International Journal of Advanced Smart Convergence Vol.5 No4 54-56 (2016)
https://doi.org/10.7236/IJASC.2016.5.4.54

IJASC 16-4-9

Code-Reuse Attack Detection Using Kullback-Leibler Divergence in IoT

Jun-Won Ho

Department of Information Security, Seoul Women's University
621 Hwarangro, Nowon-Gu, Seoul, South Korea
jwho@swu.ac.kr

Abstract

Code-reuse attacks are very dangerous in various systems. This is because they do not inject malicious codes into target systems, but reuse the instruction sequences in executable files or libraries of target systems. Moreover, code-reuse attacks could be more harmful to IoT systems in the sense that it may not be easy to devise efficient and effective mechanism for code-reuse attack detection in resource-restricted IoT devices. In this paper, we propose a detection scheme with using Kullback-Leibler (KL) divergence to combat against code-reuse attacks in IoT. Specifically, we detect code-reuse attacks by calculating KL divergence between the probability distributions of the packets that generate from IoT devices and contain code region addresses in memory system and the probability distributions of the packets that come to IoT devices and contain code region addresses in memory system, checking if the computed KL divergence is abnormal.

Key words: *Code-Reuse Attack, Kullback-Leibler Divergence, IoT*

1. Introduction

In conventional code-injection attack, attacker prepares malicious codes and injects them into target systems by exploiting the vulnerabilities of target systems. Although attacker could generate arbitrary malicious codes in accordance with his own needs, malicious codes could be readily detected once their characteristics are identified by defense system. Thus, attacker may be interested in launching code-reuse attacks [9] in which he exploits vulnerabilities with reusing instruction sequences of executables or libraries stored in code region of target memory systems. In the sense that instruction sequences in code region of target memory systems are reused for the attack, it could be likely more difficult to detect code-reuse attacks than code-injection attacks. Furthermore, attacker will likely prefer to mount code-reuse attacks rather than code-injection attacks against IoT devices which are usually resource-limited. This is reasonable in the sense that he can readily reuse the instruction sequences of devices instead of injecting substantial amount of codes into resource-restricted device.

To fight against code-reuse attacks, many researchers have recently come up with a variety of solutions [1-7]. Several prevention and detection schemes have been proposed in [1-3, 6-7]. Additionally, security analysis

Manuscript Received: Oct. 26, 2016 / Revised: Nov. 15, 2016 / Accepted: Nov. 25, 2016
Corresponding Author: jwho@swu.ac.kr
Tel: +82-2-970-5607, Fax: +82-2-970-5981
Department of Information Security, Seoul Women's University

on the existing schemes has been explored in [4-5].

In this paper, we propose a code-reuse attack detection scheme using Kullback-Leibler divergence [8] in IoT. The key idea is to check whether the packets incoming into IoT devices exhibit abnormal characteristics in terms of probability distributions of the packets that contain code region addresses in memory system.

Once being discerned as abnormal, the incoming packets are quarantined and inspected in detail.

2. Related Work

In this section, we describe the relevant work for code-reuse defense in the literature. Habibi et al. proposed a guarding technique against code reuse attacks exploiting buffer overflow vulnerabilities on raspberry pi devices. The key idea is to paralyze the attacker's capability to maneuver the return address of the function [1]. Bletsch et al. protects against code-reuse attacks through the variants of control flow integrity techniques [2]. Follner et al. detects the code-reuse attacks by dynamically tracking the execution flow [3]. Davi et al. presented thorough security analysis on various control flow integrity schemes against code-reuse attacks and found out that adversary could actually circumvent these schemes [4]. Göktas et al. showed that code-reuse protection schemes based on checking the length of reused instruction sequences have difficulty in efficiently determining the length of reused instruction sequences [5]. In [6], authors utilized both code and execution path randomization techniques to fight against ROP and JIT-ROP attacks. Davi et al. developed a tool to dynamically detect ROP attacks [7].

