The Cipher, the Random and the Ransom: A Survey on Current and Future Ransomware

Ziya Alper Genç
University of Luxembourg
Interdisciplinary Centre for Security, Reliability and Trust
ziya.genc@uni.lu

Gabriele Lenzini
University of Luxembourg
Interdisciplinary Centre for Security, Reliability and Trust
gabriele.lenzini@uni.lu

Peter Y.A. Ryan
University of Luxembourg
Interdisciplinary Centre for Security, Reliability and Trust
peter.ryan@uni.lu

ABSTRACT

Although conceptually not new, ransomware recently regained attraction in the cybersecurity community: notorious attacks in fact have caused serious damage, proving their disruptive effect. This is likely just the beginning of a new era. According to a recent intelligence report by Cybersecurity Ventures, the total cost due to ransomware attacks is predicted to exceed $5 billion in 2017. How can this disruptive threat can be contained? Current anti-ransomware solutions are effective only against existing threats, and the worst is yet to come. Cyber criminals will design and deploy more sophisticated strategies, overcoming current defenses and, as it commonly happens in security, defenders and attackers will embrace a competition that will never end. In this arm race, anticipating how current ransomware will evolve may help at least being prepared for some future damage.

In this paper, we describe existing techniques to mitigate ransomware and we discuss their limitations. Discussing how current ransomware could become even more disruptive and elusive is crucial to conceive more solid defense and systems that can mitigate zero-day ransomware, yielding higher security levels for information systems, including critical infrastructures such as intelligent transportation networks and health institutions.

KEYWORDS

ransomware threat, ransomware mitigation, malware, cybersecurity, survey

1 INTRODUCTION

When installed on a system, a ransomware encrypts files or blocks functionalities and when the job is done it asks for a ransom. The victim is left with the choice between paying up and regain access to the files and functionalities or never being able to use the system again. The ransom is usually paid in anonymous cryptocurrencies, like Bitcoin [35], leaving the ominous transactions untraceable by the authorities.

No risk of being seized together with low development effort have made ransomware a very popular weapon in the arsenal of cyber criminals. Kaspersky reports that in every 40 seconds a business is attacked by ransomware and that frequency is fourfold for individuals [26]. A ransomware is therefore a type of malware but the nature of ransomware attacks significantly differ from the ones of conventional malware in terms of the economical damage and the recoverability. When a system is infected by a crypto-ransomware variant, the victim's files are encrypted using strong cryptography [8] and the recovery may not be possible. Given this situation even the Federal Bureau of Investigation (FBI) reportedly advises victims to simply pay the ransom [30]. However paying the ransom may not guarantee to obtain the decryption keys and recover the files [33].

While Microsoft Windows platform continues to be the major target of ransomware threat [40], recently a Korean hosting firm have been hit by a Linux ransomware and had to pay $1 million [17]. Ransomware are therefore cross-platform, targeting indiscriminately private citizens and companies, and able to hit many countries at once indistinguishably. The infamous WannaCry ransomware recently attacked more than 200 000 computers in 150 countries [15].

Attacks does not seem to slow down in the near future. According to a recent intelligence report by Cybersecurity Ventures, the total cost due to ransomware attacks is predicted to exceed $5 billion in 2017 [44]. Ransomware threat is likely to have more consequences than just economical *e.g.,* denial of services, downgrade in service quality, lost of social trust or lost of trust in Information and Communications Technology (ICT). Ransomware attacks may even cause problems of civil liability and culminate lawsuits against victim companies and institutions from customers *e.g.,* by patients whose care are delayed or hindered. All these circumstances draw attention of information security community and there have been several proposals to mitigate ransomware.

In this paper we review current defense techniques for ransomware, discussing their strong and weak points. Then, we discuss what potential strategies could ransomware designer implement to bypass current countermeasures to continue causing damage for an extended period: we introduce original ransomware variants that employ rootkit techniques and white-box cryptography, and, inspired by the cybersecurity incidents occurred in real-world applications, we point out new possible ransomware targets and attack types.

2 BACKGROUND

Ransomware aims to extort money through preventing access to data or functionality on victim's system. *Cryptographic* ransomware accomplishes this goal by encrypting files using strong cryptography [8] while holding the description keys so that victims are forced to pay the ransom to obtain those keys and regain access to their files. Another variant, the *locker* ransomware, reaches this aim via taking control of

the victim's system and denies functionality. In this case, user data are untouched but the infected system becomes unusable.

