The International Journal Of Engineering And Science (IJES)
|| Volume || 5 || Issue || 6 || Pages || PP -98-102 || 2016 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Novel Advances in Measuring and Preventing Software Security Weakness: Continuous Trust Restoration (CTR)

Abdulrehman A. Mohamedand Dr. Michael W. Kimwele, Phd
School Of Computing And Information Technology (SCIT), College Of Pure And Applied Sciences (COPAS),Jomo Kenyatta University Of Agriculture And Technology (Mombasa Campus), P. O. Box94090–80107, Mombasa
School Of Computing And Information Technology (SCIT), College Of Pure And Applied Sciences (COPAS),Jomo Kenyatta University Of Agriculture And Technology (JKUAT), P. O. Box 62000- 00200, Nairobi. Kenya

----------ABSTRACT----------
Software weaknesses in design, architecture, code and deployment have led to software vulnerability exploited by the perpetrators. Although counter measure tools have been developed such as patch management systems, firewalls and antivirus, but the perpetrators have advance sophisticated tools such malware with crypto-lock and crypto-wall technologies. The current counter measures technologies are based on detection and respond model or risk management framework, which are no match to the attacker's technologies based on speed technologies such as machine generated malwares and precision or stealth technologies such as command-and-control node malwares. Although lots of ink has been poured on advances in measuring and preventing software weakness on the detection and respond concept,this study is motivated to explore the state-of-art advances specifically on the novel concept of Continuous Trust Restoration (CTR). The Continuous Trust Restoration is a process of breaking down attacker's activities kill chain and restoring the system trust. The CTR concept deploys speed, precision and stealth technologies on random route mutation, random host mutation, hypervisors, trust boot, software identities and software define infrastructure. Moreover, to deploy these technologies the study further explores a common security architectural framework with software metrics such as CVE (Common Vulnerability and Exposure), CWE (Common Weakness Enumeration), CVSS (Common Vulnerability Scoring System), CWSS (Common Weakness Scoring System), and CAPEC (Common Attack Pattern Enumeration and Classification). Finally, the study recommends a software security counter measures research paradigm shift from the current detection and respond models to Continuous Trust Restoration concept and from risk management frameworks to a Common Security Architectural Framework.

Keywords: *Continuous Trust Restoration (CTR)), CVE (Common Vulnerability and Exposure), CWE (Common Weakness Enumeration), CVSS (Common Vulnerability Scoring System), CWSS (Common Weakness Scoring System), and CAPEC (Common Attack Pattern Enumeration and Classification).*

Date of Submission: 17 May 2016 Date of Accepted: 30 June 2016

I. INTRODUCTION

1.1 Background Information

The computer platform paradigm shift from mainframe, personal computers, web, mobile, smartphones, tablets, and to cloud have ledto digital revolution of Internet of Things. The commercial industries have taken advantage of these technologies to manufacture cheap devices ranging from wearable which gather personal data to the cloud and to vehicle fitted with Wi-Fi technology. Even though some of these devices were built with automated patch enable and update enable systems, but the majority of them were built with no architectural security measures. These software weaknesses have led to enormous vulnerabilities to the cyber space to be exploited by hackers. As a result of this, there is urgency in the academic world in research advances for measuring and preventing software weakness. Therefore, the study is motivated to explore two main research concepts of; Continuous Trust Restoration (CTR) and common security architectural framework.

According to (Boyle, 2015), Continuous Trust Restorationis a process whereby networks are continually renewing themselves which can give cyber defenders the advantage of speed, stealth, and precision in getting ahead of the growing cyber threat from attackers.If an attacker scans the network to build a map, the CTR technologies such as random host mutation should be able to restore the system into a state of trust within 24 hours.

It is reported by (R. Martin, 2015) that, in today's world of Internet of Things (IoT), everything is connected and co-dependent, hence, when a system get subverted through un-patched vulnerability, or a miss-configuration or software weakness then everything is susceptible to attack. Currently, vendors use their own security

frameworks for developing devices, which make it difficult to measure and prevent software weakness. Therefore, research advances, in software weakness and measure are directed towards developing common security architectural framework.

