I. J. Computer Network and Information Security, **2018, 5, 28-36**
Published Online May 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2018.05.04

Computer Security and Software Watermarking Based on Return-oriented Programming

Ashwag Alrehily and Vijey Thayananthan
Department of Computer Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia
E-mail: ashwagalrehily@yahoo.com and vthayanathan@kau.edu.sa

Received: 18 December 2017; Accepted: 09 February 2018; Published: 08 May 2018

***Abstract*—**Applications of computer security issues are limited to the operating systems and gadgets used within the computers and all other devices integrated with supercomputers. With the growing number of users, software developers face the software piracy which could affect the computer systems. Currently, the major problem for computers in the different industries is software piracy. Despite many computer security techniques, we have proposed a software watermark design based on return-oriented programming (ROP). Here, the new design of the software watermark is considered as a method in which secure Hash algorithm plays an important role to enhance the performance of the computer security. In this method, we focused on gadgets analysis gadgets categories and a large number of gadgets. In this analysis, we selected Sjeng benchmark and ROP with different approaches. As a theoretical result, resilience and stealthy are compared with existing results. This approach can be useful to improve the application of the computer security laws with legal procedures depended on the proposed computer security algorithms and analysis.

***Index Terms*—**Computer security, Software watermark, Gadgets, Return-oriented programming and Secure Hash Algorithm.

I. Introduction

Computer security problems have many symptoms which include program lock-ups, slow PC performance, system freezes, etc. All of these cases may depend on the software piracy which creates many other security issues such as the startup and shut down problems, installation errors, and hardware failure. It is essential to investigate and eliminate the software piracy using appropriate techniques related to software watermarking.

Although several algorithms have been proposed for protecting computers and their applications, appropriate solutions for future computer security will be inevitable. Recent and computer attacks motivate us to design the suitable solution involved with software watermark based on ROP. In order to determine the threats, we focus on these four (interception, interruption, modification, and fabrication) actions.

According to [1], data execution protection (DEP), address space layout randomization (ASLR), and stack canaries are the three main mitigation strategies for preventing software attacks. In their research, ROP exploit bypasses this protection by chaining together gadgets from executable sections of memory to either produce the desired effect entirely from gadgets or to change the protections on memory containing the shellcode to make this area executable and then pass execution to the newly unprotected shellcode's entry point.

Considering computer security, the dynamic watermark is more reliable to analyze and execute the future of the computer security. Further, it secures solution because the hidden message retrieved by running and examining the specific behavior of specific path of the watermarked program. Reliable depends on the performance overhead which means that the existing watermark program and the original program should have same working conditions such as running time [2]. Resilience and credibility are also measurements in the computer security which allow us to measures the resistance against specific attempts at discovery or deletion and defines how accurately the watermark can be retrieved respectively [3]. Despite many techniques used in the computer security, the anti-piracy scheme is one of the efficient techniques, which is illustrated below.

- Obfuscation is one of the examples used for the anti-piracy technique. It transforms the target code into a more complex form. Further, it makes harder to identify the software duplication.
- Tamper-proofing prevents the attacks related to the modified or altered programs. Although many mechanisms are available, following mechanisms help us to build the efficient anti-piracy techniques. They are checksums, guards, etc.
- The anti-piracy can be configured using above mechanism through the software. Thus, the user is logged onto the Internet and computer send its serial number secretly to the software company which supports to develop the computer and its operating systems.

Although some contributions are same as our paper [4], we have added few more points such as modified theory algorithm and results after MSc thesis completed. We

Copyright © 2018 MECS

focused on gadgets analysis investigating with gadgets categories and a large number of gadgets. In this analysis, we selected Sjeng benchmark and ROP with different conditions. Further, proposed approach will allow future researchers to develop computer security laws and protocols.

The paper is designed as follow; Section 2 introduces the theoretical background of the research related to computer security and software watermarking. Section 3 provides the design of the proposed work with necessary steps and approaches. Section 4 covers performance evaluation which includes implementation, result, and discussion. Section 5 concludes with the summary of the proposed and future work.

II. Theoretical Backgrounds

In this section, relevant theoretical information and necessary backgrounds of our research are considered. In this research, we need to know following terms they are such as software watermark, techniques, related work, etc. In the computer security, software watermarking, the software piracy and use of ROP allow the researchers to improve the computer security policies. Despite many watermarking algorithms influenced by the computer security policies, the piracy rates in different countries depend on the copyright policies which includes the efficient computer security algorithms and policies. This paper shows the improvements, efficiencies, and benefits of the watermarking technique with our proposed steps and algorithms.

