International Journal of Computer Networks and Communications Security

VOL. 4, NO. 11, NOVEMBER 2016, 303–308
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

Collect, Study and Preparation of Standards for Security and Stability in Desktop Applications

Mohammad Sedighimanesh[1], Ali Sedighimanesh[2], Javad Baqeri[3]

[1, 2, 3] Department of Electrical, Computer and IT Engineering, Islamic Azad University of Qazvin, Qom, Iran

[1]*mohammad.sedighimanesh@gmail.com*, [2]*ali.sedighimanesh@gmail.com*, [3]*javad.baqer2010@gmail.com*

ABSTRACT

The use of technology has been always a double-edged sword. As each new technology can help to improve the work and life of humans, on the other hand can lead to the emergence of vulnerabilities and risks in work and life. This article contains all the technology available in the world. Perhaps a major reason for it is that every technology from start formed just for the convenience and probably accelerate of work and its creators do not think about the risks that will be occurring in the future. The issue that we have discussed in this study is data protection, database encryption, error handling, software security and desktop apps under Linux that although has many advantages and made easier doing a lot of things but at the same time results in more security vulnerabilities in works and privacy of humans.

Keywords: *Vulnerability, Data Protection, Encryption, Database Encryption, Error Handling, Software Security.*

1 INTRODUCTION

Linux [1, 2] is the core of UNIX operating systems that that was developed in 1991 by Linus Trovaldz. The kernel is the heart of the Linux operating system that manages the system's resources. These resources include {file management, multi programs, memory management, management of I / O, process management, management of peripheral devices, network support, special features (including virtual memory, shared libraries, demand loading and so on)}. Kernel decides who will be given the resources and it specifies its time and duration. After the formation of initial core of Linux, the number of programmers around the world has formed their assembly and start production of different versions according to their initial core. This version of the so-called distribution. The following are the names of some of the distributions [3]: Arch Linux, Slackware, SentOS, Debian, Fedora, Gentoo, Knoppix, Kubuntu, Linux Mint, Mandriva, openSUSE.

2 PRINCIPLES OF SECURE PROGRAMMING

In most cases, simply avoidable programming errors lead to exploitable vulnerabilities in software. Tips for secure programming that should be given to them, explain[4, 5]: Validate input, Heed compiler warning, Architect and design for security policies, keep it simple, Default deny, Adhere to the principle of least privilege, Sanitize data sent to other systems, Practice defense in depth ,Use effective quality assurance techniques, Adopt a secure coding standard, Define security requirements, Model threats

The main element in secure coding programming with multiple language, is good documentation and the applicable standards. Coding standards, encourage programmers to follow a uniform set of rules and guidelines that determined based on the needs of the project and the organization, not based on the preferences and programming skills. As soon as a determination of the above mentioned standards,it can be used as a measure of source code, either manually or automatically.

3 SOFTWARE SECURITY

According to the standards [6]OWASP, four different levels of security for each system or application can be imagined. Cursory zero level is a selective level of security Where software has taken into account some security requirements. Level one or opportunistic level of every software can have level certification of the security As long as it can be strong enough against common security vulnerabilities. The second level or standard level of software is indeed evidence that the software is able to confront against the most common security vulnerabilities. The third level or advanced level , only software can have the third level of security certification that could be confront against all types of security vulnerabilities and have a very good security design.

Areas of software security include the following items[7]: client security, authentication security, user session security, checking access rights security, checking security against the command injection, security of access to the system file, storage and information retrieval security, registration of operations footprint, events and error reporting, security in system architecture, checking the security against attack to other users (attacks on site transit, etc.), testing system deployment, source code review.

4 VULNERABILITY

Today, some users in the field of IT are convinced that Linux is more secure than Windows. A rising number of systems used to install various Linux distributions. Some of the Linux distributions are more popular and others have practiced less. There are teams in the world specifically to develop software for compliance and working on Linux platform.

