Simple and Naïve Techniques for Backdoor Elimination in RCA.

Usha Sunil Gound
Student
Dept. of Computer Engg.
DGOI, COE, Daund, Pune
Email:goundusha9@gmail.com

Shweta Satish kadam
Student
Dept. of Computer Engg.
DGOI, COE, Daund, Pune
Email: ashwet.kdm@gmail.com

Prof. Tanaji A.Dhaigude
Assistant Professor
Dept. of Computer Engg.
DGOI, COE, Daund, Pune
Email:tanajidhaigude@gmail.com

Avinash Shivaji Gaikwad
Student
Dept. of E & TC Engg.
SCOE, Vadgaon (Bk),Pune
Email:avinash.gaikwad12@gmail.com

Abstract—World is rapidly going to be digitalized and security is major challenge in digital world. Digital data should be protected against bad natured users. Number of system has come up with different solutions, some of them adopting response computation authentication. In Response Computation Authentication System, system calculates users response and if it matches with system expected value then system authenticates user. Response computation system authenticates user independently. In RCA bad natured developer have plant backdoor to avoid regular authentication procedure. Developer can add some delicate vulnerability in source code or can use some insufficient cryptographic algorithm to plant backdoor. Because of insufficient cryptographic algorithm it is very difficult to detect and eliminate backdoor in RCA. Here proposed system provides solution to check whether any system contain any backdoor or not? Login module is divided into number of components and component having simple logic are checked by code review and component which contains cryptography are sandboxed.
Keywords-: *Backdoor,Response Computation,Sandboxing.*

*****

I. INTRODUCTION

Backdoor is a method to avoid regular authentication steps to gain unauthorized access of system and getting access of plaintext but still remain undetected. In Digital System each user has unique id and password to get access of system for security purpose but many bad natured developers keep backdoor entries to gain access of system by hiding source code so that it breaks confidentiality of system. Many cases found where some developers intentionally kept the backdoor into the authentication module to access the system in unauthorized way. Using this hidden password attacker can obtain important information of clients. By taking important information of clients, attacker changes properties of client's credentials and can take access of client credentials and shipping addresses. The authentication system is divided into two categories based on how they interact with user: First, after a user responds challenge to the login system either replaces the user's response with an expected response calculated from known privileges of the user, or second uses the user's response as a complex authentication calculation, which is based on public key cryptography. In this paper I have concentrated on Response Computation Authentication (RCA) which is mostly used in Authentication System. There are two types of RCA backdoor first type is Bypassing response comparison (T1 Type backdoor). Second type is controlling computation of expected response (T2 type backdoor). In T1 type backdoor [1] attacker can gain access of system without comparison of response and response. In T1 type backdoor response computation function does not calculate response at any moment. Neglecting authentication is usually prompted when some special conditions are fulfilled. there are three types of trigger conditions are as U triggered backdoor- some special inputs are used to trigger a hidden logic or intended vulnerabilities in L() to bypass the Response comparison. G Triggered backdoor [1]. In this G-triggered backdoor Global states can used to trigger the hidden logic and neglect the authentication

Procedure. MAC addresses, specific system timing can also be used. For example, between 2:00pm and 2:01pm, Response computation function returns TRUE disregarding of reply of users response. I-triggered backdoor [1]. In I-triggered backdoor Internal states can be used to trigger hidden logic to

Neglect authentication procedure. For example, response computation function can record the frequency of failure of login attempt and return TRUE if login failure frequency falls into a specific range. For example failure frequency is added three times in source code and recorded failure frequency is matched then L () function returns true and attacker can gain access of system. In T2 type backdoor [1] response comparison is not bypassed but response Computation function is affected by attacker. Attacker can plant backdoor to guess response which is calculated by response computation function. Actually response computation function should depend on challenge user's password. Based on how response computation function gets affected by attacker there are two sub types of T2 backdoor. Type T2x [1]: Response computation function is independent of information like challenge and password. Type T2y [1]: this type of backdoor is based on collision. In T2x type of backdoor the attacker can hide his own pair of username and password. When this username and password is given as a input then this pair can be used for response computation and due to this response can be easily guessed by attacker. If response computation functions produce different output for different password then attacker can login successfully by guessing right password.

II. LITERATURE SURVEY

In this section we have reviewed the papers given in the references section.

1. In [1] authentication model is divided into several steps. Simple logic components are detected by code observation for efficiency; components having cryptographic logic are verified through testing and are sandboxed. It proposes the concept of native sandbox NaPu. It is used for ensuring pure function. .

2. In [2] static disassembly method is use to make login more difficult for attacker. If attacker uses reverse engineering then it can easily gain the access of system vulnerabilities. The executable code is disassemble which translate machine level code to assembly level code is called as reverse engineering. From machine level code attacker can easily get the source code of system.