II. CONTINUOUS TRUST RESTORATION (CTR)

2.1 Introduction

The control of cyber domain has been a nightmare according to (Boyle, 2015), but controlling the physical domain such as the air domain which brings the image of F35; the most advance fighter aircraft in the world, the sea domain which brings the images of nuclear submarines; running deep silently with stealth coordinating the surface ships, the land domain with solders equipped with laser designators and precision guided bombs, and the space domain which bring myriad of complex satellite ground station systems. All these domains can be controlled and have a common designattributes of speed, precision and stealth; as they can move fast, un-noticed and hit their target with precision.

While the cyber domain toolkit contains machine generated malwares introducing the strategy of speed and command-and-control node malwares with precision and stealth strategies, against the unmatched strategies of detect and respond currently(Bosire & Kimwele, 2015). Therefore, the advances in research for software weakness extended this idea to a concept called Continuous Trust Restoration (CTR) where the perpetrator's activities are quelled before they are mature using strategies of speed, precision and stealth.

2.2 Hypervisor

It is elaborated by (Prasad, 2014) that, hypervisor (Virtual Machine Monitor) is a piece of software, firmware or hardware that gives an impression to the guest machines (virtual machines) as if they were operating on a physical hardware. Its main purpose is to allow multiple "machines" to share a single hardware platform. It separates the operating system (OS) from the hardware by taking the responsibility of allowing each running OS time with the underlying hardware. It acts as a traffic controller to allow time to use the CPU, memory, GPU, and other hardware.

2.2.1 Types of Hypervisor

It is explained by (Sumastre, 2016) that, there are two types of hypervisors: bare metal (native or type I) hypervisors; which are run on the host's hardware to control it as well as manage the virtual machines on it. They included various examples such as Microsoft Hyper-V hypervisor, VMware ESX/ESXi, Oracle VM Server for x86, KVM, or Citrix XenServer, and the embedded (hosted or type II) hypervisors; which are run as software using an operating system such as Windows, Linux or FreeBSD. They included various examples such as Virtagehypervisor, VirtualBox, and VMWare.Therefore, native hypervisors run directly on the hardware while a hosted hypervisor needs an operating system to do its work.

Although, type I hypervisors are more secure than type II hypervisors but are target for hackers, because they are designed to control all the resources of the hardware while managing all the virtual machines residing on it. Therefore, the Continuous Trust Restoration concept explores research advancements in developing hypervisors with strategies of speed, precision and stealth.

2.3 Random Host Mutation (RHM)

It is reported by (Al-Shaer, Duan, & Jafarian, 2012) that RHM is an approach that turns end-hosts into untraceable moving targets by transparently mutating their IP addresses in an intelligent and unpredictable fashion and without sacrificing network integrity, manageability or performance. In RHM, moving target hosts are assigned virtual IP addresses that change randomly and synchronously in a distributed fashion over time. In order to prevent disruption of active connections, the IP address mutation is managed by network appliances and totally transparent to end-host. AlthoughRHM can effectively defend against stealthy scanning, many types of worm propagation and attacks that require reconnaissance for successful launching, but the IP addresses and ports still are prerequisite to many host and network attacks. Therefore, the Continuous Trust Restoration concept explores research advancements in developing RHMs with strategies of speed, precision and stealth.

2.4 Software identities

It is reported by (Rouse, 2013) that, software identities deals with identifying individuals in a system and controlling their access to resources within that system by associating user rights and restrictions with the established identity. At the core of an identity management system are policies defining which devices and users are allowed on the network and what a user can accomplish, including policy definition, reporting, alerts, alarms and other common management and operations requirements. An alarm might be triggered, for example, when a specific user tries to access a resource for which they do not have permission. Reporting produces an audit log documenting what specific activities were initiated.

The information in digital identities is used by computers to make decisions about how to interact with external agents. It allows a computer to answer two basic questions:which external agent is it interacting, and has it interacted with an external agent in the past. The information contained in a digital identity allows these questions to be answered without the involvement of human operators. The hackers take advantage of the vulnerabilities in the software of this concept to launch their attacks. Therefore, the Continuous Trust Restoration concept explores research advancements in developing software identities with strategies of speed, precision and stealth.

2.5 Random Route Mutation

It is asserted by (Duan, Al-Shaer, & Jafarian, 2013) that, Random Route Mutation (RRM) is a technique that enables changing randomly the route of the multiple flows in a network simultaneously to defend against reconnaissance, eavesdrop and DoS attacks, while preserving end-to-end quality of service (QoS) properties
Although RRM can protect at least 90% of the packet flow from being attacked against adversaries attackers, as compared with static routes but still recently the number of attacks specifically from adversaries to eavesdrop, or launch denial of service (DoS) attacks on certain network flows have increased drastically. As result of this gap, the Continuous Trust Restoration concept explores research advancements in developing random route mutation with strategies of speed, precision and stealth.

