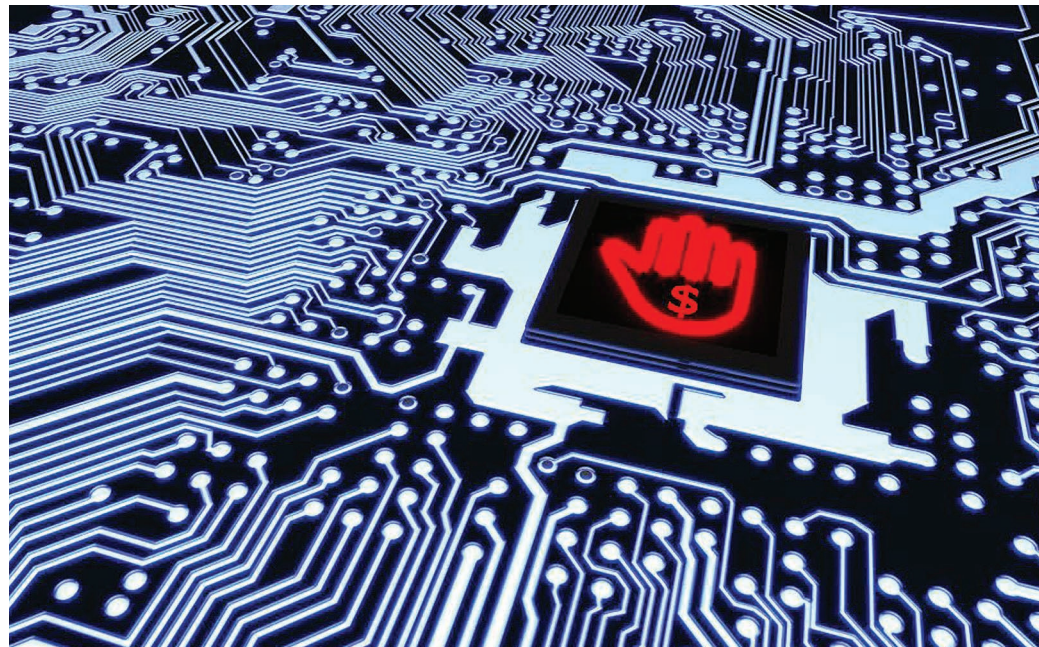


Protecting against Ransomware: A New Line of Research or Restating Classic Ideas?

Amin Kharraz | University of Illinois at Urbana-Champaign
William Robertson and Engin Kirda | Northeastern University

Malware attacks remain one of the most popular attack vectors in the wild. Compared to other types of malware, ransomware has recently become very popular among malware authors. Ransomware is a kind of scareware that locks a victim's computer until she makes a payment to regain access to her data. This class of malware is not a new concept (that is, such attacks have been in the wild since the last decade), but the growing number of high-profile ransomware attacks has resulted in increasing concern about how to defend against this class of malware. In 2016, several public and private sectors, including the healthcare industry, were impacted by ransomware.¹ Very recently, WannaCry, one of the successful ransomware attacks, impacted thousands of users around the world by exploiting the EternalBlue vulnerability, encrypting user data, and demanding a bitcoin payment in exchange for unlocking files.

In response to the increasing number of ransomware attacks, users are often advised to create backups of their critical data. Certainly, having a reliable data backup policy minimizes the potential costs of being infected with ransomware and is an important part of the IT management process. However, the growing number of paying victims suggests that technically unsophisticated users—who are the main targets of these attacks—do not follow these recommendations and easily



become paying victims of ransomware. Hence, ransomware authors continue to create new attacks as evidenced by the emergence of more sophisticated ransomware every day.

Although there has been some progress in identifying ransomware attacks, in practice, the primary defense mechanisms to detect, analyze, and defend against ransomware attacks are not very different from the detection techniques that are being used to identify other types of evasive malware attacks. Perhaps the main reason is that this type of malware, similar to other classes of malware, employs common evasion techniques to bypass known detection techniques, reach

end users, and successfully launch attacks. While this is a valid assumption about employing general evasion techniques, the current defense mechanisms cannot achieve the best detection results as evidenced by the increasing number of very successful ransomware attacks in the wild.

The security research community has recently begun tackling some of the challenges in identifying ransomware attacks. However, there are a set of high-level questions that are often asked about this specific area in malware research. In this article, we seek to answer:

- What are the new intellectual challenges in this specific area?

- What are the challenges in systems security to address these problems?
- Is it possible to tackle these challenges by incorporating current defense techniques?

We summarize some of the similarities between the defense mechanisms to detect ransomware attacks and other classes of malware as well as research problems that are specific to this area.