3. Code-Reuse Attack Detection Using KL-Divergence

In this section, we describe the details of code-reuse detection scheme based on KL divergence. The central idea of our proposed scheme is that we first use KL divergence to compare the probability distributions of packets that come into IoT devices and contain code region addresses in memory system to the probability distributions of the packets that originate from these devices and contain code region addresses in memory system. If we find out the substantial deviation from the comparison, we suspect the incoming packets as being used in code-reuse attacks and put quarantine on these packets for further examination in detail. Specifically, we apply KL divergence to detect the considerable divergence in probability distributions as follows:

We first assume that IoT device A sends a series of packets to IoT device B. Upon receiving a packet T_i from device A, device B decomposes incoming packet T_i into k blocks such that a block size is 4-byte and checks if each block's value match with one of code region addresses in memory system of device B. If so, the block in that match is regarded as the evidence of the occurrence of code-reuse attacks. Each time device B generates a packet T_o, as did against incoming packets, it performs the code-reuse match test for each block in the packet T_o. Let us assume that a block in the incoming (resp. originating) packet T_i (resp. T_o) into (resp. from) device B is decided as the clue of the occurrence of code-reuse attacks with the probability P_i (resp. P_o). Device B first initializes N_r to 0 such that N_r is used as the counter for code-reuse detection. It then computes KL-divergence z for each block in incoming packet P_i as follows:

$$z = P_o \ln \frac{P_o}{P_i} + (1 - P_o) \ln \frac{(1-P_o)}{(1-P_i)} \quad (1)$$

Whenever z exceeds a pre-defined threshold z', N_r is incremented by 1. If N_r exceeds a pre-set threshold $\aleph$,

the incoming packet P_i is suspected as the part of code-reuse attacks and it is accordingly quarantined for further investigation in detail.

4. Conclusion

In this paper, we propose a code-reuse attack detection scheme based on KL divergence in IoT. Our proposed scheme adapts KL divergence to compare the probability distributions of the packets incoming to and emanating from IoT devices such that the packets contain code region addresses in order to detect the code-reuse attacks incurred by packets incoming into IoT devices. As future work, we are interested in evaluating the security and performance of the proposed scheme. Moreover, we hope to explore the proposed scheme from the perspective of experimental study.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2016R1C1B1014126).

References

[1] J. Habibi, A. Panicker, A. Gupta, and E. Bertino, "DisARM: Mitigating Buffer Overflow Attacks on Embedded Devices", CERIAS Tech Report 2015-15, 2015.
[2] T. Bletsch, X. Jiang, V. Fresh, "Mitigating Code-Reuse Attacks with Control-Flow Locking", ACSAC, 2011.
[3] A. Follner, E. Bodden, "ROPocop - Dynamic mitigation code-reuse attacks", Journal of Information Security and Applications, 29, pp. 16-26, 2016.
[4] L. Davi, A-R. Sadeghi, D. Lehmann, F. Monrose, "Stitching the Gadgets: On the Ineffectiveness of Coarse-Grained Control-Flow Integrity Protection", Usenix Security, 2014.
[5] E. Göktas ¸ E. Athanasopoulos, M. Polychronakis, H. Bos, G. Portokalidis, "Size Does Matter: Why Using Gadget-Chain Length to Prevent Code-Reuse Attacks is Hard", Usenix Security, 2014.
[6] L. Davi, C. Liebchen, A-R. Sadeghi, K. Z. Snow, F. Monrose, "Isomeron: Code Randomization Resilient to (Just-In-Time) Return-Oriented Programming", NDSS, 2015.
[7] L. Davi, A-R. Sadeghi, M. Winandy, "ROPdefender: A Detection Tool to Defend Against Return-Oriented Programming Attacks", ASIACCS, 2011.
[8] T.M. Cover, J.A. Thomas, "Elements of Information Theory, Wiley, 2006.
[9] R. Roemer, E. Buchanan, H. Shacham, and S. Savage, "Return-oriented programming: Systems, languages, and applications", ACM Transactions on Information and System Security, 15, 1 (Mar. 2012), 2:1–2:34.