2.6 Software Define Infrastructure (SDI)

It is described by (Piff, 2015) that, Software Defined Infrastructure (SDI) is the definition of technical computing infrastructure entirely under the control of software with no operator or human intervention. It operates independent of any hardware-specific dependencies and is programmatically extensible. The concept refers to the ability to define application requirements from the infrastructure (both functional and non-functional requirements) and have physical implementation of the hardware require delivering those requirements automatically derived and provisioned (Goldman, 2015).
AlthoughSDI, have the intelligence to manage the hardware, supported workloads, automatically corrects issues, and optimizes performance to ensure security but still the adversaries have found vulnerabilities around these intelligent software to make calculated attacks. This creates a gap which can be address by the Continuous Trust Restoration concept with strategies of speed, precision and stealth.

2.7 Trusted boot

According to (Loisel & di Vito, 2015)that, the trusted Boot, is a security feature that leverages the Unified Extensible Firmware Interface (UEFI) to block the loading and operation of any program or driver that has not been signed by an OS-provided key, and thus protects the integrity of the kernel, system files, boot-critical drivers, and even antimalware software.When a rootkit is encountered, the UEFI wouldn't allow it to boot. In other words, UEFI protects the pre-OS environment. Additionally, as the system boots, the OS detects if any of the OS elements have been tampered with and automatically restores the unmodified versions.
Although, impregnable protection are achieved using read-only memory (ROM), or flash (EEPROM) memory internal to the microcontroller to store the root-of-trust software, the adversaries have gain access to various systems through this route and inflict harm. As a result which the counter measures should be to employ the concept of Continuous Trust Restoration concept with strategies of speed, precision and stealth.

III. COMMON SECURITY ARCHITECTURAL FRAMEWORK

3.1 Introduction

According to (R. Martin, 2015), the idea of standards in software security is of an urgency now than ever before, because cyber security is not an individual issue but communal issues and its mitigation is communal. Therefore, the idea of Common Security Architectural Framework is core to advances in measuring and preventing software weakness.These advances are witness in collaboration of various stakeholders, both private and government. The Consortium for IT Software Quality (CISQ) developed an Automated Source Code Security Measure that predicts the vulnerability of source code to external attack. The measures were based on the Top 25 in the Common Weakness Enumeration (CWE) repository which is managed by the MITRE Corporation. The system was also endorsed by Object Management Group (OMG), an international, open membership and not-for-profit technology standards consortium.
It is further elaborated by (R. Martin, 2015) that, the center of the Common Security Architectural Framework is MITRE Corporation, a non-profit organization that manages Federally Funded Research and Development Centers (FFRDCs) and supporting various governmental organs such the Department of Defense (DOD and the National Institute of Standards and Technology (NIST) among others. Therefore, the study is motivated to explore the metrics supported by MITRE Corporation as areas of advance research in software measures and weakness. The metrics are; CVE (Common Vulnerability and Exposure), CWE (Common Weakness

Enumeration), CVSS (Common Vulnerability Scoring System), CWSS (Common Weakness Scoring System), and CAPEC (Common Attack Pattern Enumeration and Classification).

3.2 CVE (Common Vulnerability and Exposure)

According to (MITRE, 2016), Common Vulnerabilities and Exposures (CVE) is a dictionary of common names (i.e., CVE Identifiers) for publicly known cyber-security vulnerabilities. CVE's common identifiers make it easier to share data across separate network security databases and tools, and provide a baseline for evaluating the coverage of an organization's security tools. CVE was launched in 1999 when most information security tools used their own databases with their own names for security vulnerabilities. The consequences were potential gaps in security coverage and no effective interoperability among the disparate databases and tools. In addition, each tool vendor used different metrics to state the number of vulnerabilities or exposures they detected, which meant there was no standardized basis for evaluation among the tools

The process of creating a CVE Identifier begins with the discovery of potential security vulnerability. The information is then assigned a CVE Identifier by a CVE Numbering Authority (CNA) and posted on the CVE List on the CVE website by the CVE Editor. As part of its management of CVE, The MITRE Corporation functions as Editor and Primary CNA.In addition to approving the data sources and product coverage goals for entries on the CVE List, the CVE Editorial Board oversees this process.