In many ways, ransomware benefits from classic malware development techniques, but there are specific features of ransomware that provide an advantage to defenders. At a high level, the goal of ransomware is often a reversible DoS attack on data availability. In practice, this means performing cryptographic operations on user data and modifying many data files. Defenders can use these features to enhance both the detection of and protection against ransomware in ways that are not applicable to malware in general.

Limitation of Current Defense Mechanisms

Ransomware attacks share undebatable similarities with other types of malware attacks particularly in making use of evasion techniques and distributing malicious payloads. Perhaps the main reason for this level of similarity is that adversaries' main goals before launching an attack on victims' machines are:

- to bypass common anti-malware solutions, and
- to utilize every possible distribution channel to expose as many victims as possible to such attacks.

Therefore, it is worthwhile to investigate which specific problems in detecting ransomware attacks are similar to other malware attacks and which problems are different in nature and require more

investigation. For example, similar to other types of malware attacks (for instance, Trojans), opening email attachments or clicking on malicious advertisements may increase the risk of being infected by ransomware. Therefore, some of the current techniques used to identify suspicious payloads are also useful in detecting the malicious binaries that deliver ransomware.

Similarly, some of the general static analysis techniques such as portable executable (PE) analysis tools or packet detection techniques can still provide helpful information about a given malicious binary. However, these tools and techniques rarely provide useful insights about the specific behavior of a given ransomware sample. More specifically, unlike most modern malware attacks, ransomware attacks are not usually designed to be stealthy after the infection phase, as the whole point of the attack is to notify victims that their machines are infected. Furthermore, the core functionality of a ransomware sample, the cryptosystem module, usually works similarly to the benign applications that are often used for privacy-preserving purposes. In fact, the similarity of the ransomware's behavior compared to a subset of benign applications as well as the differences from other types of malware attacks in the attack strategy have made the current automated analysis techniques less effective in detecting and analyzing the attacks, and protecting end users. Therefore, it is quite useful to develop tools that can accurately extract the ransomware behavior and improve the current automated analysis systems or end-point solutions given these similarities and differences.

Similarities and Differences with Other Classes of Malware

Similar to other malware attacks, ransomware payloads are usually

armed with techniques that make the detection or analysis of the payload more difficult. At the same time, the malicious binary has an additional set of core functionalities that differentiate the malicious payload from other types of malware attacks. This functionality determines how the encryption keys should be generated and maintained as well as how the malicious process should attack user data and request a ransomware fee. In the following sections, we explain each of these steps, highlight some of the techniques that have been introduced so far, and describe potential directions for better detection.

Enhancing Detection Techniques

Malware research is an arms race. Therefore, there is always the possibility that malware developers will find heuristics to bypass the detection mechanisms used in the analysis systems or on end-user machines. Therefore, developing techniques that can increase the cost of evasion, enhance the malware detection systems, and assist malware analysts in unmasking the inner workings and functions of the malicious code is quite useful in detecting all types of malware—including ransomware.

Automating payload analysis. Malware authors usually use several anti-analysis techniques to increase the level of the attack's sophistication. This makes the payload analysis largely a manual process. Therefore, developing techniques that facilitate the automatic examination of malicious binaries is highly desirable. Dynamic analysis is a promising technique to analyze the malicious binary and reveal the main functionalities of the malware sample. However, prior work^{2,3} showed that running a malware sample in an analysis environment and extracting its behavior is a nontrivial task as most of current malware families,

including ransomware, perform several different environmental checks to ensure that they are being executed in real user machines and not in an analysis environment. Recently, Kirat and colleagues³ proposed a bare-metal automated analysis environment, called Bare-Cloud, that does not introduce any in-guest component, which makes the proposed solution more transparent to sophisticated evasion techniques. Similarly, Kharraz and colleagues² proposed UNVEIL, a sandbox that is specifically designed to detect ransomware. UNVEIL creates a fake but enticing user environment for the malicious binary to run by manipulating the return values of some of the system functions that are frequently used by malicious processes.

Insights: While defending against evasive malware is not a new research direction or specific to ransomware attacks, building transparent analysis systems that are indistinguishable from a real host is a critical step to better characterize the behavior of malware including ransomware attacks. In fact, a potential challenge in this area is to assist malware analysts in identifying the environmental checks used by malware as well as performing behavioral analysis by providing fine-grained resource monitoring without impacting the general behavior of the analysis system.

