Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, October 29-30, 2015

Secure Coding for Safety I&C Systems on Nuclear Power Plants

Y.M. Kim[a*], T.H. KIM[b] and H. S. Park[a]
[a]*Korea Institute of Nuclear Safety, 62 Gwahak-ro, Yuseong-gu, Daejeon, Korea, 305-338*
[b]*Formal Works Inc., 110 Banpo-r, Seocho-gu, Seoul, Korea, 137-872*
[*]*Corresponding author: ymkim@kins.re.kr*

1. Introduction

The use of digital technology is increasing in I&C nuclear I&C systems. The use of digital equipment may improve their reliability and reduce maintenance costs. But, the design characteristics of nuclear I&C systems are becoming more complex and the possibility of cyber-attacks using software vulnerabilities has been increased. This paper addresses secure coding technologies which can reduce the software vulnerabilities and provides secure coding application guidelines for nuclear safety I&C systems.

2. Background

2.1 Software Error Lists

Much vulnerabilities which have been used by the attackers were caused by software errors. So the IT industries have gathered the information such as software errors and security weaknesses which might cause the software vulnerabilities.

CWE(Common Weakness Enumeration) provides a set of software weaknesses that is enabling effective selection and use of software security tools and services that can find these weaknesses in source code[1]. CWE provides over than 8 hundreds software programming defects, design defects and architecture defects which can introduce security vulnerabilities.

In 2011, MITRE published the top 25 most dangerous software errors[2]. The 2011 CWE/SANS Top 25 Most Dangerous Software Errors is a list of the most widespread and critical errors that can lead to serious vulnerabilities in software. They are dangerous because they will frequently allow attackers to completely take over the software, steal data, or prevent the software from working at all. The list is the result of collaboration between the SANS Institute, MITRE, and many top software security experts in the US and Europe.

The Common Cyber Security Vulnerabilities in ICS are the results of the ICS security program which was performed by DHS in U.S[3]. The U.S. Department of Homeland Security (DHS) National Cyber Security Division's Control Systems Security Program (CSSP) performs cyber security vendor assessments and asset owner cyber security evaluations to reduce risk and improve the security of ICS. The examples of the common ICS software/product weaknesses are improper input validation, poor code quality, permission, privileges, and assess controls, improper authentication, etc.

2.2 Secure Development Methodologies

Microsoft SDL(Security Development Lifecycle) is a software development process that helps developers build more secure software and address security compliance requirements[4]. The principals for developing secure software of the Microsoft are secure by design, secure by default, secure in deployment and communications. The Microsoft SDL is changed when the Microsoft's security policy is changed.

CLASP(Comprehensive, Lightweight Application Security Process) methodology was developed by Secure Software Inc[5]. CLASP is architectural, repeatable and predictable technique. CLASP provides detailed information such as best practices, the high-level security services, core security principles, role and activities for building secure software.

The SevenTouchpoint methodology was suggested by McGrow[6]. He suggested that the security functions and activities for each lifecycle phase can help reducing the software vulnerabilities. That methodology required the seven touch points which must managed by software developers. These are core review, architectural risk analysis, penetration testing, risk-based security tests, abuse cases, security requirement and security operation.

2.3 Secure Coding Standards

Secure coding standards they provides the coding rules during implementation phase for reducing software errors. Recently, government has recommended that the software for public should be developed with secure coding standard. Table 1 shows the differences between several coding standards[7-10].

3. Secure Coding Guidelines for safety I&C Systems

3.1 Applications for safety I&C Systems

Table 1. Differences between Coding Standards.

Coding Standard	C Standard	Security Standard	Safety Standard	International Standard
CWE	None/all	Yes	No	No
MISRA C2	C89	No	Yes	No
MISRA C3	C99	No	Yes	No
CERT C99	C99	Yes	No	No
CERT C11	C11	Yes	No	No
ISO/IEC TS 17961	C11	Yes	No	Yes
MPAS C[8]	N/A	Yes	No	No
NUREG/ CR-6463	C90	No	Yes	Yes

For applying secure coding for nuclear safety I&C system, developers shall select the security defects based on software error lists related to nuclear safety I&C systems. The countermeasures for removing these security defects should be defined and be applied to the target systems. Fig. 1 shows the applying methods for removing security defects.