A. Software Watermark

Computer security solutions may be used as policies within computer systems which provide huge supports for protecting software privacy influenced by personal and sensitive data in various industries. Further, computer security solutions will improve the arbitration during the unnecessary dispute between the users involved with computer processors or systems. Deriving computer security solutions from the software watermark technique are not only important for protecting the privacy and personal data but also secure the gadgets used within the computers. Further, billions of computers are connected to the world with reasonable computer security which protects the computer operating systems and personal information. Still, illegal activities such as blocking, copying, etc. are spoiling the computer systems. So, we need the proper computer security solutions derived from existing computer security techniques rather than simple and fundamental computer security policies. Although we have mentioned the computer security techniques in this paper, we are planning to introduce computer security solutions in the future security developments.

Applications of computer security issues influences with computer software which is being affected by software piracy. When computer software is involved with illegal and unlicensed products, daily businesses in enterprises are blocked. Software piracy is illegal to use, copying, selling and distribution of software, regardless of copyright laws or license agreement. Due to software piracy, every country lost millions of dollars every year. Business Software Alliance (BSA) released a survey in 2016 that measure the rate of the unlicensed software and commercial value of the unlicensed software in many countries like Central and Eastern Europe, Asia Pacific, Middle East and Africa, Latin America, North America and Western Europe [5]. Fig. 1: shows the comparison bar chart of piracy rate.

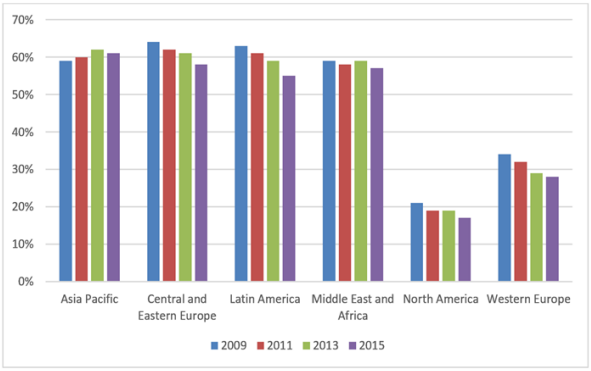

Fig.1. The Piracy Rate in Percentage.

Every year the software vendors lose millions of dollars because of the software piracy. The total of money lost is $52.2 billion. Based on the BAS the total percentage of the pricey rate for 2013 is 43%, and it decreases in 2015 to 39%. The software protection becomes a hot research topic and crucial issue in computer industry due to the high rate of pricey. Despite the basic computer security solutions, there are many techniques to protect the software, and one of them is software watermark.

B. What is Software Watermark?

The enhancement of computer security is achievable through the technology that embeds the copyright instructions and user identity information into the program called software watermark. Computer security solutions derived from software watermarking based on (ROP) protect the security algorithm developers who work in the computer organization. Software watermark is a process of embedding a secret message in the source code of the program. The secret message contains information about the real owner (copyright owner) of the software such as author, publisher, and owner. Thus, an embedded watermark in software should not affect the flow of the program or make any redundant space which will change the program high-quality. In order to apply a watermark to any software program, there are two important processes, they are the embedding watermark into the software and extracting the watermark from the software.

C. Techniques Involved in Software Watermark

Although there are many techniques available to improve the computer security using watermark, we focus on two types of software watermark techniques.

Copyright © 2018 MECS

Firstly, the static software watermark technique is embedded with the watermark in the target application that is executable as the text section and initialized data. Secondly, the dynamic software watermark technique is embedded with the watermark during the program execution. Here, static or dynamic data that gives the program a new path to execute contains the watermark.

In order to define the computer security, software watermark algorithm can be classified based on the watermark extraction, implementation, goals, and execution. Software watermarking can be robust or fragile; the attackers cannot eliminate the robust watermark. The fragile watermark can be easily destroyed by the attackers. Also, software watermark can be organized "blind" or "informed" based on the watermark extraction. Here, it is required to have the original program and watermark as inputs to extract the watermark. The blind software watermark is the opposite of the informed software watermark which does not need the original program to extract the watermark.

Although there are selected attacks controlled by the computer security solutions, there are different types of attacks that target the software watermark such as the following:

- Rewrite attacks: In this type of software watermark attacks, the attacker can write the same program from scratch.
- Additive attacks: The attacker can add his own watermark to watermarked software. This type of attack confuses the software watermark recognizer to identify which one of the watermarks is the original one.
- Distortive attacks: The attacker attempts to destroy the existing watermark by applying semantics-preserving transformations such as code obfuscations. This type of attack work on the software watermark that uses program syntax in the watermark.
- Subtractive attacks: The attacker removes part of program code that builds watermark. The attacker makes sure that the removing part did not affect program execution.