Improving monitoring techniques. Prior work³ discussed the necessity of developing reliable monitoring mechanisms in malware sandboxes to reveal the inner workings of ransomware samples. In fact, it is quite useful to understand how malware authors employ cryptosystems, how a ransomware sample makes user data inaccessible, and whether it is possible to reason about how the encryption key is generated by analyzing the execution traces. For example, UNVEIL² uses a

kernel-level module that monitors the systemwide activities by intervening in the interaction of user-mode processes with the filesystem. The filesystem monitor in UNVEIL has direct access to data buffers involved in I/O requests, giving the system full visibility of nearly all filesystem modifications. The generation of I/O requests happens at the lowest possible layer of the filesystem. Whenever a user thread invokes an I/O API, an I/O request is generated and passed to the filesystem driver. In each malware execution, UNVEIL generates a set of I/O access sequences for the sample. The particular detection criterion used by the system to detect ransomware samples is to identify privileged operations in I/O sequences in each malware run.

More recently, Xu and colleagues⁴ proposed a novel technique, called CryptoHunt, which complements current malware forensics techniques by identifying cryptographic functions in an obfuscated binary. CryptoHunt captures the semantics of possible cryptographic algorithms using bit-precise symbolic execution in a loop. While CryptoHunt can facilitate the identification of ransomware samples in an obfuscated binary and potentially expedite the malware analysis process, it is also desirable to find cryptographic functions in attacks where the malware samples incorporate customized cryptosystems rather than well-known, standard cryptosystems to bypass detection techniques. Recent studies have shown that malware authors utilize home-brewed cryptosystems to evade techniques that infer the functionality of a suspicious binary by looking at API calls imported by the program.

Insights: A potential research direction here is to enhance the cryptographic function identification techniques to be able to detect nonstandard cryptosystems, as

adversaries are increasingly using this technique.^{1,2} As prior research showed, customized cryptosystems may not be perfectly implemented, and recovering the encrypted data can be even easier in a large number of attacks including WannaCry—one of the most recent ransomware attacks. Therefore, a solution that can provide insights on how the key is created and maintained during the attack or provide information on the degree of similarity of the cryptosystem to other cryptosystems that have been previously observed in other ransomware families can assist reverse engineers and security analysts in learning how to retrieve user data without paying a ransom fee.

Developing End-Point Protection Systems

In response to the increasing number of ransomware attacks, a desirable and complementary defense mechanism would be an end-point solution that monitors the operating system resource usage and stops attacks once the ransomware starts destroying user data. This specific area has recently gained attention among security researchers. In the following, we provide the details of the proposed solutions and the security guarantees that they provide.

Software-level support. Over the past few years, several end-point protection tools have been proposed. Scaife and colleagues⁵ proposed CryptoDrop, which is built on the premise that the malicious process aggressively encrypts user files. The authors built their detection model by monitoring how a ransomware sample generates requests to access the filesystem. Very recently, Kolodenker and colleagues⁶ proposed PayBreak, which is agnostic with regard to the filesystem activities of the processes. PayBreak securely stores the

cryptographic encryption keys in a key vault that is used to decrypt affected files after a ransomware attack. In fact, PayBreak intercepts calls to functions that provide cryptographic operations, encrypts symmetric encryption keys, and stores the results in the key vault. After a ransomware attack, the user can decrypt the key vault with his private key and decrypt the files without making any payments.

In another work, Continella and colleagues⁷ proposed ShieldFS, a system that looks for indicators of using cryptographic primitives by scanning the process memory and searching for traces of the block cipher key schedules. While ShieldFS is a significant improvement over the status quo, it would be desirable to complement it with a more generic approach that is also resistant to unknown cryptographic functions.

Lastly, Kharraz and colleagues presented a generic, real-time ransomware protection approach, called Redemption.¹ Unlike ShieldFS, the detection technique is based on two main components. First, an abstract characterization of the behavior of a large class of current ransomware attacks is constructed. A process is labeled as malicious if it exhibits behaviors that match the abstract model. Second, Redemption employs a high-performance, high-integrity mechanism to protect and restore all attacked files by utilizing a transparent data buffer to redirect access requests while tracking the write contents. The authors showed that by augmenting the operating system with the proposed technique, it is possible to stop modern ransomware attacks without changing the semantics of the underlying filesystem's functionality or significantly changing the architecture of the operating system.

Insights: The analysis results in some of the proposed solutions—that is, ShieldFS and Redemption—show that recovering user data after even

an unknown ransomware attack is possible. Furthermore, these techniques illustrate that a well-defined detection model can significantly increase the cost of evasion in these attacks. However, the detection models in these solutions mainly rely on assigning an anomaly score to the processes in the end-user machine. We envision that these detection models can be improved by incorporating reliable machine learning techniques to increase the detection coverage of these solutions, as adversaries will very likely attempt to adapt their attack strategies and bypass some of the features used in the proposed detection models.

