International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Analysis of Monitored Binary Execution to Identify Data Modifications and Cryptographic Operations

Stebin Sunny[1], Anisha Antu[2]

[1]Final Year M. Tech. (Cyber Security) KMP College of Engineering, Perumbavoor, Kerala, India

[2]Assistant Professor, Department of Computer Science and Engineering, KMP College of Engineering, Perumbavoor, Kerala, India

Abstract: ***Analyzing binary executable is a tedious task in the field of cryptography. In real world internet scenarios all the data is to be encrypted using different cryptographic algorithms for improving security. Malware programs also using this cryptographic techniques to hiding them from being analyzed. We developed a framework to identify the data modifications and to decrypt different types of data even if they are having multi rounded encryption. This mechanism is based on avalanche effect in cryptography that can able to pinpoint exactly where the data modification is happened even nested cryptographic operations is happened and extended to analyze the encryption of files other than documents. Intermediate node analyzing also included to improve the accuracy checking. The same work can be carried out when different types of files having encrypted or multi encrypted with another conditions***.

Keywords: Avalanche Effect, Cryptographic Algorithms, Nested Cryptographic Operations, Malwares, Security Attacks

1. Introduction

Binary sequence analyze become tough to implement because to identify the mechanism used to generate that sequence is the first step of that schema and that is also an obstacle to the analyst. If there is a presence of malware is happened that is also another key problem for the analysts to determine it from the monitored data. When we using cryptographic mechanisms and truly transient cryptographic secrets in the malware data imputes a key restraint to proper malware resolution and impediment. To retrieve those transient secrets from the data, an analyst can able to identify exactly where and when those cryptographic secrets will be in memory and this requires him to particularly pinpoint the boundaries of each rounds of cryptographic operation even if they nested.

In this paper we present a technique to analyze and implement cryptographic operations and can able to locate the boundaries of cryptographic operation if they are nested also and identify the nodes from which data modification happened. This framework is designed based on the avalanche effect which is the covetable property of all cryptographic algorithms like public key cryptographic algorithms, symmetric cryptographic algorithms, hash functions and it says that a small change happened in the input bit may lead to a substantial change in the output bit value. The rest of this paper organized as follows: Section 2 discusses related works. Section 3 discusses proposed system. Section 4 covers the principle of proposed framework and Section 5 concludes the paper.

2. Related Works

Lot of works are carried in earlier related to the above researched area.

2.1. Exploring Multiple Execution Paths for Malware Analysis

In this paper they propose a system that allows to explore multiple execution paths and identify malicious actions that are executed only when some of the certain strategies are met. That enables to automatically extract a more complete view of the program under analysis and identify under which conditions suspicious actions are carried out. Their results demonstrate that many malware samples exploits different behavior depending on input taken from the environment. Thus, by exploring multiple execution paths, able to obtain a more complete picture of their actions [3].

2.2. Automated identification of cryptographic primitives in Binary programs

A security analyst manually pinpoint the cryptographic mechanisms and its usage to identify the malicious movements that may be time consuming. If these process is automated a better analysis is possible. In this method cryptographic primitives identified from binary programs. Execution tracing is the method of analyzing binary executable dynamically to generate a protocol schema that explains the instruction to be executed and the data accessed by the executable. A heuristic method is used to detect the cryptographic primitives. This is having some drawbacks. If the code is not executed then that cannot be analyzed, this a general constraint in dynamic analysis [2].

2.3. Binary Analysis works

Binary analysis has been broadly used in malware detection and analysis. They use different kinds of operations and interfaces to identify the cryptographic operations. Basic limitation of these instruction profiling based approaches is that they are not able to differentiate nested cryptographic operation [4] [5].

3. Proposed System

The proposed framework dynamically intercepts and instruments the instructions of binary executable and gather information from it. We demonstrate the working based on a real world scenario. The main goals are following.

1. Determine if there is any cryptographic operation like encryption, decryption hashing in the execution
2. Determine the location, size and boundary of the input and output by which the crypto operation is performed
3. Determine if there is any data modification is happened in between the sender and receiver, if yes then to pinpoint from which nodes that happened.
4. If any multiple rounds of encryption is happened to determine the modification scenario in the transmission path of encrypted data.
5. Determine the multiple format of files that to be encrypted is able to be decrypted after tracking the secret key even in the multiple rounds of encryption if happened also.
6. Determine the presence of key in each cryptographic operation and to generate the key for the data reached in different nodes

4. Principle of Proposed framework

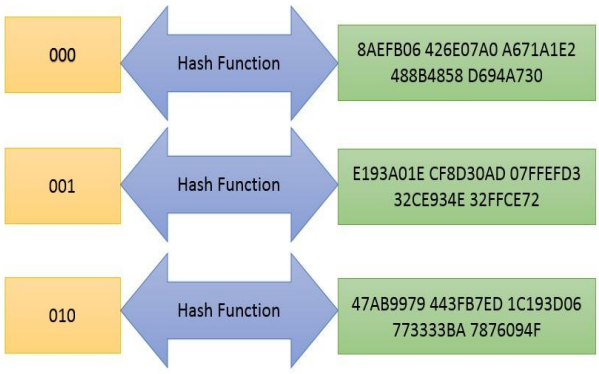

Figure 1: SHA1 exhibits good avalanche effect

The proposed system is designed upon the avalanche effect, which is the desirable property of all cryptographic operations like symmetric algorithms, asymmetric algorithms and hash functions such that when an input is changed slightly (e.g., flipping a single bit) the output changes significantly (e.g., half the output bits flip). In the case of high quality block ciphers a slight change in key or plain text should cause tremendous change in cipher text. Most of the non-cryptographic code never having this avalanche effect therefore this becoming a fairly unique and defining characteristic of all cryptographic operation.

