered in 2017.

In these two years (2016 and 2017) the number of infections increased too. The most important ransomware in terms of infections in 2016 was TeslaCrypt [17], which reached 90% of all ransomware infections. In May 2016, its developers published the decryption key and the ransomware disappeared. In terms of profits, Locky was the most profitable ransomware in 2016, with more than $7 million [18].

In 2017, there were important infections in enterprises, like the WannaCry attack (May 2017) [19], which hit some important enterprises, such as the Spanish telecommunications company Telefónica. This ransomware variant was still active (with some updates) in 2018, hitting some important hospitals in the United Kingdom and costing near £92 million to the British National Health Service [20]. In that year, the case of Cerber was also very important, earning $6.9 million [18]. There were more than five different versions of this ransomware, defined by the authors themselves in the ransom note left in the infected machine.

Other ransomware variants, such as GandCrab, Ryuk, and BitPaymer, hit some important enterprises worldwide during 2018 and 2019 [21]–[23].

In general, the behaviour of all cryptoware strains is similar. Figure 3 presents the five steps of the infection process. We briefly name and describe them below:

1) Infection: The attack vectors are the same as in other types of malware. The most common method is sending e-mail-attached files that are executed by the user (39.4% of infections [9]). In other cases, the binaries are downloaded from infected web pages. Finally, some ransomware strains implement a worm-like behaviour to propagate through a Local Area Network.
2) Contact C&C servers: Some ransomware strains contact a C&C server to obtain or store the encryption key. This server can be located using statically configured IP addresses, static DNS names, or dynamically generated DNS names. This step can take place after the data encryption if the ransomware works offline, and the C&C server is contacted in the last step, only to store the decryption key.
3) Encryption key management: The key can be obtained from C&C servers or be generated locally and then stored on the server. The ransomware encrypts locally generated keys with one obtained from the C&C server.
4) Data encryption: This is the main task of cryptoware. The ransomware encrypts and deletes user files. It usually also affects files and volumes mounted using a network file-sharing protocol.
5) Extortion: As the last step, the malware requests the payment of ransom to decrypt the files. It can place some files in different formats (plain text, html, images) in each directory, explaining how to pay the ransom. It usually sets a deadline and threatens to delete some of the user's files every hour.

Different ransomware strains carry out each step in different ways. The detection tools must be aware of these differences, as they aim to detect all ransomware strains, if possible, before the fourth step (data encryption), to prevent the loss of files.

The detection in the first step of the infection process is accomplished using firewalls, anti-malware software installed in the user machines, or educating the users to neither download nor execute files with doubtful origins.

In the second step, the detection must be done by analysing the users' traffic. Firewalls can block traffic to some blacklisted IP addresses or some DNS requests. Until the destinations can be blacklisted, firewalls do not block the requests. These are the zero-days attacks. Some ransomware strains, such as Locky, use a Domain Generation Algorithm (DGA) for its C&C server address. This makes identifying these servers more difficult. To address this issue, there are some tools that try to detect names generated by a DGA [24], [25].

To detect the malware in the third step, some tools analyse the system function calls. The frequent use of a set of functions can indicate that a process is encrypting files. Cryptographic keys in process memory can also be recognised owing to their structure. Monitoring software can inspect the process memory, search for keys, and alert the user. The success rate of these tools depends on how the ransomware manages cryptographic keys and what kind of cryptographic algorithms they use (asymmetric, symmetric, or hybrid). Detection in the first and second phases of ransomware action does not require the tools to be installed in the machine; however, in this third step, the tools must analyse some parameters that are available only in the infected machine.

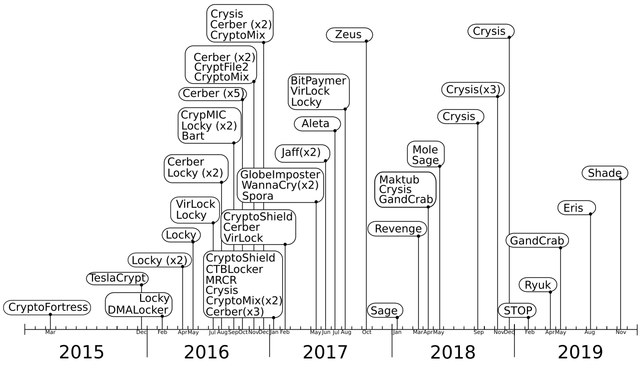

FIGURE 4. Timeline of collected samples.

The fourth step is the last opportunity for detection before losing data. The tools that focus on this phase usually analyse some parameters obtained from user files: change of entropy in the content, modification of magic bytes, read and write frequency, or directories in which the malware encrypts files are some examples. These parameters are more useful in some ransomware strains than in others because special malware behaviour can be determinant for detection. Some examples of ransomware behaviour that can evade detection by these tools are:

- Some Cerber samples do not encrypt the first bytes of the files (magic bytes) [26]; thus, the tools that analyse the change of these bytes [27] will not be able to detect them.
- Some ransomware strains, such as CryptFile2, do not delete the files or change the file extension. They just overwrite the file with the encrypted data. Some tools base detection primarily on these two aspects [6], [28].
- The read and write frequency is commonly used by these tools to detect the malware [29], [30]. Every ransomware must read and write the files to encrypt them, which they can do slowly to avoid being detected by these tools.

Finally, ransomware detection in the fifth step is useless, as the data have already been encrypted. The data could be recovered from a backup, but that would be recovery, not detection.

The repository that we present in this paper collects ransomware samples from the most important strains since 2015 (and it is regularly updated). In 2016 and 2017, there was an increase of ransomware sample appearances, which is reflected in the number of samples that we collected. Figure 4 shows a timeline of the sample collection.

In the next sections, we analyse in depth the content and structure of the repository, as well as its possible uses for the research community.

III. SCENARIO AND METHODOLOGY

This section explains in detail the scenario in which the samples were executed. In recent years, new ransomware strains have focused on corporations, attacking desktop computers of large companies. To run the ransomware samples, we selected a scenario commonly used in enterprise networks, in which user documents are centralised in file servers. This facilitates document sharing and economic backups. The file-sharing protocol used is Server Message Block (SMB), which is the default and most common option in a Microsoft Windows environment.

The content of the shared directory on the server must be set carefully to simulate a real directory. File sizes and file distribution in a tree structure could alter ransomware behaviour and detection rate.