3. In [3] Defensive strategy Blue-chip is use which has both a design time component and a runtime component. It uses unused circuit identification (UCI) while verification of design phases to identify doubtful circuitry. When Blue Chip detects such suspicious behavior it discards it and replaces it with exception Generation hardware..

4. In [4] Author proposed a system that is able to identify virulent actions and examine multiple execution paths. It happens when only certain conditions are met. This automatically detects the suspicious activities and prevents system from backdoor. It also observes behavior of malware depending on input read from the system.

5. In [5] it is an encrypted program which creates completely independent components of the system. A compiler level based is used here to generate obfuscated binary from malware source program.

6. In [6] these systems make observation for indicators. This indicator detects the activities which software is trying to hide.

7. In [710] Microprocessors are more vulnerable to insider attacks and can affect security integrity and privacy of computer system. In this paper a method is proposed to establish the trust in the hardware system. Microprocessor is embedded with practical, lightweight attack detectors. These light weight detectors can protect against malicious logic which is inserted in microprocessor.

8. In [811] author concentrated on kernel root kit identification system for the Limbo, Limbo is a windows platform. It checks legitimacy of each kernel driver before loading it into main kernel and which prevents kernel root kits.

III. PROPOSED METHOD

The develop solution secures response computable authentication from backdoors. This solution either checks and detect for hidden backdoors or ensures the bad-natured developer cannot neglect authentication even with a backdoor are inserted in the authentication process. This introduces key component - the NaPu sandbox into a new RCA framework.

The proposed system has following main roles:

A) User login Module.
B) Explicit Response Comparison.
C) Function Purification.
D) Secure Function Testing.

A. User Login Module:

In user login module the login system is followed by the Response comparison module. The user must enter the accurate login Id and password pair which he stored while registering himself.

B. Explicit Response Comparison:

In login module if the user ID and the password entered by the user are valid then system generates a task for user. This task is given by the server and it is calculated by both user and server. If the user response and the server response are matched then the user is goes for next module.

C. Function Purification:

The login module is then passing through the sandboxing technique. Sandbox contains four components as Deterministic Memory allocator [1], memory wiper [1], and pure function interfaces [1] and instruction validator [1]. The function purification interfaces and instruction validator avoids to get global state, this global state can be used to triggering the system at specific time. Function running in NaPu having some special instruction to get global state of system NaPu throws Exception and removes global state. Deterministic memory allocator and memory wiper always load states before every execution of untrusted module to avoid backdoor which are generated by persistent internal states. If for every login request address is different then it can be easily guessed and tracked. So deterministic memory allocator is used to start each login request from fixed address value. Memory wiper is used to fill buffer with zero to avoid uncertainty in memory allocation. Each and every time proposed solution resets memory for calculation of response computation function f, and allocate memory in fixed manner

D. Secure Function Testing:

This module describes the final hand over a backdoor detection and elimination. It ensures that the system does not content any backdoors. This module uses the collision testing to finalize the elimination of the backdoors.

E. Algorithm:

Steps:
a) Input the login system consists of backdoors.
b) User authentication process.
c) Challenge generation and Response computation.
d) Response comparison and decomposition.

 ISSN: 2321-8169
4144 – 4147

e) Sandboxing method for vulnerability isolation and internal state resetting.

f) Function purification.

g) Secure function testing based on collision testing.

Explanation

The user must enter the accurate login Id and password pair Which he stored while registering himself. If the user ID and the password entered by the user is valid then system generates a task for user. This task is given by the server and it is calculated by both user and server. If the user response and the server response are matched then the user is goes for next process. Sandbox contains four components as Deterministic memory allocator, memory wiper, and pure function interfaces and instruction validator. The function purification interfaces and instruction validator avoids to get global state, this global state can be used to triggering the system at specific time. Function running in NaPu having some special instruction to get global state of system NaPu throws Exception and removes global state. Deterministic memory allocator and memory wiper always load states before every execution of untrusted module to avoid backdoor from being prompted by permanent internal states. The function purification describes the final hand over a backdoor detection and elimination. It ensures that the system does not content any backdoors. This module uses the collision testing to finalize the elimination of the backdoors.