Fig 1 shows the avalanche effect in SHA1 exhibition. When a single bit changed the hash sum becomes completely different. To quantitatively measure the avalanche effect between two buffers , we use C(a,i,b,j)to denote the buffer a's i^{th} contribution rate to j-byte buffer b, which is defined as the portion of first j bytes of buffer b that would be tainted by every byte from the first i bytes of buffer a. C(a,i,b,j) ≈ 100% if and only if there exists the avalanche effect from the first I bytes of buffer a to the j bytes of buffer b.

In this system, the sender encrypt the data (AES Encryption) [6] and also generate the secret key using hash function (SHA1). Instead of sending the data directly to the destination, the data is send to the intermediate user. If the intermediate receiver tries to decrypt the data; a security procedure is added to enhance the security. If the receiver tries to decrypt the data, a secret key will be asked. The secret key will not be send to the intermediate receiver.

If the intermediate receiver tries to modify the encrypted content and send it to the destination, another hash key will be generated for the intermediate receiver based on the avalanche effect. In this way multiple intermediate receivers can be added and the accuracy can be checked. In the destination receiver gets the original data send by the sender without any modification, the secret hash key get matched and the received data can be decrypted Else destination receiver cannot decrypt the data. This is shown in Fig 2 as basic architecture. By taint propagation method the intermediate attacker can be identified and it can be blocked. This scenario is also can able to evaluate if there multiple rounds of encryption is happened.

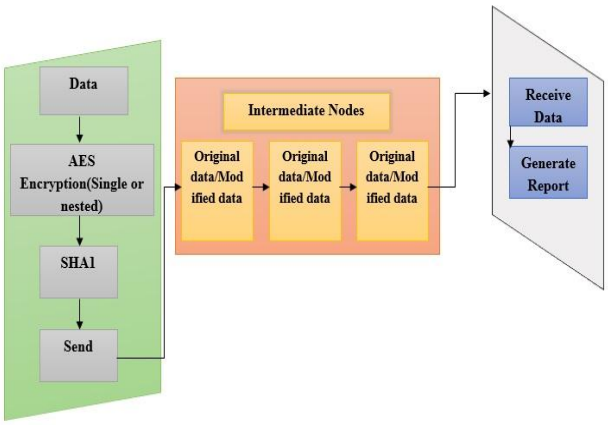

Figure 2: Basic Architecture

While considering the multiple types of file encryption (e.g., image files, music files, .txt files etc) schemas we tested the system with 3-DES encryption along with multiple round of encryption and the efficient decryption of the file after the key generation. The goals that we cited earlier is merely achieved as part of the experiment conducted with different alternatives and that becoming the advantages of this framework.

5. Conclusion

We have presented a framework based on defining characteristic of all cryptographic mechanisms- the avalanche effect, that to be able to detect the data changes in public key cryptography, block cipher, hash operations and pinpoint the boundaries of these mechanisms even if multiple rounds of encryption is carried out. Also included multiple intermediate users to find the accuracy and using the taint propagation method the attacker node can be able to found out. Those mechanisms can able to analyze the files other that

documents and able to generate report based on this dynamic analysis.

References

[1] Xin Li, Xinyuan Wang, and Wentao Chang, "CipherXRay: Exposing Cryptographic Operations and Transient Secrets from Monitored Binary Execution" IEEE transactions on dependable and secure computing, vol. 11, no. 2, march/april 2014
[2] F. Grobert, C. Willems, and T. Holz, "Automated Identification of Cryptographic Primitives in Binary Programs," Proc. 14th Int'l Symp. Recent Advances in Intrusion Detection (RAID '11), Sept. 2011
[3] C.K. Andreas Moser and E. Kirda, "Exploring Multiple Execution Paths for Malware Analysis," Proc. IEEE Symp. Security and Privacy (S&P '07), pp. 231-245, May 2007
[4] T. Pettersson, "Cryptographic Key Recovery from Linux Memory Dumps," Presentation, Chaos Communication Camp. Aug. 2007.
[5] J. Newsome and D. Song, "Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software," Proc. 12th Network and Distributed System Security Sump. (NDSS '05), Feb. 2005.
[6] Intel Advanced Encryption Standard (AES) Instructions Set- Rev 3. http://software.intel.com/en-us/articles/intel-advancedencryption-standard-aes-instructions-set/, 2012

Author Profile

Stebin Sunny received the B.Tech degree in Information Technology from University Of Calicut in 2011 and currently pursuing final year M. Tech degree in Computer Science and Engineering with specialization in Cyber Security from KMP College of Engineering, Perumbavoor.

Anisha Antu received B.Tech in Information Technology and M.E in Computer Science from Mahatma Gandhi University Kottayam in 2011 and Anna University Chennai in 2013 and currently working as assistant professor in KMP College of Engineering Perumbavoor in Computer Science and Engineering Department