The basic parameters defining the scenario for each run of a ransomware sample are:

1) Operating system: Some ransomware samples were run in a Windows 7 environment, while others were run in both Windows 7 and Windows 10 installations. Windows 7 used version 2 of the SMB protocol, while Windows 10 used SMB version 3.
2) Network speed: By default, the network speed is set to the maximum allowed (10 Gb/s). It is possible to control the network speed by placing a router between the client and the server. This was used to simulate a ransomware strain that operates in a slower fashion.
3) Fileset: Random filesets were created. The file population of the directory can affect the ransomware behaviour because the fileset is encrypted by the ransomware sample. In Section III-B, this fileset generation is explained in depth.

A. NETWORK SCENARIO

All ransomware samples are run in the user's machine, where the network volume is mounted. All the ransomware that we found encrypt the shared directory. We captured the traffic between the client and the server, which contains all the file operations (open, read, write, rename, delete, or close operations over the files in the shared directory).

Although capturing the I/O calls locally in the client machine would be easier, this would cause an increase in CPU load, which could alter the measurement. A passive network traffic capture reduces interference during ransomware action. By capturing the file access traffic between the client and the server, we created a more general repository, useful for testing tools based on network traffic and on I/O operations, that we extracted from the trace using a custom traffic analysis tool.

Figure 5 shows a network scenario similar to the networks used in a corporate environment, in which the samples were executed. The infected host accesses the files from a machine acting as a Network Attached Storage (NAS) *filer* or server.

In production scenarios, the link from the user to the network is usually at most Gigabit Ethernet, while the links from the NAS filer are at least 10 GbE. The bottleneck is created by the disk access latency, which is several times larger than the network latency or the ransomware encryption time; therefore, a networking scenario without speed limitations or losses is not affected by the networking component as much

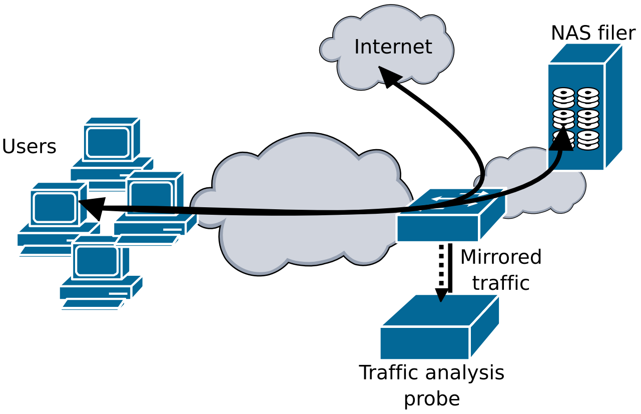

FIGURE 5. Network scenario.

as by the effect of the disk access speed at the server, which is similar to running the ransomware locally in the client.

There are two traffic flows suitable for capture in this scenario. The first one is the traffic between the infected host and the NAS filer. This is the most important flow, and we provide this traffic for all the ransomware samples that we ran during at least 5 years. The second flow contains the traffic between the infected host and servers in the public Internet. This traffic contains all the DNS requests and the actions taken by the ransomware to contact C&C servers. Not all ransomware strains require Internet connectivity (Cerber, CTBLocker, and Sage do not require C&C servers). We included a traffic capture of the second flow for some of the samples because certain ransomware detection techniques are based on this traffic and can also be tested with these input data.

Both user and server hosts are Windows machines running virtualised in a VirtualBox environment with a Network Address Translation configuration in a host with an Intel Core 2 Duo CPU E6750 of 2.66 GHz (one CPU core per virtual machine). Windows 7 uses version 2 of SMB by default for the file-sharing traffic, while Windows 10 (and sometimes Windows 8) uses version 3 of SMB, which is an encrypted protocol. Some binaries were run in both Windows 7 and Windows 10 scenarios. The scenario using Windows 10 is similar to the Windows 7 one, but the change in operating system can result in not only different versions of SMB protocol, but also other differences in the TCP/IP stack implementation. Both traffic traces are available at the repository.

An important advantage of running the ransomware samples in a network scenario is the possibility of modifying the network speed. This could be used to simulate a slow ransomware encryption of files or a slow disk. When running the ransomware samples locally, the speed of the encryption is fully determined by the machine in which the samples are run. We provide samples in which the network speed changes from 1 to 100 Mb/s to simulate such slower ransomware.

The content of the shared directory is also a fundamental part of the scenario because it could impose conditions on the results of some detection tools. The file sizes must represent the real population of files, spread along directories. A simpler file structure — for example, equal-size files in a single directory — could introduce some bias in certain detection tools. For example, if all the files are very large, the time between two files encryption will be long, which would not be representative of a user who creates mainly small files. We created the directories using a tool developed based on a study of the content of real user hard disks [31]. Some samples are run using more than one shared directory, providing samples with different populations of file systems. In Section IV-B, we give examples of how different ransomware strains follow a different order when encrypting files, so the directory (file types, file sizes, or subdirectories) can influence the actions taken by the ransomware and therefore the effectiveness of different detection techniques.

We automated the process of powering on the user and server machines, copying the ransomware binary to the user's host and executing it. After contact with the C&C (when it is necessary), the ransomware starts the encryption of the user files, including the server shared directory, which is mounted on the user machine. Because the size of the traffic trace is monitored during the process, we know when the ransomware finishes the encryption, as the file maintains the same size for some minutes (30 min, for example). When this occurs, the machines are automatically powered off and their images are restored. During the entire process, the traffic flows described earlier are stored.

Most ransomware detection algorithms take the I/O operations as input information. We extracted this information from the traffic trace files, providing an easy-to-use file format describing all these operations. Tools such as TShark or Wireshark were considered for the task; however, not only are they severe RAM and CPU hogs, but we also detected missing messages in the results of their analysis. For example, when the SMB2 header is fragmented between two TCP segments and in some cases of TCP disorders, Wireshark cannot follow the stream of SMB commands [32]. We developed our own tool to process the traces and extract the I/O operations. For version 3 of the SMB protocol, it is not possible to extract these I/O operations, as the SMB data are encrypted.

B. DIRECTORIES SHARED BY THE SERVER

Except for some old samples, all ransomware binaries were executed in a scenario with a shared directory created following the parameters described by N. Agrawal *et al.* in [31]. Some samples were also run in smaller directories with different characteristics (see Table 1) to create a more varied repository. These smaller directories have more file types and, in general, smaller files. The traces obtained from these samples can be used to analyse how different directories affect ransomware behaviour. For example, different file types or very small files cause some ransomware strains to not encrypt all the documents.

For the construction of the *5GB* directory, we used the software (*Impressions*) described in [31]. The authors used snapshots of file-system metadata collected over a five-year period representing over 60000 Windows PC file systems

TABLE 1. Directories information.