D. Related Work

Static software watermark has a lot of different algorithms as given in [6] where authors have proposed an algorithm which uses the equation reordering. Here, the developers ensure that the results of the equation should be same when watermark algorithm reorders the operands of the equation and hide the watermark data. The algorithm selects the equations to be ordered based on some criteria as described. In this criteria, they must be mathematical equations because they are in every program and some of the arithmetic operations are symmetric such as addition and multiplication and changing the order of the arithmetic operations will not affect the equations result. The order of equation operands is changed per the watermark information influenced by the computer security solutions.

The watermark embedding process will be after a sequence of division and multiplication. The watermark extraction process starts some operation by using the secret key from the copyright owner. However, the evaluation of the proposed work provides the robustness of the watermark. As far as the papers [6 & 7] are concerned, it provides the better performance without affecting the code length or the speed of the program.

The existing watermark based on the equation of resort algorithm has in fact easy attacked by using the random re-sequencing technology. Despite this attack, [8] solved this problem by proposing a new software watermark that decomposed the watermark using Chinese remainder theorem. Although the diversity of authentication is part of computer security issues, the watermark authentication center provides the watermark information which is an integer number. The watermark will be decomposing using Chinese remainder theorem and hide in the source code without any additional codes.

Table 1. Summary of Related Work

Ref	Software watermark type	Watermark embed method	Results
[9]	Dynamic	Graph-based watermark	Create first dynamic software watermark
[10]	Static	code obfuscation	Prove robustness of the watermarking.
[11]	Static	Equation reordering	Algorithm has limited capacity for hiding the information
[12]	Dynamic	Reorder code functions	The algorithm cannot be attacked through the Additive, distractive attacks
[13]	Dynamic	Hash encryption and decryption functions	The algorithm resisted subtractive, distortion and additive attacks
[14]	Dynamic	Shamir threshold scheme & branch based watermark.	Prove robustness of the proposed algorithm
[15]	Static	Reorder Coefficients of equation	Algorithm can be attacked easily by using the random re-sequencing technology
[16]	Static	Code obfuscation	Efficient robustness and increase the running time
[17]	Static	Chinese remainder theorem.	Prove robustness of the watermark. The algorithm did not affect the code length or the speed of the program, and it has better performance.

Java is a well-known programming language, and any code written in the Java language face more threads more than any other languages because of the bytecode. The bytecode presents thorough information about the code that leads the malicious program users to decompose the code into reusable class files also decompiled into the source code. Table 1 shows the quick results for existing watermark embedded methods and corresponding

software watermark types.

According to [18], an attacker may induce the random behavior in the software program without knowing computer security issues. Further, without injecting any piece of code, an attacker can divert the programs' control flow. A ROP chains all gadgets together and makes short instruction sequences that already present in a program's address space, each of which ends in a "return" instruction.

A robust software watermarking influenced with obfuscated interpretation technique can prevent the various attacks such as collisions [19]. The obfuscated interpretation technique allows us to hide the functionality of a given program and to provide an alternative way to modify the code.

According to [20], identity management and protection of user privacy allow us to improve the security issues. Identity management improves the algorithms used in the computer security. Although this research provides many solutions, authors introduced the International Mobile Subscriber Identifier (IMSI) for improving the solutions. In this approach, introducing of pseudonyms that replace the user permanent identifier used for identification. The IMSI scheme can also be useful to implement the new computer security algorithms with watermarking.

Despite the watermarking used in the computer security, article [21] presents various semi-fragile watermarking algorithm. In this article, authors have studied the applications of watermarking through the literature survey which helps us to analyze the best watermarking algorithm. In order to improve the security issues, authors have employed the semi-fragile watermarking techniques.

According to [22], we have studied various copyright protection techniques which are useful for improving the computer security algorithms. Although this research paper presents a new technique for copyright protection, researchers have decided to use existing algorithms with this new technique called as an integer wavelet transform (IWT). Here, singular value decomposition (SVD) and Arnold transform are used for protecting images. Further, authors improve the security issues using Steganography approach which not only hides the secret information but also improves the copyright protections. Regarding the security solution, the effects of the geometric attacks are also considered.