2.1 Defense Systems

Current Mitigations. Existing anti-ransomware applications, excluding the inefficient and ineffective practice to back-up and restore files, can be grouped in three main families. The first includes defences which monitor an application's activity in real time in search for patterns that justify blocking a potential ransomware from working (*behavioral analysis*). The second contains defences that create the conditions to nullify or reverse the effect of a ransomware (*key escrow strategies*). The last one includes defences that isolate the binary of applications and analyze their code in search for calls to cryptographic operations which would reveal at least in potential the presence of a malicious intention (*detection of cryptographic primitives*). In detail:

- *Behavioral analysis*: In this approach, ransomware defense systems examine the behavior of an application and its interactions with the environment, *e.g.,* file system activity, network connections and modifications on operating system (OS) components. There are various proposals in the literature that uses behavioral analysis approach. One of them, UNVEIL [27] generates an artificial user environment and monitors desktop lockers, file access patterns and I/O data entropy. Another one, CRYPTODROP [37] observes file type changes and measures file modifications using similarity-preserving hash functions and Shannon Entropy to recognize ransomware. Moreover, SHIELDFS [13] monitors low-level file system activities and collects the following features: *folder listing, file read/write/rename, file type* and *write entropy*. A ransomware is detected by comparing these characteristics with that of benign applications. Unlike the previous two, SHIELDFS can recover the files which are already encrypted before detection, though this capability comes with a significant performance overhead.
- *Key escrow*: In this approach, cryptographic materials generated by ransomware on the victim's system are obtained and held in escrow to later use for recovery. For instance, PAYBREAK [28] is a key escrow based mitigation system and works by intercepting cryptographic Application Programming Interface (API), extracting passed parameters and storing them in a secure key vault. In the case of infection, the defense system tries to decrypt the encrypted files using the stored keys and parameters. However, this approach can succeed only if the cryptographic functions employed by the ransomware are correctly recognized and the parameters passed to the APIs are logged. While this is feasible for built-in cryptographic functions on the host system, ransomware that utilizes third-party libraries can bypass detection through *obfuscation* [28] as we will discuss in Section 3.2.
- *Detection of cryptographic primitives*: In this approach, binary programs are analyzed to identify cryptographic operations in their executable codes. To this goal, [19] traces the execution of applications and monitors I/O relationship in the program flow. Based on the occurrences of *bitwise arithmetic instructions* and *loops*, and relationships between *the inputs and outputs of the program routines*, heuristics are applied to recognize the cryptographic algorithms. On the other hand, [31] uses static analysis and Data Flow Graph (DFG) isomorphisms to identify cryptographic algorithms in the binary programs. Basically, this technique work as follows: First, the DFG of binary program is build. Next, the DFG in hand is normalized using rewrite rules in order to remove the variations due to complier optimizations. Finally, subgraphs which are isomorphic to graph signatures of cryptographic algorithms are searched in the DFG. A match directly flags that the corresponding algorithm exists in the analyzed program.

Other Methods. The main shortcoming of behavioral analysis approach for ransomware prevention is the potential false results due to the lack of an accurate decision mechanisms. In order to increase the accuracy of detection, anti-ransomware systems aim to consider more indicators which distinguish ransomware from benign applications. As the number of rules increases, simple decision techniques become inadequate. For this purpose, Machine Learning (ML) algorithms are used to analyze benign applications and known ransomware samples to extract feature vectors, build models and classify them. Recently, a ML based ransomware defense system has been made commercially available [21]. Meanwhile, the debate over the security of ML based malware defense systems continues. For instance, Hu and Tan proposed an algorithm to generate adversarial examples which cause the ML based malware detection systems to misclassify the applications [24].

Beside technical solutions, Lu and Liao suggest improving user awareness to help mitigate ransomware [32]. Security education for end users would effectively prevent ransomware attacks originating from phishing or spam emails. However, the attack surface that ransomware can exploit is far more larger. As the recent WannaCry attack demonstrates, ransomware evolution has enabled it to spread over the network. Especially, zero-day attacks can amplify the damage of ransomware and user education cannot help in this case.

3 POTENTIAL NEW THREATS

We start by giving high-level descriptions of advanced techniques that ransomware may utilize to defeat the defense systems characterized in the previous section. Next, we point out new areas that ransomware may exploit and extend the attack surface that next generation ransomware may target. In each discussion, our observations are supported by the real world incidents.