	Size	Largest file	File types	Generation	Number of files	Number of samples
Small	73 MB	1 MB	txt, doc, xls, jpg, mp3, wmv	Manually	768	22
Medium	1.88 GB	25.6 MB	txt, doc, xls, jpg, mp3, wmv	Manually	1712	4
5GB	4.5 GB	838 MB	pdf	[31]	5034	67

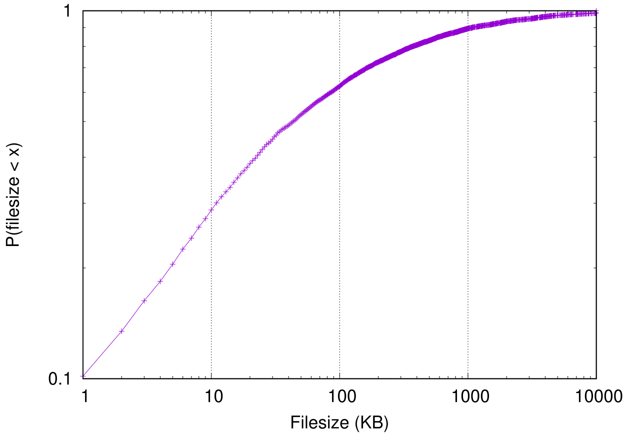

FIGURE 6. Cumulative distribution of file sizes in the *5GB* dataset.

in a large corporation. These snapshots were used to study distributions and temporal changes in file size, directory size, namespace structure, and other characteristics. Once the tool was created, to ensure the accuracy of generated images, the authors compared the generated distributions with the ones obtained from the dataset.

Using the same parameters but a different seed for the random number generation, we can generate statistically similar directories. We can also change the random variable parameters and create directories with different characteristics and rerun old or new binaries if we notice that other characteristics of files or directories are interesting.

To create directory trees, *Impressions* uses a Monte Carlo simulation and the size of the file is sampled from a hybrid distribution, the body of which is approximated by a lognormal distribution ($\alpha_1 = 0.76$, $\mu = 9.48$, and $\sigma = 2.46$), with a Pareto tail distribution ($k = 0.91$, $\chi_m = 512\ MB$). These parameters were obtained by fitting the respective curves to file sizes obtained from the file system dataset.

The distribution of file sizes is important for the analysis of ransomware behaviour because some ransomware strains do not encrypt files smaller or larger than a certain size. Additionally, for a realistic directory, the distribution of file sizes (shown in Figure 6) is important because it is used to determine the ransomware infection time.

The structure of the directory is also important because not all the ransomware strains iterate through it in the same manner. Some of them iterate alphabetically, others iterate in size order, and others randomly. If we placed all files in the same directory without subdirectories, we would miss these differences between the ransomware strains. Examples of the impact of these differences are presented in Section IV-B. Figure 7 shows the directory tree generated by Impressions in the *5GB* dataset, where the directory names are numbers.

IV. TRAFFIC TRACES, I/O OPERATIONS, AND NETWORK REQUESTS

This section describes the network traffic traces and I/O operations files in the repository. First, we present the different ransomware strains and the available samples from each strain. We present two examples of sample analysis using the I/O operations file of an execution of Cerber and another one of Locky. These examples show four possible parameters that can be analysed from the datasets that are present in this repository, but it is possible to analyse more parameters if they affect the detection strategy.

In the repository web page, we have included data plots for some of these parameters to enable their comparison. For brevity, we show the data from only these two samples.

A. SAMPLE SUMMARY

During the past 5 years, we have collected more than 70 ransomware samples from 31 different strains. The binaries were downloaded from hybrid-analysis [11] and malware-traffic-analysis [10]. We consider that two samples are different if the binary is different, and that two samples are from different families if the website from which they were downloaded considers them different. Sometimes, it is difficult to confirm whether a sample is an update of an old strain or it is a new one, and sometimes this depends on the opinions expressed at the consulted website.

Table 2 shows a summary of the ransomware samples collected. In each row, one sample from one strain is presented. For some strains, there is more than one ransomware sample; in these cases, there is a *(+)* sign following the strain name. For the samples executed with more than one shared directory, the data presented in the table were extracted using the *5GB* directory. The complete table and links to the samples are available at our public repository.

The *Directory* column in Table 2 indicates the type of shared directory that the ransomware encrypted when it ran. The *Traffic trace size* column indicates the file size of the network traffic trace captured during ransomware execution. This is important because not all ransomware strains encrypt the files in the same manner and the traffic traces do not have the same size, although they are executed with the same shared directory. Some ransomware samples (such as Ryuk or Spora) encrypt only a certain percentage of each file. Others do not encrypt all files: Cerber does not encrypt files smaller than 3 kB, BitPaymer encrypts only some files,

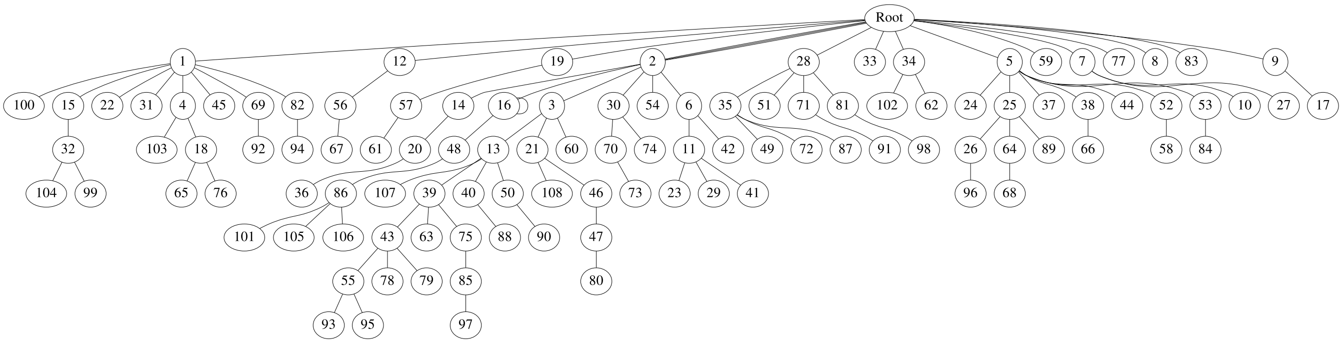

FIGURE 7. Tree of the directory generated. Each circle represents one directory, whose name is a number.

TABLE 2. Ransomware families in the repository. Characteristics.