Based on paper [23], privacy issues provides huge supports to improve the computer security. Protecting software privacy influenced by personal and sensitive data is one of the main problems in the potential industries. Although cryptography and steganography do not provide a direct and ultimate solution for privacy preservation in open systems, authors proposed a novel threefold model which allows us to improve the computer security algorithms.

III. Design of The Proposed Scheme

In this proposed work, we have developed watermark program using ROP technique, and we add our proposed ROP trigger. Despite the computer security during the design procedure, this section explains in detail about the methodology of our proposed work.

A. Embed Watermark into ihe Software

In the proposed design, Fig. 2 illustrate the procedure of software watermark embedding. In this procedure, embedding can be represented by (1) which takes necessary inputs mentioned in the following function describes the watermark embedding.

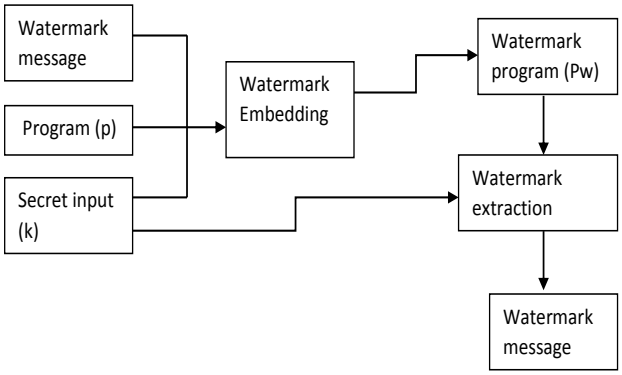

Fig.2. Software Watermark Process.

$$\text{Embed}(\text{P}, \text{W}, \text{k}) \rightarrow P_w \quad (1)$$

In this function (1), P, W, k, and P_w are representing the program, watermark, secret input and watermarked program respectively.

B. Extract Watermark from Software

To display the copyright of the software owner the watermark W can be extracted from the watermarked program Pw by watermark extractor and secret input k. The following function describes the software watermark extraction.

$$\text{Extract}(P_w, \text{k}) \rightarrow \text{W} \quad (2)$$

Although extraction is the main target in the computer security, hackers try to extract illegally.

C. Watermark Program based on ROP

The ROP was used on many platforms such as x86 architecture, ARM, SPARC and it is proven that it can be used in iOS applications [24-26]. Although these platforms are designed with basic computer security procedures, ROP based on a return to Libc attack uses functions from the Libc library. The ROP code formats in a selected platform that cannot be run directly. Details of ROP such as own instructions which create the unexpected execution path can be found in [27].

Watermark program based on ROP built by ROP gadgets performs four main steps. According to [28], our watermark message "007" and string S are used with same assumptions. In order to embed the watermark, we selected three important gadgets, they are <pop ecx; ret>, <pop eax; ret> and <mov [ecx], eax; ret>.

In our proposed work, main steps are finding watermarking gadgets, ROP payload, payload chain and triggering ROP via function pointer overwriting. In [29], we have studied and collected all these four main steps in details.

To locate gadgets, there are a lot of existing techniques used to search for gadgets. Here, we use Immunity Debugger program with Mona.py tool to help to search for the gadget; the following subsection describes the Immunity Debugger program and Mona.py:

- Immunity Debugger: The Immunity Debugger program is a new approach to analyze program malware, write exploits and ability to reverse engineer binary files. The Immunity debugger written in Python, it has a powerful simple, and understandable graphic user interface (GUI) also provide a command line to enable the user to write any command needed.
- Mona.py tool: It is a powerful tool designed as a py command file, py command is command base class build to allow the programmer to create executable command in python programs.

The Immunity Debugger program has the ability to add any executable command written in the py command file; thus, Mona.py used as plugin concept for Immunity Debugger. The purpose of Mona.py is to find gadgets and create simple gadgets chain. Mona.py search for gadgets after entering ROP command by the user. The ROP command in Mona.py generates four text files which are the following:

a) ROP gadget (txt): it includes all the available gadgets in a specific module such as program and libraries based on command enter by the user.
b) ROP gadget suggestion (txt): it builds a list of suggested gadgets after classifying the ROP gadgets.
c) ROP chain (txt): it generates simple chain functions in different programming languages.
d) ROP stack pivot (txt): list of all the gadgets that can change Extended Stack Pointer (ESP) register value of stack to hold the value of fake stack.

In our method, we use mona.py to search for all available gadgets in Immunity Debugger, and our search took only 15 seconds to search for available gadgets.