3.1 Rootkit-based Ransomware

Rootkit is a type of malware that has the ability to conceal its activities on the target computer system, *e.g.,* code executions, file I/O, network and connections [22]. The capability of hiding malicious operations is achieved by hooking operating system's APIs in order to filter and remove the rootkit's

traces, as depicted in Figure 1. Since a rootkit clears its footprints from APIs that inspect file and memory access, the rootkits are harder to detect than other types of malware.

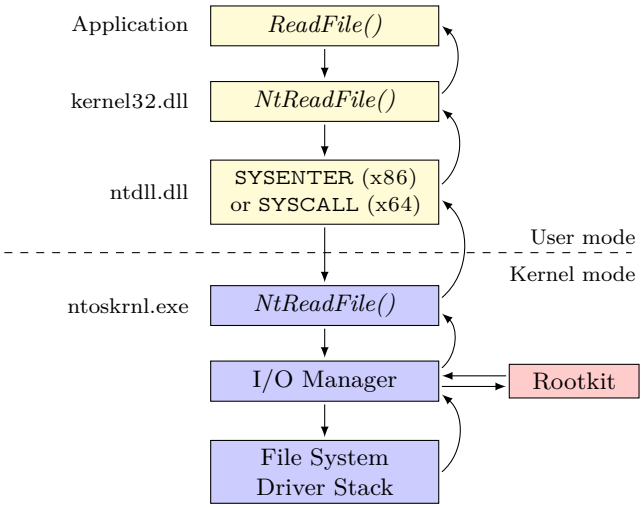

Figure 1: Interception of read calls by a kernel mode rootkit in order to hide its trace.

Hooking system APIs can be accomplished in several ways, including changing the function addresses in Import Address Table (IAT), patching System Service Dispatch Table (SSDT) in kernel level, and injecting code into applications (DLL injection) [39]. Starting from Windows Server 2003, x64-based versions of Windows platform introduced Kernel Patch Protection (KPP) which forces kernel mode drivers to be digitally signed, hence prevents unknown modification of code or critical structures in Windows kernel [34]. Nevertheless, cybercriminals frequently used stolen certificates to sign malware in order to penetrate this defense [9, 41]. Ransomware authors also seems to have this capability. A recent VirusTotal report shows that a sample of Razy ransomware has a valid digital signature [45].

Implementations of current ransomware defense approaches deeply rely on the security guarantees of the host OSes. While increasing the bar for cybercriminals, state-of-the-art ransomware defense systems utilizes user mode hooks or kernel mode drivers to monitor behavior of applications and stop ransomware [13, 27, 28, 37]. Although there is currently no known ransomware which utilizes the advanced techniques of rootkits, the aforementioned defense systems may not detect a rootkit-based ransomware.

3.2 Obfuscation

Obfuscation is the practice of making a software implementation incomprehensible through a sequence of transformations while preserving the program semantics [12]. Originally, legitimate vendors utilized obfuscation to protect intellectual property in software implementation. However, malware authors also take advantage of obfuscation to conceal malicious executable code in the binary programs. Concordantly, obfuscated malware can evade from *signature based detection* techniques which is one of the oldest approaches in the battle with malware.

```
push    rbp
mov     rbp, rsp
mov     WORD PTR [rbp-2], 1
mov     WORD PTR [rbp-4], 2
movzx   eax, WORD PTR [rbp-2]
movzx   edx, WORD PTR [rbp-4]
imul    eax, edx
mov     WORD PTR [rbp-6], ax
movsx   eax, WORD PTR [rbp-6]
pop     rbp
ret
```

```
push    rbp
mov     rbp, rsp
mov     WORD PTR [rbp-2], 1
mov     WORD PTR [rbp-4], 2
movzx   eax, WORD PTR [rbp-2]
add     eax, 1
mov     WORD PTR [rbp-2], ax
and     WORD PTR [rbp-2], 32767
movzx   eax, WORD PTR [rbp-2]
sub     eax, 1
mov     WORD PTR [rbp-2], ax
and     WORD PTR [rbp-4], 32767
movzx   edx, WORD PTR [rbp-4]
movzx   eax, WORD PTR [rbp-2]
imul    eax, edx
mov     WORD PTR [rbp-6], ax
movsx   eax, WORD PTR [rbp-6]
pop     rbp
ret
```

Figure 2: Two code fragments that are semantically equivalent and multiply the integers 1 and 2. Left, the original function. Right, the transformed function by adding ineffective instructions shown inside red boxes. Note that the code's appearance is changed while keeping its behavior same.