Software is ordinary and generally component in equipment and systems that make up our part of real life. These systems are usually complex and various programming have produced them. Programmers often make mistakes in the code, which can create software vulnerabilities. A software vulnerability is a gap or failure in the software structure that an attacker exploit it to obtain some rights and Authorities in the system. This means that vulnerability provides an entry and penetration point to into the system. Despite knowledge about the vulnerabilities, today the growing number of vulnerabilities is reported. More software security vulnerabilities are in the following categories from prospective of sources [8, 9]: Buffer overflow, not validated input, competition conditions, access control problems, lack of authentication, authorization, or encryption, XSS or Cross Site Scripting, SQL Injection, format string bug.

Twenty-five software error based on the source CWE [10] divided errors into three parts:

- insecure interaction between components: inappropriate neutralizing of certain elements used in the SQL injection, inappropriate neutralizing of certain elements used in operating system commands (OS command injection), inappropriate neutralizing of input during the production of the web page (cross-site scripting injection), unlimited upload of dangerous type files, cross-site request forgery (CSRF), redirect URL to unsecured sites.
- High risk source management: Copy buffer without checking the input size (classic buffer overflow), improper limiting of a path to the restricted directory (directory traversal),code loading without integration checking, including functionality from untrusted control area, the use of potentially dangerous function , incorrect calculation of the buffer size, uncontrolled string format, integer overflow or Wraparound,
- permeable defense: the lack of authentication for critical function, lack of permission, use certificates with fixed input, lack of encryption of sensitive data, reliance on untrusted input in a security decision, run with unnecessary points, incorrect permission, incorrect permission allocation to the critical resource, using a broken or dangerous encryption algorithm, inappropriate restrictions in an attempt of authentication , using one-way hash without Salt.

5 DATA PROTECTION AND DATABASE ENCRYPTION

Many experts believed that, most important, or rather the most precious thing on a computer, is user-created data and maybe the existence of this information that justify the necessity use of computer or network. Data-centric protection requires to survey data discovery and classification, random workflows, create / manage the policies and detection of data transmission. The technology

required to detect these cases are widespread and include [11, 12]:

- Full codding for endpoints used data, data being transfered and storage in a program (email, file servers), are among sensitive data transferred to a portable storage device.
- Data leakage prevention (DLP)is Host-based in local detect and prevent leaks of information to the data in use, in transit or disabled.
- Prevent data leakage by detection and data analysis, network monitoring (by the developed protocol and analysis program support), prevention ability for internal and external content.

Security experts described the defect location of potential security factors in three sections[12, 13]: Unauthorized information release: unauthorized person can read the computer information and make advantage from stored information, modification of Unauthorized information: unauthorized person can make changes to the information stored , denying unauthorized use: the offender could hinder the access or edit information by authorized users, even if the offender is unable to edit or access to data. Causing a malfunction.

Here there are 8 design principles that apply to security mechanisms[14]: Economy of mechanism, Fail-safe defaults, Complete mediation, Open design, Separation of privilege, Least privilege, Least common mechanism, Psychological acceptability.

Most of todays organizations heavily dependent to their database systems. Today the importance of information is to the extent that sometimes is considered as information of the all assets of an organization. Today data base must fulfill different aims. They must be reliable, provide quick access to information, and provide advanced characteristics for analysis and data storage. In addition, it should be flexible enough to allow them to adapt with many different scenarios and types of using. When data are significant, the security issue of them arises. Security services are desirable services that if available for us, we can use it to immune from attacks on our information. Main security consists of four services [15]: authentication: prove the considered person is who that claimed. Confidentiality: Information is available for allowable personnel. Integrity: data are not changed after production. Non-repudiation: data producer can not deny data production by itself. Data encryption at the database level,there are large areas of encryption that have been applied and managed. Encryption is not a magic solution and can not solve all problems, but it can decrease a lot of security risks that organizations face with them. Understanding what is working of Encryption and what it couldn't provide for organizations is critical so we can provide a proper risk management on our sensitive and regulated data [16]. Encryption algorithms can be generally divided into two general categories[17] :

- Encryption systems with symmetric-key, which uses the same key to encrypt and decrypt.
- Asymmetric encryption system, which uses two different keys instead of a key,one for encryption and another for decryption of links.