Strain	Date of appearance	Directory	Traffic trace size	Time (min)	Packets (M)	MB R&W	Open / R&W ops (K)	DNS requests	TCP conn.
Aleta	2017	*5GB*	6.8GB	121	17.29	5733	97 / 123	No	0
Bart	2016	*5GB & small*	7GB&175MB	26.7	11.3	6641	30.4 / 293.3	No	2
BitPaymer	2019	*5GB*	7.7GB	121.36	11.74	7277	24.9 / 124.5	No	0
CTBLocker	2015	*5GB & small*	5.4GB&682MB	19.2	8.2	5183	14.9 / 91.4	No	2
Cerber (+)	2016	*5GB & small*	1.6GB&323MB	9.2	63.2	1479	6.7 / 29.3	Yes	4
CrypMIC	2016	*5GB & small*	11GB&173MB	41.43	15.82	10176	10.2 / 170.7	No	2
CryptFile2	2016	*5GB & small*	7.9GB&166MB	30.53	11.92	7536	15.5 / 127.2	No	2
CryptoFortress	2015	*5GB*	5.5GB	24.37	41.27	5114	19.7 / 97.6	No	1
CryptoMix (+)	2016	*5GB & small*	7.9GB&318MB	29.13	11.88	7536	15.5 / 127.2	No	2
CryptoShield (+)	2016	*5GB*	8.3GB	23.7	23.7	7972	20.1 / 134	No	1
Crysis (+)	2016	*5GB*	4.5GB	28.5	150.6	4730	185.5 / 113.1	No	0
DMALocker	2016	*5GB*	20GB	67	28.3	18466	15.3 / 319.9	No	2
Eris	2019	*5GB*	5.6GB	23.2	8.5	5381	24 / 101.6	Yes	4
GandCrab (+)	2018	*5GB*	9.8GB	28.2	15	9326	60.2 / 171.4	Yes	3
GlobeImposter	2017	*5GB*	5.3GB	33.5	10.4	4595	15.3 / 877	No	2
Jaff (+)	2017	*5GB*	9.8GB	31.3	14.6	9408	12 / 158.2	Yes	2
Locky (+)	2016	*5GB & small*	9.5GB&279MB	43.6	28.58	9141	16.9 / 153	Yes	4
MRCR	2017	*5GB*	6GB	1038	9.1	5693	36.1 / 97.7	No	2
Maktub	2016	*5GB*	504MB	3.5	0.8	464	8.2 / 13.4	No	2
Mole	2017	*5GB*	7.8GB	23.3	11.7	7435	20.4 / 156.7	No	3
Revenge	2017	*small*	4GB	21.5	6	3678	7.4 / 72.6	No	0
Ryuk	2019	*5GB*	4.5GB	10.2	7.4	2141	6.2 / 41.1	No	2
STOP	2019	*5GB*	5.1GB	10.8	4.3	2693	10.9 / 46	No	3
Sage (+)	2018	*5GB*	11GB	25	17.2	11669	64.2 / 200.5	Yes	1
Sodinokibi	2019	*5GB*	2.6GB	33	2	2468	16 / 53.8	No	8
Spora	2017	*5GB*	969MB	8.9	35.4	809	21.9 / 31.7	No	13
Shade	2019	*5GB*	5.2GB	18.5	7.8	5007	9.2 / 96.2	Yes	7
TeslaCrypt	2015	*5GB*	6.9GB	24	10.2	6595	15 / 111	Yes	9
VirLock (+)	2014	*5GB*	4GB	91.3	15.6	9769	45.8 / 162.2	No	2
WannaCry (+)	2017	*5GB*	11GB	31.2	16.2	10300	34.9 / 200.5	Yes	3
Zeus	2017	*5GB*	4.6GB	12.5	118.1	4374	23.7 / 99.8	No	2

and CryptoShield does not encrypt large files. Samples of the Crysis strain encrypt system files and cause the machine to crash. This is the reason why the size of Crysis samples are smaller than others.

The last two columns are related to the traffic from the user to the general Internet. One column indicates whether the ransomware requested any DNS resolution (we ignored any name resolution related to Microsoft system domains, although they are included in the trace). Some ransomware strains such as Locky generate the C&C server domain names using a DGA. This is why the Locky samples contain more DNS requests than other samples. Some samples do not send any DNS requests because they have the C&C server IP hardcoded and they do not need to request the DNS, or because they do not need to contact the C&C server before the encryption of the files (or ever).

The last column indicates the number of TCP connections that the user establishes during the encryption process. The samples that do not establish any TCP connection do not contact their C&C server for the encryption, sometimes because they generate the keys locally or because they have them hardcoded. Some of them contact their server at the end of the encryption to request the key with which they will encrypt the locally generated one.

Some ransomware strains (such as Cerber) scan the network looking for other active machines with certain vulnerable services or to perform a DDoS attack [33]. This is not shown in the table, but it can be observed by analysing the traffic trace.

The columns *Packets (M)*, *MB R&W* and *Open/R&W ops (K)* describe the number (millions) of packets in each trace, the aggregated amount of megabytes moved in read and write operations and both the number of files opened and the number of read and write operations. This information is related to the user's I/O calls accessing the files in the server. It can be useful for testing some tools that detect the ransomware based on local I/O operations. The number of these operations is not the same or even similar for each ransomware strain, as not all ransomware strains encrypt the files in the same manner. Regarding open operations, the differences between the ransomware samples are even more notable, as some ransomware strains open each file just once to encrypt it, whereas others open the files more times (one for reading, another for writing, another for renaming, etc.).

The information in these I/O-related columns is extracted from the I/O operations file, which has been extracted from the traffic trace. The values in the table refer only to successful operations, as the status of each operation is in the I/O operations file. Other examples of information that can be extracted from the I/O operations file are the number of deletions and rename operations, the names of the encrypted files, and the file extensions.

In Section IV-B, an in-depth description of two ransomware samples from different strains is presented and some of the parameters extracted from the I/O operations file are analysed in detail.

B. SAMPLE ANALYSIS

Two of the most important ransomware strains that appeared in 2016 and 2017 are Locky and Cerber [17]. Both have some specific characteristics that we analyse below. In terms of network traffic, Locky's C&C server is located in a domain generated by a DGA, whose name resolution requests can be found in the traffic traces. Cerber does not generate its C&C domain name using a DGA, but it scans the network by sending UDP packets to port 6892 to perform a DDoS attack [33], which can be seen in the network trace.

In terms of the encryption process, the read and written bytes, the time between open operations, the number of deletions, and the file sizes are present in the I/O operations files.