Obfuscating malware can be categorized into four types: *encrypting*, *oligomorphic*, *polymorphic* and *metamorphic* malware [48]. The members of the first type encrypts malicious code segment in the binary program and decrypt it in the runtime. This involves a decryptor function embedded in the malware body to decrypt and execute the malicious code. Anti-malware systems, though, would still recognize the decryptor function and identify malicious software. Thus, the second type, oligomorphic malware, carries a set of encrypted decryptors in data segment of binary and changes the decryptor in each generation. However, the number of decryptors is limited and therefore all of them eventually gets identified by anti-malware systems. On the other hand, polymorphic malware mutates its decryption engine randomly, hence evades signature based detection. The means of mutation include dead code insertion, register reassignment, subroutine reordering, instructor substitution, code transposition & integration. For instance, dead code insertion is the practice of adding code that has no effect on the functionality of the software and is shown in Figure 2. For the details of other techniques, we refer the reader to [2]. Anti-malware vendors developed *sandboxing* approach to help detection, which works by observing the program's behavior in a safe environment. Once the polymorphic malware is executed in sandbox and the constant malicious part is decrypted in the memory, signature based detection can be applied. The race between cybercriminals and anti-malware vendors resulted the appearance of metamorphic malware which actively recognizes, parses and mutates its whole body. As it does not contain a constant body, and thus cannot be detected via signature analysis [7], metamorphic malware has been considered to be most dangerous type.

In the ransomware side, the situation seems to be safe for now. As of today, there is no known instance of obfuscated ransomware through aforementioned techniques. Contemporary ransomware utilizes binary packers, *e.g.,* UPX[1], ASPack[2] or PEtite[3], which are used to compress the compiled code in order to make the size of executable even smaller. However, malware authors do not confine themselves to well-known packers, often write their own obfuscator routines and utilize combined packers [43]. This multi-layer protection may hinder defense systems based on API monitoring (if third party crypto libraries statically linked) and sandboxing. In the case of an unlucky event of infection, such a ransomware can be devastating.

3.3 White-Box Cryptography

White-box cryptography is the concept of protecting the sensitive data hard-coded in a software implementation [10, 11]. In particular, main focus of this domain is to embed secret keys into the source code in such a way that it is hard to extract them from compiled binary. An example of a Feistel network based block cipher and its fixed-key white-box implementation are illustrated in Figure 3. Although white-box cryptography is not a new idea (it is first introduced in 2002), no secure white-box implementation of the block cipher AES exists yet, for instance, previous proposals are found to be open to key extraction and table-decomposition attacks [5]. Nevertheless, white-box cryptography still continues to be an active field of research [3, 6, 25].

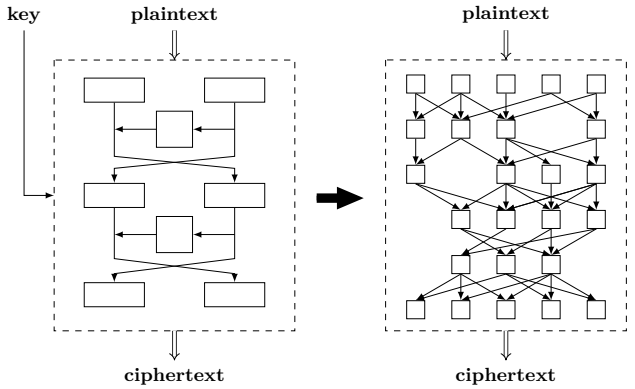

Figure 3: Left, a block cipher algorithm based on Feistel network structure. Right, a white-box implementation of that block cipher where a key is hard-coded into the algorithm.

Currently, ransomware implementations cannot protect the secret keys in the memory during the encryption process. Using this weakness, defense systems can extract these keys using various techniques. For instance, a key escrow like approach monitors calls to known cryptographic APIs (either built in or third party) and stores parameters of encryption functions in a vault [28]. In virtual environments, point-in-time snapshot of memory would also reveal those keys and recovery could be possible. Furthermore, some ransomware families encrypt victim's files using a key which is hard-coded in the ransomware body [29]. In this case, binary analysis can be utilized to search for static encryption keys in the compiled code. In other words, one can interact with the ransomware and propose solutions if the encryption keys resides unprotected in the memory. That being said, key extraction from securely implemented white-box algorithms is meant to be hard. Therefore, introducing of secure white-box implementations of block ciphers can tip the balance in favor of ransomware authors.