IV. Performance Evaluation

In order to evaluate our approach, we have selected Sjeng as a benchmark which allows us to prove that our design works correctly. Thus, Fig. 3 shows the flowchart of Sjeng program which contains the detailed instructions and rules of a Chess game. In each user's legal move, the program checks the move which is the possibility of being the secret input to extract the watermark. This Sjeng program examines the extent to which legal move based on the computer security can allow the users to play the chess game properly.

Although many versions of ROP play an important role in the computer security research, ROP without return is introduced in [30]. Despite many algorithms and mechanisms introduced for developing the best software watermarking, paper [31] provides the efficient approach and mechanism of a watermarking scheme to protect the copyrights of the software.

Like any other software watermark, the performance of the proposed algorithm depends on the parameters. Here, input parameters will be program source code and the output will be the watermarked program. ROP based software watermark consists of:

1) Program source code:

Use as input to convert it to the watermarked program through code rewrite.

2) ROP execution:

ROP code consist of two parts:

I. Gadgets are small pieces of instruction that located somewhere in the code, and they should end with an indirect call or jump or return.
II. The payload is string bit that contains the above gadgets addresses.

ROP execution needs gadgets and payload to work. Thus, the gadgets scanner is responsible for creating gadgets from system libraries and also generate ROP payload that follows the gadgets into the watermark code.

3) Code rewrite:

Take the program source code and modify it to embed the watermark code. Thus, it will take the gadget, payload, and ROP trigger.

4) Watermark program:

The final output that contains the program source plus the watermark code.

In order to achieve the better performance, above steps help us to evaluate the proposed algorithm which is the ROP based software watermark.

In the performance evaluation of computer security, proposed algorithm and software development of the watermark provide the main roles. Thus, proposed ROP based watermarking implemented using a combination of C++ and C programming language. To implement this combination, the Microsoft Visual Studio 2015 has been used as the main template for testing the proposed work. It provides a lot of features such as creating software regardless of the software size and complexity, ability to set environment only once, fixing error easily and flexibility to analyze code quality and performance.

Copyright © 2018 MECS

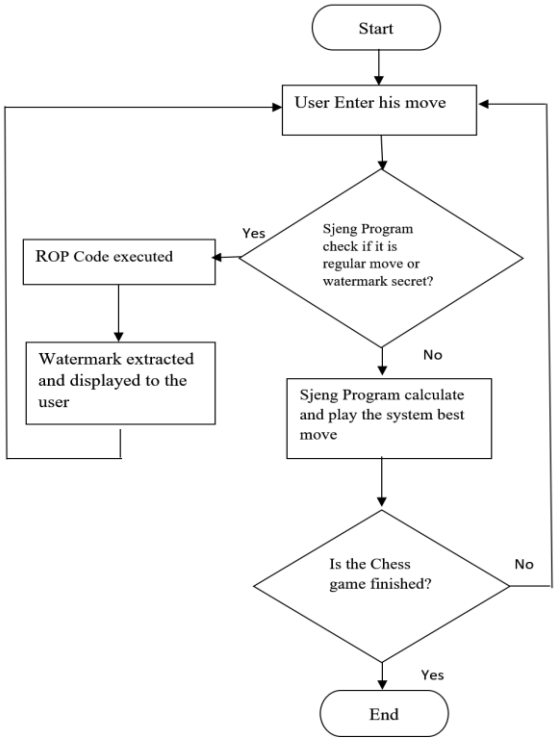

Fig.3. Sjeng Program Flow Chart

A. Results

Fig. 4 shows the basic analysis of processing time during the embedding stage which mainly focuses on encryption. Although software watermark is a process of embedding a secret message in the source code of the program, encryption scheme (Hash) used in this research with the proposed approach and specific benchmark, takes some time for processing. According to this analysis, our proposed takes less time than the original approach when we increase the data sizes. In this processing, embedding and retrieving messages depend on the data sizes which influence with heap sizes. In this analysis, assume that heap size is almost constant when our proposed ROP is employed in this research. Here, the analysis may allow us to improve the computer security using best processing time with best encryption scheme and, selected types and format of gadgets.

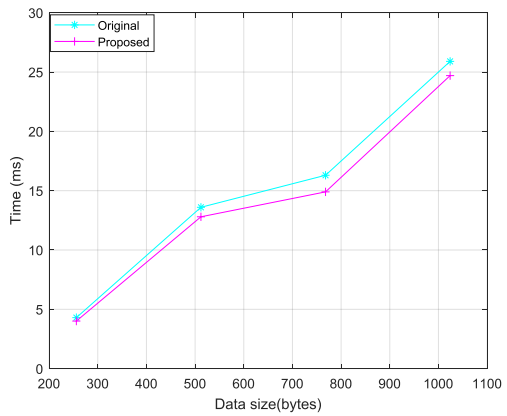

Fig.4. Average Processing Time in Sjeng Benchmark.