Many organizations, use from the databases as the behind scenes for sale applications or their own desirable developed applications.

Database encrypted is said to the using encryption techniques to transfer a simple database to a (a part of) encrypted database, so, it will non-readable for everyone except those who know the encryption keys[18]. Protection of sensitive / confidential data was stored in a repository, is called database security. This work is dealing with keep secure the database from any unauthorized access or threatat any level.

Security database, demanding that reject or accept the user activities on the database and its goals.

Secure database properties: Confidentiality: create limitations for secure data recovery, and therefore reject unauthorized access to data. Integration: is data are not infected in any way,and availability:is the availability of data at any given moment[18].

There are four types of control that has been proposed by Denning to provide protection of the database for us, these include[18, 19]: Access Control: This assures us that all direct access to the system are authenticated. Information flow control: control what can access to the system goals. encryption Flow control:control data with encrypts them (secured). And control is inductive (final).

Database security risks, the list is adopted from Imperva Application Defense Center [18, 20]: Excessive Privilege Abuse, Legitimate Privilege Abuse, Privilege Elevation, Database Platform Vulnerabilities, SQL Injection, Weak Audit Trail, Denial of Service, Database Communication Protocol Vulnerabilities, Weak Authentication, Backup Data Exposure

To remove security threats, every organization must identify a security law. And the security rules should include given appropriate security features that some of them are listed below from the perspective of the source [18, 20]: Access Control, Inference Policy, User Identification/Authentication, Accountability and auditing, Encryption.

Access control: Make insures that all communication with the database and other systems purposes, are based on rules and given controls. Make sure that any intervention by any attacker both internal and external, has occurred and so it protect the database against potential errors that can have a big impact, such as stopping operations of companies.

Inductive rules: inductive rules are required for data protection at a certain level. It also determines how the clear-up information can be prevented.

User Authentication/identification: identification and user authentication is necessary because it confirms the safety and also determined the identification methods, sets of people who have access to the data, and provides complex mechanisms for their access and preventing modify sensitive data by any ordinary user.

Accountability and inspection: Checking accountability and inspection to ensure the physical integrity of the data is required to manage access that is necessarily must be defined for the database through inspection and registration.

Encryption: Encryption is a process of hiding or transform the data by means of tools such as a password or code that make the Information non-readable for other people except those who have the key of information.

6 LOGGING AND ERROR HANDLING

All programs are failures or corrupted, and these errors occur at compile time or runtime. In most programming languages, runtime exceptions are related to the unauthorized and incorrect implementation of code (eg, syntax errors) and often to the form of system messages.

These defects and system messages derived from them, if not managed properly, can lead to various security risks. Among these risks, counting, buffer attacks, disclose of sensitive information, and so on. If an attack occurs, it is important that people could track attacker's activities with the aid of the law. One of the main tasks of software is creating logs. The software logs according to the importance and sensitivity of its security could be includes a lot or a little item. Many standards have suggested minimums of logs that we have described here just PCI DSS standard [21, 22]:

Following item Shall be log according to the standard: access to information, activities of people who are in the system with administrator-level access, access to logs, stop or start of recording mechanisms of log, unsuccessful access attempts to resources and information, authentication activities, the change in the authorized information stored.

The error handling can be done in two ways: 1. structured exception handling functions 2. functional error checking exception structure handling is always preferable and is used as the easiest way to cover 100% of the codes. Reports can be created for the real-time of intrusion, detection, performance and system monitoring tools. All components of intrusion into the system must be done with a synchronized server, so that record all entering to the system effectively and without error. Programs should always have a secured default. If a program fails to recognize different states, it is likely that a hacker could use this situation to gain access to unauthorized possibilities, or even worse, modify or destroy information.