3.4 Ransomware of Things

Internet of Things (IoT) refers to the interconnected network of physical devices that can communicate over the Internet [20]. An IoT device can be equipped with electronic components, firmware, software, various types of sensors to collect information and actuators that allows to interact with the physical environment. Besides electronic devices like televisions, mobile phones and surveillance systems, in today's world, cars, planes, buildings, kitchen gadgets and even toys are also connected to the web.

IoT devices has been a part of our daily lives for a long time and can be seen virtually everywhere. However, IoT devices are inherently resource-constrained (CPU with low clock rate, small memory size). As such, the available options for cryptographic algorithms to use is limited when designing a secure communication protocol [49]. The security issues with IoT have always been a concern in information community [1], most importantly access control problems.

Given that the vulnerabilities in IoT devices and the high motivation of cyber criminals, there have already occurred several alarming and threatening ransomware incidents as follows. Hackers took control of ticket machines of San Fransisco's public transportation network and claimed ransom [47]. Furthermore in Austria, a hotel had to pay ransom after a ransomware infected its management system and blocked generating new cards [4]. Researchers demonstrated a proof-of-concept that the control of an Internet-enabled thermostat can be taken by a ransomware, allowing them to change the heating settings [42]. Similarly, A recent security report states that cybercriminals launched a Permanent Denial of Service (PDoS) attack on IoT devices which wipes all data on the device and destroy its firmware and/or basic functions, causing a permanent corruption [36].

By extending the attack surface and lack of adequate security, IoT has a potential of opening doors to novel ransomware attacks. For example, researchers demonstrated that it is possible to take control of a car and remotely stop it [18]. Also, another group of researchers showed that 75% of bluetooth smart door locks can be wirelessly hacked [46]. Given these facts, it is reasonable to ask the following questions: Consider that your car was remotely stopped in a rural area. *Would*

[1]Ultimate Packer for eXecutables, https://upx.github.io/
[2]ASPack, http://www.aspack.com/aspack.html
[3]PEtite, http://www.un4seen.com/petite/

you pay the ransom to re-activate the car's engine? Likewise, when you return your home in the middle of the night and see that your door is locked. *Would you pay the ransom to go in your home?* The picture may become worse for the enterprises, as the ransom amounts can be set higher and this makes the enterprises a more plausible target for cyber criminals. But the negative effects of a ransomware attack is beyond the money: the damage in the reputation and work loss should also be counted. Taking into the account that the security flaws in IoT devices do not seem to be fixed soon, or even fixable [38], ransomware attacks may gravitate towards IoT in the near future.

3.5 Socio Technical Attacks

The ultimate goal of cyber-criminals is to obtain money as much as possible. To achieve this, they can become very creative and employ novel marketing strategies. In one of these, a ransomware variant called Popcorn Time offers an option to victims who want to get decryption keys without paying. The condition is first victim infects other two ones and these two victims pay the ransom. Then, the first victim obtains the keys. The initial samples of Popcorn Time ransomware have an encryption key embedded in the malware body [14]. Although the key can be extracted from the current sample of Popcorn Time and files can be recovered for now, previous evolution of ransomware suggests that future samples of Popcorn Time may become more effective.

To this day, the vast majority of famous ransomware families share the same principle. Extortion by holding decryption keys can be expected to succeed when its vital for victims to regain access to their data. However, on the other side of medallion, there is another fact. Some data may need to be kept private such that when leaked, data owner may lose advantage and/or have economical damage. Thus, another way to extort victims can be to exfiltrate sensitive data and ask for a ransom to not make it public. These data types may include trading secrets, financial records, medical history, government documents, details of high-tech projects, blue-prints of critical infrastructures, and internal/private communications. For example, the disclosure of data breaches reduced the purchase price of Yahoo by $350 million when it is acquired by Verizon [16]. It comes to mind that, instead of selling the leaked data in the underground market, hackers can try to claim a ransom to get a higher revenue. Another attack hit Sony Pictures, hackers compromised the computers and released sensitive data including company's financial records and e-mail messages of executives [23]. The contents of the breach put the company in a difficult situation so that one may ask the question: *Would Sony Pictures pay a ransom if attackers demand it?*

Lastly, we would like to point an important difference between extortion via encryption and data exfiltration. In the former case, the instance of threat comes to an end when the victims regain access to their files. In contrast, no one can guarantee that could retain cyber-criminals from asking for ransom again in the latter case. In this situation, it would be safe to expect that extortion via stealing sensitive information may be an increasing trend in the near future and prepare the network infrastructures against this threat.