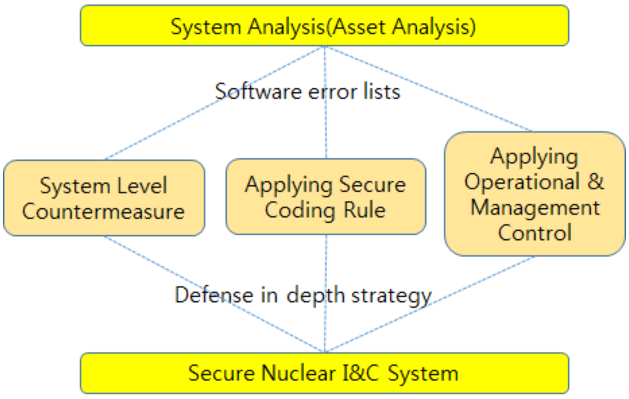
System Analysis(Asset Analysis)
Software error lists
System Level Countermeasure
Applying Secure Coding Rule
Applying Operational & Management Control
Defense in depth strategy
Secure Nuclear I&C System

Fig. 1. Applying methods for secure coding

The security defects can be removed not only by coding rules during implementation phase but also by operational and management controls during operation phase. The countermeasures for security defects can be developed by the following points.

- design and architecture: design architecture and detail contents for system
- implementation(coding): coding based on system design
- operation: providing services after finishing system development

Table 2 shows the examples of the countermeasures for security defects.

Table 2. Examples of the Countermeasure for Security Defects

Software Defects		Execute coding using improper privilege
Description		Executing code with higher privilege than required least privilege can lead to software defects or affect other defects triggering.
Phase	Design and Architecture	- execute code using least privilege - identify functions and assets which require special privilege and defer the privilege elevation point as late as possible.
	Implementation (Coding)	- validate inputs before executing user inputs with special privilege - allocate special privilege specifically such as read, write and network socket, etc.
	Operation	- operate with security configuration for service system

3.2 Guidelines for Secure Coding

The secure coding is a set of activities which can reduce the software errors which cause the software vulnerabilities. Guidelines for secure coding should be defined for reducing software errors. Followings are guidelines which must be included essentially.

- define the software errors which can reduce security vulnerabilities in project initial phase
- train the developers about the identified software errors and countermeasures
- implement software with consideration for software errors which can cause the security vulnerability
- manage each applying items with consideration for characteristics of the security vulnerabilities and the target system

4. Conclusions

Software defects, bugs and logic flaws have been consistently the primary causes of software vulnerabilities which can introduce security vulnerabilities. In this study, we described a applying methods for secure coding which can reduce the software vulnerabilities. Software defects lists, countermeasures for each defect and coding rules can be applied properly depending on target system's condition.

We expect that the results of this study can help developing the secure coding guidelines and significantly reducing or eliminating vulnerabilities in nuclear safety I&C software.

Acknowledgements

This work was supported by the Nuclear Safety Research Program through the Korea Radiation Safety Foundation(KORSAFe), granted financial resource from the Nuclear Safety and Security Commission (NSSC), Republic of Korea (No. 1305003-0315-SB130)

REFERENCES

[1]CWE(Common Weakness Enumeration), http://cwe. mitre.org, Visited 2014
[2]2011 CWE/SANS Top 25 Most Dangerous Software Errors, http://cwe.mitre.org/top25/, MITRE, 2011
[3]Common Cybersecurity vulnerabilities in Industrial Control Systems, ICS-CERT, 2011
[4]Microsoft Security Development Lifecyle, http://www. microsoft.com/security/sdl, Visited 2014
[5]CLASP Project, https://www.owasp.org/ index.php/ CLASP, visited 2014
[6] SOFTWARE SECURITY: BUILDING SECURITY IN, Addison-Wesley, 2006
[7]CERT C Coding Standard, https://www.securecoding.cert org/confluence/display/seccode /CERT+C+Coding+Standard, Visited 2014
[8]C Secure Coding Guide, Ministry of Public Administration and Security, 2011
[9]ISO/IEC TS 17961, “Information Technology − Programming languages, their environments and system software interfaces − C Secure Coding Rules, 2013
[10]MISRA-C: 2004 Guidelines for the use of the C language in critical systems , 2004