3.3 CWE (Common Weakness Enumeration)

(Rouse, 2013) describes that, Common Weakness Enumeration (CWE) is a universal online dictionary of weaknesses that have been found in computer software. The dictionary is maintained by the MITRE Corporation and can be accessed free on a worldwide basis. The purpose of CWE is to facilitate the effective use of tools that can identify, find and resolve bugs, vulnerabilities and exposures in computer software before the programs are publicly distributed or sold.

As a result of this, it also described by (Needham, 2015) that, the CISQ developed an Automated Source Code Security Measure and it'sAutomated Quality Characteristic Measures are conformant to the definitions of the following quality characteristics in ISO/IEC 25010; Security : Identifies critical security violations in the source code drawn from the Top 25 security weaknesses in the Common Weakness Enumeration (CWE) repository, Reliability: Identifies critical violations of availability, fault tolerance, and recoverability of software, Performance Efficiency: Identifies critical violations of response time, as well as processor, memory, and utilization of other resources by the software, and Maintainability: Identifies critical violations of modularity, architectural compliance, reusability, analyzability, and changeability in software

3.4 CVSS (Common Vulnerability Scoring System)

It is asserted by (Czagan, 2013) that, the Common Vulnerability Scoring System (CVSS) is a free and open industry standard for assessing the severity of computer system security vulnerabilities. CVSS attempts to assign severity scores to vulnerabilities, allowing responders to prioritize responses and resources according to threat. Scores are calculated based on a formula that depends on several metrics that approximate ease of exploit and the impact of exploit. Scores range from 0 to 10, with 10 being the most severe. While many utilize only the CVSS Base score for determining severity, Temporal and Environmental scores also exist, to factor in availability of mitigations and how widespread vulnerable systems are within an organization, respectively.

The CVSS assessment measures three areas of concern: base Metrics for qualities intrinsic to vulnerability, temporal Metrics for characteristics that evolve over the lifetime of vulnerability, and environmental Metrics for vulnerabilities that depend on a particular implementation or environment. A numerical score is generated for each of these metric groups. A vector string (or simply "vector" in CVSS), represents the values of all the metrics as a block of text.

3.5 CWSS (Common Weakness Scoring System)

According to (B. Martin, 2014) the Common Weakness Scoring System (CWSS) provides a mechanism for prioritizing software weaknesses in a consistent, flexible, open manner. It is a collaborative, community-based effort that is addressing the needs of its stakeholders across government, academia, and industry. CWSS is organized into three metric groups: base finding, attack surface, and environmental. Each group contains multiple metrics - also known as factors - that are used to compute a CWSS score for a weakness.

Various weakness scoring systems have been used or proposed over the years. Automated tools such as source code scanners typically perform their own custom scoring; as a result, multiple tools can produce inconsistent scores for the same weakness. The Common Vulnerability Scoring System (CVSS) is perhaps the most similar scoring system. However, it has some important limitations that make it difficult to adapt to software security assessment.

3.6 CAPEC (Common Attack Pattern Enumeration and Classification)

According (Picuira, 2016), CAPEC (Common Attack Patterns Enumeration and Classification) is a community-developed formal list of common attack patterns. Attack patterns are descriptions of common methods for exploiting software providing the attacker's perspective and guidance on ways to mitigate their effect. They derive from the concept of design patterns applied in a destructive rather than constructive context and are generated from in-depth analysis of specific real-world exploit examples. Security-Database use CVEs along the appropriate CAPECs if available.

"CAPEC-compatible" means that a tool, Web site, database, or other security product or service uses CAPEC names in a manner that allows it to be cross-referenced with other products that employ CAPEC names. Security-Database is creating a new generation of complete XML feed. The complete XML feedwill enumerate all known information on vulnerability (CVE, CPE, OVAL ID, CVSS, CWE, CAPEC, CCE, Vendor Patches etc.)

IV. RECOMMENDATION

The reviews revealed a shift away from the traditional reactive detection and respond principles to more proactive continuous trust restoration principles. Therefore, the study recommends a further exploration of software security counter measures research paradigm shift from the current detection and respond models to Continuous Trust Restoration concept and from risk management frameworks to a Common Security Architectural Framework.