4 CONCLUSION

Ransomware is a class of malware whose goal is to extort money, a goal that is facilitated by current anonymous currencies which guarantee to cyber-criminals to be paid without being traced. Then we need solid defense systems against what can easily degenerate in a pandemia of digital crimes. However, unlike conventional anti-malware systems, ransomware mitigation does not tolerate mistake. If the ransomware is implemented properly and the attack succeeds, then the damage taken may be irreversible.

Existing ransomware mitigation systems are build upon the analysis of collected samples but a better strategy is to anticipate the future, and be prepared for the ransomware that will come. In this respect, we described possible threats that ransomware may pose by relying on novel techniques, like root-kit, obfuscation, and white-box, not yet adopted in real attack as well as by targeting critical domains, such as the Internet of Things and the Socio-Technical systems, which will worrisomely amplify the effectiveness of ransomware attacks. Our research is timely, since it is known that we must design products keeping security in mind, not integrating after whereas network infrastructures must be carefully configured and fully patched in order to prevent ransomware attacks through data exfiltration. We hope that our observations help developing and building more robust defense systems against ransomware threat.

Acknowledgements. This work is supported by a partnership between "pEp Security SA" and the Interdisciplinary Centre for Security, Reliability and Trust.

REFERENCES

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The Internet of Things: A survey. *Computer Networks* 54, 15 (2010), 2787 – 2805.
[2] Arini Balakrishnan and Chloe Schulze. 2005. Code obfuscation literature survey. (2005).
[3] Marc Beunardeau, Aisling Connolly, Remi Geraud, and David Naccache. 2016. White-box cryptography: Security in an insecure environment. *IEEE Security & Privacy* 14, 5 (2016), 88–92.
[4] Dan Bilefsky. 2017. Hackers Use New Tactic at Austrian Hotel: Locking the Doors. (30 Jan. 2017). Retrieved June 19, 2017 from https://www.nytimes.com/2017/01/30/world/europe/hotel-austria-bitcoin-ransom.html
[5] Andrey Bogdanov and Takanori Isobe. 2015. White-Box Cryptography Revisited: Space-Hard Ciphers. In *Proc. 22nd ACM Conf. Comput. and Commun. Security (CCS '15)*.
[6] Andrey Bogdanov, Takanori Isobe, and Elmar Tischhauser. 2016. Towards Practical Whitebox Cryptography: Optimizing Efficiency and Space Hardness. In *Proc. 22nd Int. Conf. Theory and Application of Cryptology and Inform. Security (ASIACRYPT '16)*.
[7] Jean-Marie Borello and Ludovic Mé. 2008. Code obfuscation techniques for metamorphic viruses. *Journal in Computer Virology* 4, 3 (2008), 211–220.
[8] Bromium. 2014. Understanding Crypto-Ransomware. (2014). Retrieved June 22, 2017 from https://www.bromium.com/sites/default/files/rpt-bromium-crypto-ransomware-us-en.pdf
[9] Thomas M. Chen and Saeed Abu-Nimeh. 2011. Lessons from stuxnet. *Computer* 44, 4 (2011), 91–93.

[10] Stanley Chow, Philip Eisen, Harold Johnson, and Paul C. Van Oorschot. 2003. White-Box Cryptography and an AES Implementation. In *Proc. Int. Workshop Select. Areas in Cryptography (SAC '02)*.

[11] Stanley Chow, Phil Eisen, Harold Johnson, and Paul C. van Oorschot. 2003. A White-Box DES Implementation for DRM Applications. In *Proc. ACM Workshop on Digital Rights Manage. (DRM '02)*.

[12] Christian Collberg, Clark Thomborson, and Douglas Low. 1998. Manufacturing Cheap, Resilient, and Stealthy Opaque Constructs. In *Proc. 25th ACM Symp. Principles of Programming Languages (POPL '98)*.