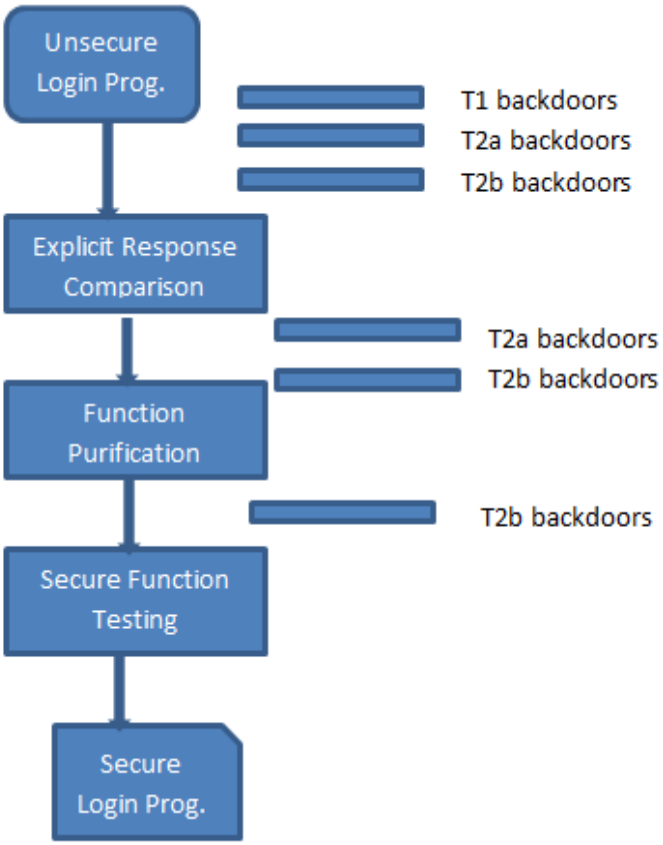

Fig. 1: Product Overview

IV. RESULT

System is able to detect and remove all backdoors present in RCA framework. This prevention of backdoors is effective and highly secure. System does not allow any malicious entry inside login system. Detection of backdoor and its prevention done in less time, Output is accurate and efficient. In graphical analysis system shows login attempt of malicious developer. Graphical simulation shows backdoor detection.

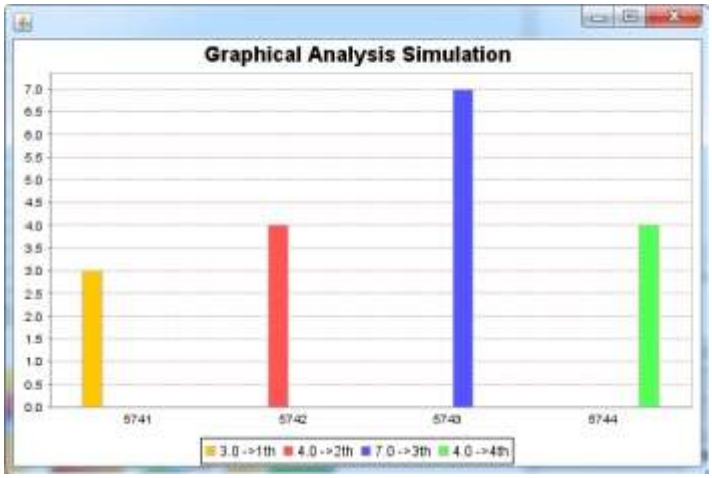

Fig. 2: Backdoor Detection

V. CONCLUSION

Thus here I come to conclude that proposed method which uses combination of several methods and it can be able to prevent backdoor entries efficiently. Here I have use RCA response computation function L () along with sandboxing pure function F () and secure function testing which is also known as collision testing.

ACKNOWLEDGMENT

I would like to express my sincere thanks to my project guide Prof.Dhaigude T.A, for giving me admirable guidance, worthwhile feedback and consistent inspiration during project work. His suggestions were great help for me during entire project work. His guidance during project work makes me do project efficiently. Last but not least I would like thanks to all those who helped in project work.

REFERENCES

[1] “A Framework to Eliminate Backdoors from Response-Computable Authentication” Shuaifu Dai1, Tao Wei1, 2, Chao Zhang1, Tielei Wang3,Yu Ding1, Zhenkai Liang4, Wei Zou1.

[2] Obfuscation of executable code to improve resistance to static disassembly”, C. Linn and S. Debray.

[3] Detecting and Removing Malicious Hardware Automatically, M. Hicks, M. Finnicum, S. T. King, M. M. K. Martin, and J. M. Smith.

[4] Exploring Multiple Execution Paths for Malware Analysis, A. Moser, C. Kruegel, and E. Kirda.

[5] ”Impeding malware analysis using conditional code obfuscation”,M. Sharif, A. Lanzi, J. Giffin, and W. Lee.

[6] ”Detecting Certified Pre-owns Software”,Tyler Shields, Veracode Chris Wysopal, Veracode.

[7] ”Tamper Evident Microprocessors”, A. Waksman and S. Sethumadhavan.
[8] ”A forced sampled execution approach to kernel rootkit identification”,J. Wilhelm and T.cker Chiueh.
[9] ”Silencing Hardware Backdoors”, A. Waksman.
[10] ”Native Client: A Sandbox for Portable, Untrusted x86 Native Code”. B. Yee, D. Sehr, G. Dardyk, J. B. Chen, R. Muth, T. Ormandy, S. Okasaka,
[11] ”Remote timing attacks are practical”,D. Brumley and D. Boneh,