ACKNOWLEDGMENT

I would like to acknowledge and appreciate the material and moral support that I have always received from my family, colleagues and friends. Special thanks to my lecturer and mentor Dr. Michael W. Kimwele, Deputy Director, School of Computing and Information Technology (SCIT), College of Pure and Applied Sciences (COPAS)Jomo Kenyatta University of Agriculture and Technology (JKUAT), P. O. Box 62000- 00200, Nairobi- Kenya for his kind words of encouragement and guidance.

REFERENCE

[1]. Al-Shaer, E., Duan, Q., & Jafarian, J. H. (2012). Random Host Mutation for Moving Target Defense. In A. D. Keromytis & R. D. Pietro (Eds.), Security and Privacy in Communication Networks (pp. 310–327). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-36883-7_19

[2]. Bosire, A., & Kimwele, M. (2015). Advances in Measuring and Preventing Software Security Weaknesses. International Journal of Advanced Research in Computer Science and Software Engineering Research Paper, 5(12), 150–154.

[3]. Boyle, V. (2015). The Cutting Edge of Cybersecurity Research. In How Cyber Defenders Can Get Ahead: Speed, Stealth, and Precision Grumman. Washington, D.C. Retrieved from http://www.northropgrumman.com/Capabilities/Cybersecurity/Pages/CSM_Passcode_Cybersecurity_Event.aspx

[4]. Czagan, D. (2013). Common Vulnerability Scoring System - InfoSec Resources. Retrieved June 17, 2016, from http://resources.infosecinstitute.com/common-vulnerability-scoring-system/

[5]. Duan, Q., Al-Shaer, E., & Jafarian, H. (2013). Efficient Random Route Mutation considering flow and network constraints. In 2013 IEEE Conference on Communications and Network Security (CNS) (pp. 260–268). http://doi.org/10.1109/CNS.2013.6682715

[6]. Goldman, E. (2015). Where Are You on the Road to Software-Defined Infrastructure? | CIO. Retrieved June 17, 2016, from http://www.cio.com/article/2931083/infrastructure/where-are-you-on-the-road-to-software-defined-infrastructure.html

[7]. Loisel, Y., & di Vito, S. (2015). Securing the IoT: Part 2 - Secure boot as root of trust. Retrieved June 16, 2016, from http://www.embedded.com/design/safety-and-security/4438300/Securing-the-IoT--Part-2---Secure-boot-as-root-of-trust-

[8]. Martin, B. (2014). CWE - Common Weakness Scoring System (CWSS). Retrieved June 17, 2016, from https://cwe.mitre.org/cwss/cwss_v1.0.1.html

[9]. Martin, R. (2015). Latest Advances in Cybersecurity and the NEW CISQ Security Standard | CISQ - Consortium for IT Software Quality. In Latest Advances in Cybersecurity and the NEW OMG/CISQ Security Standard. Washington, D.C. Retrieved from http://it-cisq.org/cisq-webcast-latest-advances-in-cybersecurity-and-the-new-cisq-security-standard/

[10]. MITRE. (2016). CVE - About CVE. Retrieved June 17, 2016, from https://cve.mitre.org/about/

[11]. Needham, M. A. (2015). CISQ Announces New Measures for Software Quality. Retrieved June 17, 2016, from http://www.darkreading.com/application-security/cisq-announces-new-measures-for-software-quality-/d/d-id/1322198

[12]. Picuira, B. (2016). CAPEC Compatibility - Security Database. Retrieved June 17, 2016, from https://www.security-database.com/about.php?type=capec

[13]. Piff, S. (2015). CIO-Asia - Drive for innovation boosts demand for software-defined technologies: IDC. Retrieved June 17, 2016, from http://www.cio-asia.com/mgmt/leadership-and-mgmt/drive-for-innovation-boosts-demand-for-software-defined-technologies-idc/

[14]. Prasad, D. (2014). Comparison Type 1 vs Type 2 Hypervisor ~ GoLinuxHub. Retrieved from http://www.golinuxhub.com/2014/07/comparison-type-1-vs-type-2-hypervisor.html

[15]. Rouse, M. (2013). What is identity management (ID management) ? - Definition from WhatIs.com. Retrieved June 16, 2016, from http://searchsecurity.techtarget.com/definition/identity-management-ID-management

[16]. Sumastre, M. G. (2016). Virtualization 101: What is a Hypervisor? Retrieved from https://www.pluralsight.com/blog/it-ops/what-is-hypervisor