7 CONCLUSION

In this work, the introduction of Linux mentioned then we tried to express the principles of secure programming and then software security was studied and then expressing vulnerabilities of software in the next section and finally database security is explained. As mentioned in the article, all software has vulnerabilities, which for various reasons, however ,penetration of hackers and disclosure and system failures could be prevented with appropriate security mechanisms.

8 REFERENCES

[1] T. R. H. Nemeth, G. Snyder, and B. Whaley-Prentice, "Basic Linux System Administration," Computer Science, 2014.

[2] R. Love, Linux Kernel Development (Novell Press): Novell Press, 2005.

[3] D. P. Bovet and M. Cesati, Understanding the Linux kernel: " O'Reilly Media, Inc.", 2005.

[4] M. Graff and K. R. Van Wyk, Secure coding: principles and practices: " O'Reilly Media, Inc.", 2003.

[5] B. Taylor, M. Bishop, D. Burley, S. Cooper, R. Dodge, and R. Seacord, "Teaching secure coding: report from summit on education in secure software," in Proceedings of the 43rd

ACM technical symposium on Computer Science Education, 2012, pp. 581-582.

[6] M. Boberski, J. Williams, and D. Wichers, "Owasp application security verification standard 2009," ed.

[7] D. P. Gilliam, T. L. Wolfe, J. S. Sherif, and M. Bishop, "Software security checklist for the software life cycle," in Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on, 2003, pp. 243-248.

[8] W. Jimenez, A. Mammar, and A. Cavalli, "Software Vulnerabilities, Prevention and Detection Methods: A Review1," in Proc. European Workshop on Security in Model Driven Architecture, 2009, pp. 6-13.

[9] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," IEEE Communications Surveys & Tutorials, vol. 18, pp. 623-654, 2015.

[10] S. Christey, R. Martin, M. Brown, A. Paller, and D. Kirby, "2011 CWE/SANS Top 25 Most Dangerous Software Errors," Common Weakness Enumeration (CWE). h ttp://cwe. mitre. org/top25, 2011.

[11] I. Basharat, F. Azam, and A. W. Muzaffar, "Database security and encryption: A survey study," International Journal of Computer Applications, vol. 47, 2012.

[12] A. Arasu, K. Eguro, R. Kaushik, and R. Ramamurthy, "Querying encrypted data," in Proceedings of the 2014 ACM SIGMOD international conference on Management of data, 2014, pp. 1259-1261.

[13] J. Ward and J. Peppard, The Strategic Management of Information Systems: Building a Digital Strategy: John Wiley & Sons, 2016.

[14] M. Bishop, "Introduction to computer security," 2005.

[15] M. Coles and R. Landrum, "Transparent Data Encryption," in Expert SQL Server 2008 Encryption, ed: Springer, 2009, pp. 127-150.

[16] D. Deshmukh, A. Pasha, and D. Qureshi, "Transparent Data Encryption--Solution for Security of Database Contents," arXiv preprint arXiv:1303.0418, 2013.

[17] E. Surya and C. Diviya, "A Survey on Symmetric Key Encryption Algorithms," International Journal of Computer Science & Communication Networks, vol. 2, pp. 475-477, 2012.

[18] I. Basharat, F. Azam, and A. W. Muzaffar, "Database security and encryption: A survey study," International Journal of Computer Applications (0975–888) Volume, 2012.

[19] E. Shmueli, R. Vaisenberg, E. Gudes, and Y. Elovici, "Implementing a database encryption solution, design and implementation issues," Computers & security, vol. 44, pp. 33-50, 2014.

[20] A. Shulman and C. Co-founder, "Top Ten Database Security Threats," How to Mitigate the Most Significant Database Vulnerabilities, 2006.

[21] A. Chuvakin, K. Schmidt, and C. Phillips, Logging and log management: the authoritative guide to understanding the concepts surrounding logging and log management: Newnes, 2012.

[22] OWASP, " Error Handling, Auditing and Logging," 12 May 2013, https://www.owasp.org/index.php/Error_Handling,_Auditing_and_Logging.