Hardware-level support. While software-based solutions, presented in the previous section, have shown that recovering user data is possible in a large number of ransomware attacks, researchers have recently explored the possibility of providing hardware-level guarantees to defend against ransomware attacks. The immediate benefit of a hardware-level anti-ransomware mechanism is its resistance even against kernel-level ransomware attacks, such as WannaCry, where an OS kernel is compromised. Very recently, Huang and colleagues⁸ proposed FlashGuard, a ransomware-tolerant solid-state drive (SSD), which has a firmware-level recovery mechanism that allows quick and effective recovery from cryptographic ransomware. In fact, FlashGuard leverages the out-of-place write mechanism in SSD, which is used to reduce the long erase latency of flash memories. When a page is updated or deleted, the older copy of the page stays in the SSD. FlashGuard slightly modifies the garbage collection mechanism of the SSD to retain the copies of the data that were encrypted during a ransomware attack. This allows FlashGuard to effectively launch data recovery and restore the encrypted files.

Insights: The notion of enabling hardware to provide security guarantees with regard to ransomware attacks is an interesting research direction. However, we envision that many challenges will arise in providing hardware guarantees for real-world deployments without impacting the performance or reliability of SSD technology. Furthermore, because malware authors have significant freedom in adapting their malicious code and attacking the proposed technique (for instance, forcing the filesystem to crash), research challenges will include incorporating higher-layer security properties and defining a hardware–software design approach to address some of the limitations in this area.

In general, ransomware authors, similar to other malware authors, need to develop code that can bypass common detection techniques, successfully reach end users' machines, and launch an attack on those machines. To this end, ransomware authors usually use evasion techniques that are not necessarily different from the ones seen in other classes of malware attacks. Therefore, some of the techniques that have been proposed by security researchers to detect evasive malware are still quite useful in analyzing payloads that deliver ransomware. However, properly defending against ransomware attacks requires solving an additional set of novel intellectual challenges. Overcoming some of these problems requires developing new security mechanisms. For example, new techniques that can reveal how a cryptosystem module—the core function of a ransomware sample—operates during a ransomware attack is quite useful, and can increase the chance of extracting the encryption key, rendering the ransomware attack ineffective. Similarly, the detection

models that can reliably identify ransomware attacks considering the similarities of these attacks compared to a set of benign applications are another avenue that can enhance the detection of anomalous operations. Finally, techniques that can provide data recovery with minimal modification to the hardware and software stack could lead to a better defense against ransomware attacks. ■

References

1. A. Kharraz and E. Kirda, "Redemption: Real-Time Protection against Ransomware at End-Hosts," *Proceedings of the 20th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 17)*, 2017.
2. A. Kharraz et al., "UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware," *25th USENIX Security Symposium*, 2016.
3. D. Kirat, G. Vigna, and C. Kruegel, "BareCloud: Bare-Metal Analysis-Based Evasive Malware Detection," *23rd USENIX Security Symposium*, 2014, pp. 287–301.
4. D. Xu, J. Ming, and D. Wu, "Cryptographic Function Detection in Obfuscated Binaries via Bit-Precise Symbolic Loop Mapping," *IEEE Symposium on Security and Privacy (SP 17)*, 2017, pp. 921–937.
5. N. Scaife, H. Carter, and K. Butler, "CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data," *IEEE International Conference on Distributed Computing Systems (ICDCS 16)*, 2016.
6. E. Kolodinker et al., "Paybreak: Defense against Cryptographic Ransomware," *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (ASIA CCS 17)*, 2017, pp. 599–611.
7. A. Continella et al., "ShieldFS: A Self-Healing, Ransomware-Aware Filesystem," *Proceedings of the 32nd Annual Conference on Computer Security Applications*, 2016, pp. 336–347.
8. J. Huang et al., "Flash-Guard: Leveraging Intrinsic Flash Properties to Defend against Encryption Ransomware," *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 2231–2244.

Amin Kharraz is a research associate at the University of Illinois at Urbana-Champaign. Contact at kharraz@illinois.edu.

William Robertson is an associate professor at Northeastern University. Contact at wkr@ccs.neu.edu.

Engin Kirda is a professor at Northeastern University. Contact at ek@ccs.neu.edu.

myCS
Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>




Call for Software Engineering Award Nominations

Established in memory of Harlan D. Mills to recognize researchers and practitioners who have demonstrated long-standing, sustained, and impactful contributions to software engineering practice and research through the development and application of sound theory. The award consists of a \$3,000 honorarium, plaque, and a possible invited talk during the week of the annual International Conference on Software Engineering (ICSE), co-sponsored by the IEEE Computer Society Technical Council on Software Engineering.

The award nomination requires at least 3 endorsements. Self-nominations are not accepted. Nominees/nominators do not need to be IEEE or IEEE Computer Society members.

Deadline for 2018 Nominations:
1 October 2017

Nomination site:
awards.computer.org

IEEE  computer.society