Table 2 shows the relevant information of the proposed and original watermarking schemes.

As far as the future computer security is concerned, following information will be useful to develop the computer security protocols.

- Data rate: Number of bits, which influences with watermark
- Embedding overhead: Depend on the processing time of the watermarked application.
- Resilience against manual attacks (stealth): Comparison of the statistical properties of the watermarked and original program
- Resilience against transformations: It influences with semantics-preserving

Regarding the watermarks based on ROP used by the previous researchers, proposed approach survive the transformations such as code optimization and code obfuscation depended on the categorized ROP gadgets count. The overhead of these transformations can be defined according to the applications and the watermark which should not be recognized when enough transformations have been applied to the applications.

Table 2. Summary of Proposed and Original Watermarking

Sjeng benchmark	Our ROP-based watermarking	Original ROP-based watermarking
Runtime Overhead (ms)	18	19.4
Additional heap space required	164	Between 156 or 188
Time to embed the watermark (second)	20 sec	More than 20 seconds
Time to retrieve the watermark (minute)	Between 3 and 5 minutes	More than 5 minutes

B. ROP Gadgets Analysis

As far as the categories of the gadgets are concerned, counting and analyzing gadgets using few algorithms are explained in [32]. With the use of a ROP gadget finding tool, we can analyze the details of the potential and ROP gadgets. In this analysis, obfuscated and un-obfuscated binary allow us to compare the list of all used gadgets in the proposed approach.

As sown in Fig. 5, the categories of gadgets which allow researchers to develop the computer security rules and protocols, are analyzed for counting used gadgets. Here, we used memory, arithmetic (Arth), logic, control and other instructions as gadgets.

Regarding the further analysis of Fig. 5, assume that jump and return instructions are used as other categories of gadgets. Despite many other gadgets and solutions for ROP attacks, few cases of ROP attacks based on indirect jumps may not be possible with some algorithms. In this situations, instructions such as return can be considered during the implementation. Further, proposed ROP can help us to detect the corrupted return address when the return address is moved by the call instruction. In order to

Copyright © 2018 MECS

decide whether the jump is legal or not during the runtime, details of the program's structure are important. Although the characteristics of a ROP attack based on the indirect jump are unique, ordinary programs may not be affected frequently. Generally, return (gadget) appears more frequently than the jump (gadget). Memory regions may not be executable when ROP attacks allow the adversaries to push many different return instructions onto the stack. Although many gadgets are available in the computer security, we have analyzed some frequently using gadgets.

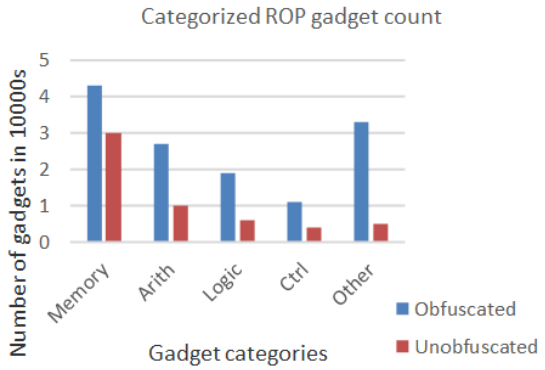

Fig.5. Number of Gadgets Involved in Sjeng

V. Conclusions

Despite the computer security issues lagging behind the latest security advantages, new privacy and data protection approaches based on watermark technique still play an effective role in ensuring that computer users do not damage their data. Applications of future computer security are investigated and studied with existing techniques which allow us to design a new software watermarking technique using ROP functions. Computer security issues are always depending on efficient and legal security solutions which protect not only the components of the computers but also all operating systems installed to configure the computer systems. In this new approach, we employed the ROP with complex trigger ROP function which uses the SHA256. Although we studied the theoretical analysis of the many libraries, gadgets, benchmark involved with computer security, we selected the Sjeng benchmark to prove the new design. In the trigger function, we used hash function SHA256 to compare between the secret input and user entered input. Using this benchmark, we completed the design successfully with the basic results focused on resilience, stealthy, etc.