1) BYTES READ OR WRITTEN

Figures 8 and 9 show the number of megabytes read and written per minute by the two ransomware samples (Locky and Cerber respectively). Both samples take more or less the same time to encrypt the directory. However, the distribution of megabytes per minute is not the same. During the first 5-6 min, Locky encrypts few bytes (approximately three times fewer than Cerber), and the rest of the time it increases its speed two or three times, while Cerber maintains the encryption rate constant during the entire process.

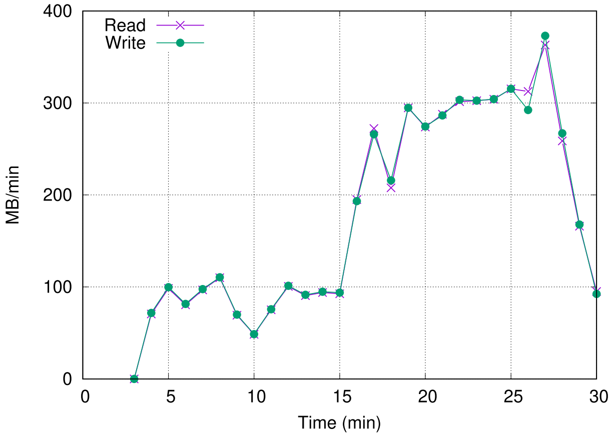

FIGURE 8. Read and written bytes per minute by Locky ransomware.

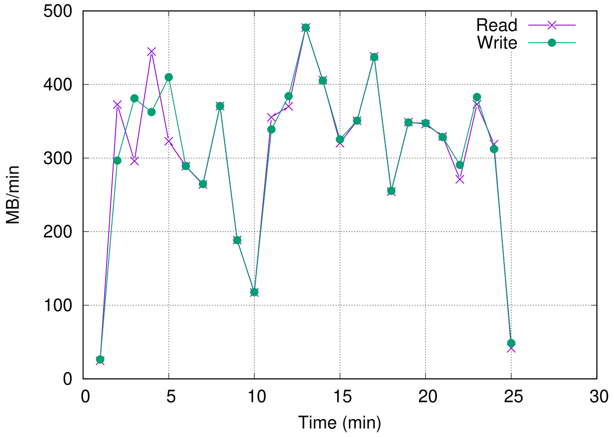

FIGURE 9. Read and written bytes per minute by Cerber ransomware.

Locky's speed variation can be caused by different factors but, as we describe in the next sections, the main one is the size variation of the files that it encrypts. Locky starts with the smallest files and when the files are small, the encryption speed falls. Cerber follows an alphabetic order for the encryption; thus, its speed remains approximately constant.

2) TIME BETWEEN OPENING OPERATIONS

We focus on the time between open operations for the two ransomware samples (Figure 10). The vast majority of these times (98–99%) are below 1 s, but Cerber shows smaller values. Locky opens files very fast, with 50% of the open operations separated by less than one millisecond, while for Cerber, only 20% of the operations are so close together. This is because Locky opens each file more than once, even before it reads any data. These open operations cause separation times below 1 ms.

Long times (above 30 ms) between open operations are more probable for Locky than for Cerber because Cerber performs faster encryption because it does not encrypt files individually. Cerber starts the encryption of the next file before

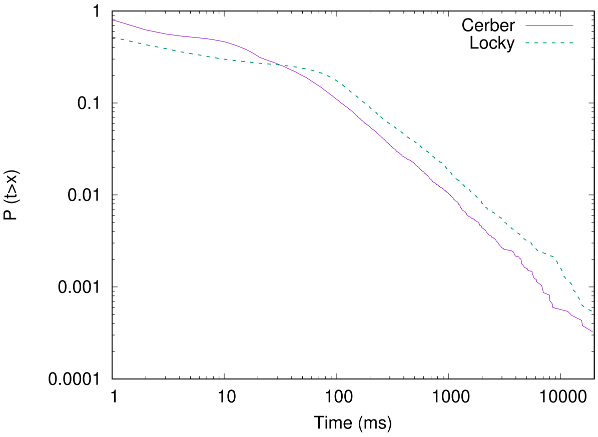

FIGURE 10. Read and written bytes per minute by Locky and Cerber.

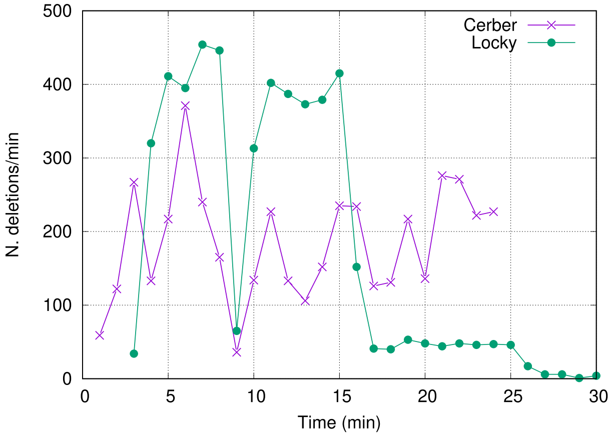

FIGURE 11. Delete operations per minute by Locky and Cerber.

it ends the previous one. This fact is reflected in the total duration of the directory encryption (see Figures 8 and 9).

These two different behaviours have been analysed using the information extracted from the traffic traces and I/O operations files. More ransomware strain behaviours can be analysed using the files available in this repository.

3) NUMBER OF DELETE OPERATIONS

The number of delete operations (Figure 11) shows the same behaviour as for the read and write operations (Figures 8 and 9). Although Cerber maintains the rate of delete operations as more or less constant during the encryption phase, Locky deletes more files at the beginning of this process. Locky starts with the encryption of small files; therefore, the number of deletions is higher at the beginning and then falls for larger files.

The ordering of files in the encryption process is important because the time that a detection tool takes to detect a ransomware sample has different consequences for different ransomware strains. In the case of Locky, delaying the detection by 2 min means the loss of more than 500 files; in the case of Cerber, fewer than 300 files would be lost. However, the opposite happens in terms of byte loss because Cerber encrypts more bytes at the beginning of the process compared with Locky.