[13] Andrea Continella, Alessandro Guagnelli, Giovanni Zingaro, Giulio De Pasquale, Alessandro Barenghi, Stefano Zanero, and Federico Maggi. 2016. ShieldFS: A Self-healing, Ransomware-aware Filesystem. In *Proc. 32nd Annu. Conf. Comput. Security Applicat. (ACSAC '16)*.

[14] Paul Ducklin. 2016. Popcorn Time ransomware lets you off if you infect two other people. (15 Dec. 2016). Retrieved July 13, 2017 from https://nakedsecurity.sophos.com/2016/12/15/popcorn-time-ransomware-lets-you-off-if-you-infect-two-other-people/

[15] Shona Ghosh. 2017. The massive global cyberattack affecting 200,000 victims will cause more chaos on Monday. (14 May 2017). Retrieved June 23, 2017 from http://uk.businessinsider.com/europol-said-there-are-200000-cyberattack-victims-and-the-number-will-go-up-2017-5

[16] Vindu Goel. 2017. Verizon Will Pay $350 Million Less for Yahoo. (21 Feb. 2017). Retrieved July 13, 2017 from https://www.nytimes.com/2017/02/21/technology/verizon-will-pay-350-million-less-for-yahoo.html

[17] Dan Goodin. 2017. Web host agrees to pay $1m after it's hit by Linux-targeting ransomware. (6 June 2017). Retrieved June 20, 2017 from https://arstechnica.com/security/2017/06/web-host-agrees-to-pay-1m-after-its-hit-by-linux-targeting-ransomware/

[18] Andy Greenberg. 2015. Hackers Remotely Kill a Jeep on the Highway—With Me in It. (21 July 2015). Retrieved June 23, 2017 from https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[19] Felix Gröbert, Carsten Willems, and Thorsten Holz. 2011. Automated Identification of Cryptographic Primitives in Binary Programs. In *Proc. 14th Int. Conf. Recent Advances in Intrusion Detection (RAID '11)*.

[20] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems* 29, 7 (2013), 1645–1660.

[21] Peter Hale. 2017. Acronis True Image 2018: Artificial Intelligence Meets Intelligent Backup. (30 Aug. 2017). Retrieved October 2, 2017 from https://www.acronis.com/en-us/blog/posts/acronis-true-image-2018-artificial-intelligence-meets-intelligent-backup

[22] Greg Hoglund and James Butler. 2006. *Rootkits: subverting the Windows kernel*. Addison-Wesley Professional.

[23] Amanda Holpuch. 2014. Sony email hack: what we've learned about greed, racism and sexism. (15 Dec. 2014). Retrieved July 13, 2017 from https://www.theguardian.com/technology/2014/dec/14/sony-pictures-email-hack-greed-racism-sexism

[24] Weiwei Hu and Ying Tan. 2017. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. https://arxiv.org/abs/1702.05983. (2017).

[25] Yin Jia, TingTing Lin, and Xuejia Lai. 2016. A generic attack against white box implementation of block ciphers. In *Proc. Int. Conf. Comput. Inform. and Telecommun. Systems (CITS '16)*.

[26] Kaspersky. 2016. Security Bulletin 2016. (Dec. 2016). Retrieved June 22, 2017 from https://securelist.com/files/2016/12/KSB2016_Story_of_the_Year_ENG.pdf

[27] Amin Kharaz, Sajjad Arshad, Collin Mulliner, William Robertson, and Engin Kirda. 2016. UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware. In *Proc. 25th USENIX Security Symp. (USENIX Security '16)*.

[28] Eugene Kolodenker, William Koch, Gianluca Stringhini, and Manuel Egele. 2017. PayBreak: Defense Against Cryptographic Ransomware. In *Proc. ACM Asia Conf. Comput. and Commun. Security (ASIACCS '17)*.

[29] Ondrej Kubovič. 2016. Ransomware is everywhere, but even black hats make mistakes. (28 April 2016). Retrieved June 19, 2017 from https://www.welivesecurity.com/2016/04/28/ransomware-is-everywhere-but-even-black-hats-make-mistakes/

[30] Security Ledger. 2015. FBI's Advice on Ransomware? Just Pay The Ransom. (22 Oct. 2015). Retrieved June 22, 2017 from https://securityledger.com/2015/10/fbis-advice-on-cryptolocker-just-pay-the-ransom/

[31] Pierre Lestringant, Frédéric Guihéry, and Pierre-Alain Fouque. 2015. Automated Identification of Cryptographic Primitives in Binary Code with Data Flow Graph Isomorphism. In *Proc. 10th ACM Symp. Information Comput. and Commun. Security (ASIACCS '15)*.