In the future work, the proposed work will be expanded to test and verify other benchmarks. Updating computer security solutions and minimizing the number of gadgets reduces the ROP attacks. It may be automated to improve the future super/computer security where formats and types of the future libraries/gadgets depend on the future technologies. The future work of software watermark using ROP is to improve the computer security by embedding a hard watermark into the program such as adding watermark in the object of class instead of a simple string. Also, improve the efficiency of software watermark by distributing ROP payload among the program to make it harder for an attacker to recognize it. According to [33], privacy by design is evolving with upgradable operating systems, computer systems, and technologies. Here, the privacy of the personal and sensitive data should be protected by computer security laws.

Acknowledgment

This paper consists part of my Master's Thesis at King Abdulaziz University. I hereby take the opportunity to thank my supervisor Dr. Vijey Thayananthan for his essential guidance and his kind assistance to this attempt.

References

[1] Creech, Gideon. "New approach to return-oriented programming exploitation mitigation." Information Security Journal: A Global Perspective (2017): 1-16.

[2] Tang, Zhanyong, and Dingyi Fang. "A tamper-proof software watermark using code encryption." In Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on, pp. 156-160. IEEE, 2011.

[3] Ma, Haoyu, Kangjie Lu, Xinjie Ma, Haining Zhang, Chunfu Jia, and Debin Gao. "Software Watermarking using Return-Oriented Programming." In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, pp. 369-380. ACM, 2015.

[4] Alrehily, Ashwag, and Vijey Thayananthan. "Software Watermarking based on Return-Oriented Programming for Computer Security." International Journal of Computer Applications 166, no. 8 (2017).

[5] BSA, Seizing Opportunity Through License Compliance. BSA, Software Alliance, 2016.

[6] Chionis, Ioannis, Maria Chroni, and Stavros D. Nikolopoulos. "WaterRPG: A Graph-based Dynamic Watermarking Model for Software Protection." arXiv preprint arXiv:1403.6658 (2014).

[7] Sha, Zonglu, Hua Jiang, and Aicheng Xuan. "Software watermarking algorithm by coefficients of the equation." In Genetic and Evolutionary Computing, 2009. WGEC'09. 3rd International Conference on, pp. 410-413. IEEE, 2009.

[8] Jiang, Hua, Hanlei He, and Xin Wang. "Software watermark algorithm based on Chinese remainder theorem." In Conference Anthology, IEEE, pp. 1-4. IEEE, 2013.

[9] Collberg, Christian, and Clark Thomborson. "Software Watermarking: Models and dynamic embeddings." In Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of programming languages, pp. 311-324. ACM, 1999.

[10] Tu, Ronghui, Feiyuan Wang, Jiying Zhao, and Abdulmotaleb El Saddik. "Copyright Protection of Web Applications through Watermarking." In Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on, vol. 3, pp. 78-82. IEEE, 2006.

[11] Shirali-Shahreza, Mohammad, and Sajad Shirali-Shahreza. "Software watermarking by equation reordering." In Information and Communication Technologies: From Theory to Applications, 2008.

ICTTA 2008. 3rd International Conference on, pp. 1-4. IEEE, 2008.

[12] Gupta, Gaurav, and Josef Pieprzyk. "Source code watermarking based on function dependency oriented sequencing." In Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP'08 International Conference on, pp. 965-968. IEEE, 2008.

[13] Zhang, Xuesong, Fengling He, and Wanli Zuo. "Hash function based software watermarking." In Advanced Software Engineering and Its Applications, 2008. ASEA 2008, pp. 95-98. IEEE, 2008.

[14] Jian-qi, Zhu, Liu Yan-heng, Yin Ke, and Yin Ke-xin. "A Robust Dynamic Watermarking Scheme based on STBDW." In Computer Science and Information Engineering, 2009 WRI World Congress on, vol. 7, pp. 602-606. IEEE, 2009.

[15] Pervez, Zeeshan, Yasir Mahmood, and Hafiz Farooq Ahmad. "Semblance based dis-seminated software watermarking algorithm." In Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on, pp. 1-4. IEEE, 2008.

[16] Chen, Liang, and Chaoquan Zhang. "A novel algorithm for. NET programs are watermarking based on obfuscation." In Instrumentation & Measurement, Sensor Network and Automation (IMSNA), 2012 International Symposium on, vol. 2, pp. 583-586. IEEE, 2012.

[17] JIANG Hua SHA Zong-lu XUAN Ai-cheng Software watermarking algorithm based on inverse number of expression [J] Journal of Computer Applications 2009 29(12) 3188-3190

[18] Roemer, Ryan, Erik Buchanan, Hovav Shacham, and Stefan Savage. "Re-turn-oriented programming: Systems, languages, and applications." ACM Transactions on Information and System Security (TISSEC) 15, no. 1 (2012): 2.