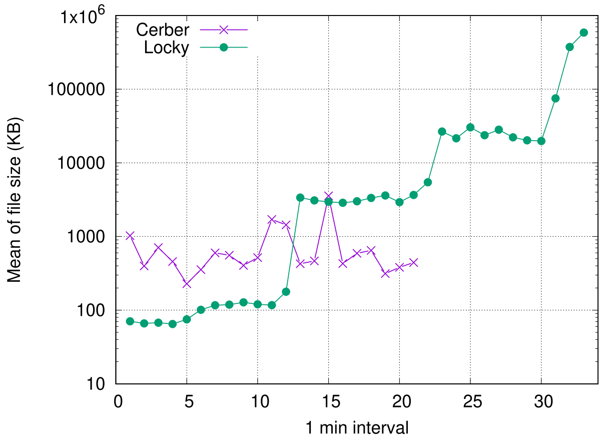

FIGURE 12. Size of files opened. Averages in 60-s intervals.

4) FILE SIZES

The size of the files encrypted by the two ransomware samples follows different ordering. Figures 12 shows the average file sizes in 1-min intervals. While Locky starts the encryption with the smallest files, Cerber follows an alphabetical order. This fact can be important for some detection tools because some of them base detection on the number of delete operations in a specific time period exceeding a threshold. It is easier for the ransomware to encrypt a large number of small files in a short interval than to encrypt random files, which could be larger.

The file sizes have been obtained from the opening operations in the I/O operations files. Only one opening operation for each file has been considered (Locky opens each file more than once), and the 1-min interval has been chosen to assess the differences between the two ransomwares strains.

V. POSSIBLE USES FOR THE DATA IN THE REPOSITORY

The purpose of this repository is to be used for testing new and old ransomware detection tools or proposals. We collected network traffic because it helps in obtaining most of the information required by the detection tools. We found at least seven detection tools that cannot operate without accessing the network traffic from the ransomware, because it is required by the detection process (see [9]). Moreover, most detection tools use information from the file I/O operations, which we can extract from network traffic because the attacked files are placed in a network shared directory. We have developed specialised software for extracting the I/O operations from the traffic trace, making our repository useful for most of the existing detection tools.

Table 3 presents 29 detection tools and the parameters used for detection. The last column indicates whether the tool can be tested using the data in the repository.

Some tools base detection only on network parameters (column *Network* in Table 3), such as DNS requests [3], DNS

TABLE 3. Detection tools and parameters used.

Reference	Data access information	Metadata access information	Network	Function calls	Local static	Testability
M. M. Ahmadian et al. [24]			Yes			Full
Paik et al. [28]	Yes	Yes				Full
N. Scaife et al. [27]	Yes	Yes				Full
K. Cabaj et al. [3]			Yes			Full
C. Moore [34]		Yes				Full
D. Sgandurra et al. [5]	Yes	Yes		Yes		Partial
M. M. Ahmadian et al. [35]	Yes	Yes		Yes		Partial
F. Mbol et al. [36]	Yes					Full
M. Shukla et al. [29]	Yes	Yes				Full
A. Continella et al. [37]	Yes	Yes		Yes		Partial
Y. Feng et al. [38]		Yes				Full
S. Chadha et al. [25]			Yes			Full
E. Kirda [39]	Yes	Yes				Full
R. Vinayakumar et al. [40]				Yes		No
Z. Chen et al. [41]				Yes		No
A. Kharraz et al. [6]	Yes	Yes				Full
T. Lu et al. [30]		Yes		Yes		Partial
M. M. Hasan et al. [7]		Yes	Yes	Yes	Yes	Partial
JA. Gómez-Hernández et al. [42]		Yes				Full
S. S. Khashif et al. [8]	Yes	Yes		Yes	Yes	Partial
M. Alaam et al. [43]				Yes		No
F. Quinkert et al. [44]			Yes			Full
B. A. S Al-rimy et al. [45]				Yes		No
H. Zhang et al. [46]		Yes		Yes		Partial
R. Moussaileb et al. [47]		Yes				Full
S. Mehnaz et al. [48]	Yes	Yes		Yes		Partial
D. Morato et al. [49]			Yes			Full
A. O. Almashhadani et al. [4]			Yes			Full
K. Lee et al. [50]	Yes					Full

domain names generated by DGA [24], general traffic [4], and SMB traffic [49]. The traffic traces provided in the repository can be used to fully test these tools.

Other tools are based on the analysis of parameters obtained locally at the infected machine, such as the frequency of read and write operations [27], [28], [34], the number of files encrypted [6], [39], and the file data entropy [36], [50]. These parameters can be obtained by intercepting system calls used for file access. The repository provides a file for each trace containing the I/O operations. File data entropy cannot be computed from this I/O operations file, because it contains only the metadata of the operations; however, all the data accessed (read or written) are present in the traffic trace (as the user accessed all files in the directory and the data were captured), so these tools can be fully tested.

Some tools use the presence of certain function calls, their frequency, and/or the use of cryptographic primitives for detection (column *function calls* in Table 3). We do not offer this information yet; however, only four tools base their detection exclusively on these parameters. Most tools that analyse system function calls or the cryptographic primitives combine them with other parameters, like file extensions [5], I/O operations [7], [30], [37], and data access information [48]. In these cases, the repository can be used to test the influence of each parameter except the function calls (these detection tools are labelled as partially testable in Table 3). Therefore, the repository is useful for testing the majority of tools present in the literature.

Finally, two tools [7], [8] analyse the binary of the malware file, in addition to other parameters, to detect the ransomware. We provide a link to the security-related web pages from which we obtained the binary samples; therefore, these tools can be tested using information that is available outside the repository, and they are labelled as partially testable.

VI. CONCLUSION

In this paper, we have presented a public repository containing the activity of more than 70 samples of ransomware, which were acquired while the ransomware was encrypting user files. The samples were collected over 5 years and the repository is still being updated. At the date of writing, the repository contains more than 1 TB in 206 traffic traces. The computing and networking scenarios in which the samples were run were aimed to emulate a corporate environment with shared documents in a workgroup. This proved to be appropriate not only for providing a realistic environment, but also for simplifying the running of the samples in different network conditions or disk access speeds.

The traces provided in the repository were used to test the ransomware detection tool presented in [49], and now other authors can benefit from these samples. We encourage their use for comparing performance results of ransomware detection algorithms; for this purpose, we have classified the existing detection tools according to whether they can be fully or partially tested using these data. We found that more than 86% of the tools can be at least partially tested, based either on network traffic or on the ransomware activity over user files. For easier data handling, besides the traffic capture files, the repository offers text files containing all the I/O operations performed on the user documents by the infected host. This information was extracted from the raw capture files.

Finally, we have presented two examples of the analysis of ransomware activity using the public data in the repository to show its possibilities. Using the exported I/O operations list, we could reveal different activity patterns in the two selected ransomware samples. These patterns influence the effectiveness of some detection techniques.