[32] Xin Luo and Qinyu Liao. 2007. Awareness Education as the Key to Ransomware Prevention. *Information Systems Security* 16, 4 (2007), 195–202.

[33] Trend Micro. 2016. Kansas Hospital Hit by Ransomware, Extorted Twice. (23 May 2016). Retrieved June 23, 2017 from https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/kansas-hospital-hit-by-ransomware-extorted-twice

[34] Microsoft. 2007. Kernel patch protection: frequently asked questions. (Jan. 2007). Retrieved June 13, 2017 from https://msdn.microsoft.com/en-us/library/windows/hardware/Dn613955(v=vs.85).aspx

[35] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf. (2008).

[36] Radwire. 2017. "BrickerBot" Results In PDoS Attack. (4 May 2017). Retrieved June 23, 2017 from https://security.radware.com/ddos-threats-attacks/brickerbot-pdos-permanent-denial-of-service/

[37] Nolen Scaife, Henry Carter, Patrick Traynor, and Kevin R.B. Butler. 2016. CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. In *Proc. 36th Int. Conf. Distributed Computing Syst. (ICDCS '16)*.

[38] Bruce Schneier. 2014. The Internet of Things Is Wildly Insecure – And Often Unpatchable. (6 Jan. 2014). Retrieved June 23, 2017 from https://www.wired.com/2014/01/theres-no-good-way-to-patch-the-internet-of-things-and-thats-a-huge-problem/

[39] Spencer Smith and John Harrison. 2012. Rootkits. (2012). Retrieved June 13, 2017 from http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/rootkits.pdf

[40] Symantec. 2016. An ISTR Special Report: Ransomware and Businesses 2016. (19 July 2016). Retrieved June 20, 2017 from https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/ISTR2016_Ransomware_and_Businesses.pdf

[41] Peter Szor. 2011. Duqu–Threat Research and Analysis. (Nov. 2011). Retrieved June 13, 2017 from https://securingtomorrow.mcafee.com/wp-content/uploads/2011/10/Duqu.pdf

[42] Andrew Tierney. 2016. Thermostat Ransomware: a lesson in IoT security. (Aug. 2016). Retrieved June 19, 2017 from https://www.pentestpartners.com/security-blog/thermostat-ransomware-a-lesson-in-iot-security/

[43] Xabier Ugarte-Pedrero, Davide Balzarotti, Igor Santos, and Pablo G. Bringas. 2015. SoK: deep packer inspection: a longitudinal study of the complexity of run-time packers. In *Proc. 36th IEEE Symp. on Security and Privacy (S&P '15)*.

[44] Cybersecurity Ventures. 2017. Ransomware Damage Report. (18 May 2017). Retrieved June 23, 2017 from http://cybersecurityventures.com/ransomware-damage-report-2017-5-billion/

[45] VirusTotal. 2017. Scan report. (15 June 2017). Retrieved July 13, 2017 from https://virustotal.com/en/file/81fdbf04f3d0d9a85e0fbb092e257a2dda14c5d783f1c8bf3bc41038e0a78688/analysis/

[46] Paul Wagenseil. 2016. 75 Percent of Bluetooth Smart Locks Can Be Hacked. (Aug. 2016). Retrieved June 19, 2017 from http://www.tomsguide.com/us/bluetooth-lock-hacks-defcon2016,news-23129.html

[47] Elizabeth Weise. 2016. Ransomware attack hit San Francisco train system. (28 Nov. 2016). Retrieved June 22, 2017 from https://www.usatoday.com/story/tech/news/2016/11/28/san-francisco-metro-hack-meant-free-rides-saturday/94545998/

[48] Ilsun You and Kangbin Yim. 2010. Malware Obfuscation Techniques: A Brief Survey. In *Proc. 5th Int. Conf. Broadband, Wireless Computing, Commun. and Applicat. (BWCCA '10)*.

[49] Kai Zhao and Lina Ge. 2013. A Survey on the Internet of Things Security. In *Proc. 9th Int. Conf. Computational Intelligence and Security (CIS '13)*.