[19] Zeng, Ying, Fenlin Liu, Xiangyang Luo, and Chunfang Yang. "Robust software watermarking scheme based on obfuscated interpretation." In Multimedia Information Networking and Security (MINES), 2010 International Conference on, pp. 671-675. IEEE, 2010.

[20] Muthana, Abdulrahman A., and Mamoon M. Saeed. "Analysis of User Identity Privacy in LTE and Proposed Solution." International Journal of Computer Network and Information Security 9, no. 1 (2017): 54.

[21] Tiwari, Archana, and Manisha Sharma. "Semi-fragile Watermarking Schemes for Image Authentication-A Survey." International Journal of Computer Network and Information Security 4, no. 2 (2012): 43.

[22] Singh, Siddharth, and Tanveer J. Siddiqui. "Copyright Protection for Digital Images using Singular Value Decomposition and Integer Wavelet Transform." International Journal of Computer Network and Information Security 8, no. 4 (2016): 14.

[23] Lone, Auqib Hamid, and Moin Uddin. "A Novel Scheme for Image Authentication and Secret Data Sharing." International Journal of Computer Network and Information Security 8, no. 9 (2016): 10.

[24] Anley, Chris, John Heasman, Felix Lindner, and Gerardo Richarte. The shellcoder's handbook: discovering and exploiting security holes. John Wiley & Sons, 2011.

[25] Buchanan, Erik, Ryan Roemer, Hovav Shacham, and Stefan Savage. "When good instructions go bad: Generalizing return-oriented programming to RISC." In Proceedings of the 15th ACM conference on Computer and communications security, pp. 27-38. ACM, 2008.

[26] Shacham, Hovav. "The geometry of innocent flesh on the bone: Return-into-libc without function calls (on the x86)." In Proceedings of the 14th ACM conference on Computer and communications security, pp. 552-561. ACM, 2007.

[27] Palsberg, Jens, Sowmya Krishnaswamy, Minseok Kwon, Di Ma, Qiuyun Shao, and Yi Zhang. "Experience with software watermarking." In Computer Security Applications, 2000. ACSAC'00. 16th Annual Conference, pp. 308-316. IEEE, 2000.

[28] Collberg, Christian, Stephen Kobourov, Edward Carter, and Clark Thomborson. "Error-correcting graphs for software watermarking." In Proceedings of the 29th Workshop on Graph-Theoretic Concepts in Computer Science, pp. 156-167. 2003.

[29] Ashwag Alrehily and Vijey Thayananthan, "Software Watermarking based on Re-turn-Oriented Programming for Computer Security," International Journal of Computer Applications, Volume 166 – No.8, pp. 21-28, May 2017.

[30] Checkoway, Stephen, Lucas Davi, Alexandra Dmitrienko, Ahmad-Reza Sadeghi, Hovav Shacham, and Marcel Winandy. "Return-oriented programming without returns." In Proceedings of the 17th ACM conference on Computer and communications security, pp. 559-572. ACM, 2010.

[31] Zhu, William, Clark Thomborson, and Fei-Yue Wang. "A survey of software watermarking." In International Conference on Intelligence and Security Informatics, pp. 454-458. Springer Berlin Heidelberg, 2005.

[32] Joshi, Harshvardhan P., Aravindhan Dhanasekaran, and Rudra Dutta. "Impact of software obfuscation on susceptibility to Return-Oriented Programming attacks." In Sarnoff Symposium, 2015 36th IEEE, pp. 161-166. IEEE, 2015.

[33] Anna Romanou, The necessity of the implementation of Privacy by Design in sectors where data protection concerns arise, Computer law & Security Review: The International Journal of Technology law and Practice (2017), doi: 10.1016/j.clsr.2017.05.021.

Authors' Profiles

Ashwag Alrehily received her Master's degree in Computer Science from King Abdulaziz University, KSA. She has been an IT application specialist at Saudi Airlines, Jeddah, KSA since 2012.

Vijey Thayananthan received his Ph.D. degree in Engineering (Communication Engineering) from Lancaster University in 1998, UK. He is currently an Associate Professor at King Abdulaziz University, KSA. His research interests include wireless communication algorithm design and analysis, security management of big data, applied cryptography, computer security, cybersecurity and wireless sensor network.

How to cite this paper: Ashwag Alrehily, Vijey Thayananthan,"Computer Security and Software Watermarking Based on Return-oriented Programming", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.5, pp.28-36, 2018.DOI: 10.5815/ijcnis.2018.05.04

Copyright © 2018 MECS