Ransomware samples can become deactivated because their C&C servers are taken down, therefore making it impossible to test new detection tools against their behaviour. The data provided in this repository can help the testing phase of new tools against old behaviours, which could appear again in new malware. The aim of this study was to contribute to the development of new ransomware detection tools not only by offering new algorithms, but also by making the testing process easier and comparable, as the collection of samples is always complicated.

REFERENCES

[1] D. Emm, R. Unuchek, M. Garnaeva, A. Liskin, D. Makrushin, and F. Sinitsyn, "IT threat evolution in Q3 2016," Kaspersky Labs, Moscow, Russia, Tech. Rep., 2016. [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07183248/KL_Q3_Malware_Report_ENG.pdf

[2] B. O'Gorman, "ISTR Internet security threat report," Symantec, Mountain View, CA, USA, Tech. Rep. 24, Feb. 2019.

[3] K. Cabaj and W. Mazurczyk, "Using software-defined networking for ransomware mitigation: The case of CryptoWall," *IEEE Netw.*, vol. 30, no. 6, pp. 14–20, Nov. 2016.

[4] A. O. Almashhadani, M. Kaiiali, S. Sezer, and P. O'Kane, "A multi-classifier network-based crypto ransomware detection system: A case study of locky ransomware," *IEEE Access*, vol. 7, pp. 47053–47067, 2019.

[5] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," 2016, *arXiv:1609.03020*. [Online]. Available: http://arxiv.org/abs/1609.03020

[6] A. Kharraz and E. Kirda, "Redemption: Real-time protection against ransomware at end-hosts," in *Research in Attacks, Intrusions, and Defenses*. Atlanta, GA, USA: Springer, 2017, pp. 98–119, doi: 10.1007/978-3-319-66332-6_5.

[7] M. M. Hasan and M. M. Rahman, "RansHunt: A support vector machines based ransomware analysis framework with integrated feature set," in *Proc. 20th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2017, pp. 1–7.

[8] S. K. Shaukat and V. J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning," in *Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2018, pp. 356–363.

[9] E. Berrueta, D. Morato, E. Magana, and M. Izal, "A survey on detection techniques for cryptographic ransomware," *IEEE Access*, vol. 7, pp. 144925–144944, 2019.

[10] *Malware Traffic Analysis*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.malware-traffic-analysis.net/

[11] *Hybrid Analysis*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.hybrid-analysis.com

[12] (2019). *Ransomware*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.knowbe4.com/ransomware

[13] R. Wainwright, "Internet organised crime thread assessment (IOCTA)," Europol European Police Office, Hague, The Netherlands, Tech. Rep., 2016, doi: 10.2813/275589.

[14] C. De Bolle, "Internet organised Crime thread assessment (IOCTA)," Europol European Police Office, Hague, The Netherlands, Tech. Rep., 2019.

[15] R. Smani and C. Beek, "McAfee labs threats report," McAfee Labs, Santa Clara, CA, USA, Tech. Rep., Dec. 2018.

[16] B. O'Gorman, "McAfee labs threats report," McAfee Labs, Santa Clara, CA, USA, Tech. Rep., Aug. 2019.

[17] M. Labs, "Cybercrime tactics and techniques," Malwarebytes, Santa Clara, CA, USA, Tech. Rep., Apr. 2017.

[18] R. Brandom. (Jul. 2017). *Ransomware Victims Have Paid Out More Than 25 Million, Google Study Finds*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.theverge.com/2017/7/25/16023920/ransomware-statistics-locky-cerber-google-research

[19] T. Ganacharya. (May 2017). *WannaCrypt Ransomware Worm Targets Out-of-Date Systems*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.microsoft.com/security/blog/2017/05/12/wannacrypt-ransomware-worm-targets-out-of-date-systems/?source=mmpc

[20] M. Field. (Oct. 2018). *WannaCry Cyber Attack Cost The NHS 92m as 19,000 Appointments Cancelled*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.telegraph.co.uk/technology/2018/10/11/wannacry-cyber-attack-cost-nhs-92m-19000-appointments-cancelled/

[21] C. Cimpanu. (Aug. 2019). *Ransomware Hits Hundreds of Dentist Offices in the US*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.zdnet.com/article/ransomware-hits-hundreds-of-dentist-offices-in-the-us/

[22] J. M. Esparza and T. B. Team. (Nov. 2019). *Spanish Consultancy Everis Suffers Bitpaymer Ransomware Attack: A Brief Analysis*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.blueliv.com/cyber-security-and-cyber-threat-intelligence-blog-blueliv/research/everis-bitpaymer-ransomware-attack-analysis-dridex/

[23] (Oct. 2019). *Ransomware Hits Several Spanish City Halls*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.pandasecurity.com/mediacenter/news/ransomware-spanish-city-halls/

[24] M. M. Ahmadian, H. R. Shahriari, and S. M. Ghaffarian, "Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares," in *Proc. 12th Int. Iranian Soc. Cryptol. Conf. Inf. Secur. Cryptol. (ISCISC)*, Sep. 2015, pp. 79–84, doi: 10.1109/iscisc.2015.7387902.

[25] S. Chadha and U. Kumar, "Ransomware: Let's fight back!," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 925–930.

[26] A. S. Team. (Jun. 2017). *Cerber Ransomware Is Not on Vacation This Summer*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.acronis.com/en-us/blog/posts/cerber-ransomware-not-vacation-summer

[27] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and drop It): Stopping ransomware attacks on user data," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2016, pp. 303–312, doi: 10.1109/ICDCS.2016.46.

[28] J.-Y. Paik, K. Shin, and E.-S. Cho, "Poster: Self-defensible storage devices based on flash memory against ransomware," in *Proc. IEEE Symp. Secur. Privacy*, May 2016, pp. 1–2.

[29] M. Shukla, S. Mondal, and S. Lodha, "Poster: Locally virtualized environment for mitigating ransomware threat," in *Proc. SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Oct. 2016, pp. 1784–1786, doi: 10.1145/2976749.2989051.

[30] T. Lu, L. Zhang, S. Wang, and Q. Gong, "Ransomware detection based on V-detector negative selection algorithm," in *Proc. Int. Conf. Secur., Pattern Anal., Cybern. (SPAC)*, Dec. 2017, pp. 531–536.

[31] N. Agrawal, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Generating realisticimpressionsfor file-system benchmarking," *ACM Trans. Storage*, vol. 5, no. 4, pp. 1–30, Dec. 2009, doi: 10.1145/1629080.1629086.

[32] E. Berrueta, D. Morato, E. Magana, and M. Izal, "High-speed analysis of SMB2 file sharing traffic without TCP stream reconstruction," in *Proc. IEEE Int. Symp. Meas. Netw. (M&N)*, Jul. 2019, pp. 1–6.

[33] (May 2016). *CryptXXX and Cerber Ransomware Get Major Updates*. Accessed: Dec. 19, 2019. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/cryptxxx-and-cerber-ransomware-get-major-updates

[34] C. Moore, "Detecting ransomware with honeypot techniques," in *Proc. Cybersecur. Cyberforensics Conf. (CCC)*, Aug. 2016, pp. 77–81.

[35] M. M. Ahmadian and H. R. Shahriari, "2entFOX: A framework for high survivable ransomwares detection," in *Proc. 13th Int. Iranian Soc. Cryptol. Conf. Inf. Secur. Cryptol. (ISCISC)*, Sep. 2016, pp. 79–84.

[36] F. Mbol, J.-M. Robert, and A. Sadighian, "An efficient approach to detect torrentlocker ransomware in computer systems," in *Proc. Int. Conf. Cryptol. Netw. Secur.* Milan, Italy: Springer, 2016, pp. 532–541.

[37] A. Continella, A. Guagnelli, G. Zingaro, G. D. Pasquale, A. Barenghi, S. Zanero, and F. Maggi, "ShieldFS: A self-healing, ransomware-aware filesystem," in *Proc. 32nd Annu. Conf. Comput. Secur. Appl. ACSAC*, 2016, doi: 10.1145/2991079.2991110.

[38] Y. Feng, C. Liu, and B. Liu, "Poster: A new approach to detecting ransomware with deception," in *Proc. 38th IEEE Symp. Secur. Privacy Workshops*, May 2017, pp. 1–2.

[39] E. Kirda, "UNVEIL: A large-scale, automated approach to detecting ransomware (keynote)," in *Proc. IEEE 24th Int. Conf. Softw. Anal., Evol. Reeng. (SANER)*, Feb. 2017, p. 1, doi: 10.1109/saner.2017.7884603.

[40] R. Vinayakumar, K. P. Soman, K. K. Senthil Velan, and S. Ganorkar, "Evaluating shallow and deep networks for ransomware detection and classification," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 259–265.

[41] J. Chen, C. Wang, Z. Zhao, K. Chen, R. Du, and G.-J. Ahn, "Uncovering the face of Android ransomware: Characterization and real-time detection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1286–1300, May 2018.

[42] J. A. Gómez-Hernández, L. Álvarez-González, and P. García-Teodoro, "R-locker: Thwarting ransomware action through a honeyfile-based approach," *Comput. Secur.*, vol. 73, pp. 389–398, Mar. 2018.

[43] M. Alam, S. Bhattacharya, D. Mukhopadhyay, and A. Chattopadhyay, "RAPPER: Ransomware prevention via performance counters," 2018, *arXiv:1802.03909*. [Online]. Available: http://arxiv.org/abs/1802.03909

[44] F. Quinkert, T. Holz, K. Tozammel Hossain, E. Ferrara, and K. Lerman, "RAPTOR: Ransomware attack PredicTOR," 2018, *arXiv:1803.01598*. [Online]. Available: http://arxiv.org/abs/1803.01598

[45] B. A. Saleh Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Redundancy coefficient gradual Up-weighting-based mutual information feature selection technique for crypto-ransomware early detection," 2018, *arXiv:1807.09574*. [Online]. Available: http://arxiv.org/abs/1807.09574

[46] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based on N-gram of opcodes," *Future Gener. Comput. Syst.*, vol. 90, pp. 211–221, Jan. 2019.

[47] R. Moussaileb, B. Bouget, A. Palisse, H. Le Bouder, N. Cuppens, and J.-L. Lanet, "Ransomware's early mitigation mechanisms," in *Proc. 13th Int. Conf. Availability, Rel. Secur.*, 2018, pp. 1–10.

[48] S. Mehnaz, A. Mudgerikar, and E. Bertino, "Rwguard: A real-time detection system against cryptographic ransomware," in *Proc. Int. Symp. Res. Attacks, Intrusions, Defenses*. Heraklion, Crete, Greece: Springer, 2018, pp. 114–136.

[49] D. Morato, E. Berrueta, E. Magaña, and M. Izal, "Ransomware early detection by the analysis of file sharing traffic," *J. Netw. Comput. Appl.*, vol. 124, pp. 14–32, Dec. 2018.

[50] K. Lee, S.-Y. Lee, and K. Yim, "Machine learning based file entropy analysis for ransomware detection in backup systems," *IEEE Access*, vol. 7, pp. 110205–110215, 2019.

EDUARDO BERRUETA was graduated in telecommunication engineering from the Public University of Navarre (UPNA), Spain, in 2016. He received the M.Sc. degree in telecommunication engineering from UPNA, in 2018, where he is currently pursuing the Ph.D. degree with the Telecommunications, Networks and Services Research Group. Previously, he attended the University of Turin for completing his thesis on software defined networking. In 2016, he held a Scholarship at the Automatics and Computing Department. In 2017 and 2018, he was a Research Assistant with the Telecommunications, Networks and Services Research Group, UPNA.

DANIEL MORATO received the M.Sc. degree in telecommunication engineering and the Ph.D. degree from the Public University of Navarre, Spain. In 2002, he was a Visiting Postdoctoral Fellow with the Electrical Engineering and Computer Sciences Department, University of California at Berkeley, Berkeley. Since 2006, he has been working at the Public University of Navarre. He is currently an Associate Professor with the Department of Electrical, Electronic and Communications Engineering. In 2014, he became a member of the Institute of Smart Cities. His research interests include high-speed networks, the performance and traffic analysis of Internet services, and network monitoring.

EDUARDO MAGAÑA received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Public University of Navarre, Pamplona, Spain, in 1998 and 2001, respectively. Since 2005, he has been an Associate Professor with the Public University of Navarra. In 2002, he was a Postdoctoral Visiting Research Fellow with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley. His main research interests are network monitoring, traffic analysis, and performance evaluation of communication networks.

MIKEL IZAL received the M.Sc. and Ph.D. degrees in telecommunication engineering, in 1997 and 2002, respectively. In 2003, he worked as a Scientific Visitant at the Institute Eurecom, France, performing measures in network tomography and peer-to-peer systems. Since 2013, he has been with the Public University of Navarre, where he is currently an Associate Professor. His research interests include traffic analysis, network tomography, high-speed next generation networks, and peer-to-peer systems.