Analysis of the Potential Malicious Code on Control Systems

Wan-Hui Tseng and Chin-Feng Fan
Computer Science and Engineering Dept., Yuan-Ze University, Taiwan
s929403@mail.yzu.edu.tw
Steven M. H. Chen and Tsung-Chieh Cheng
Institute of Nuclear Energy Research, AEC, Taiwan

ABSTRACT

After the 911 terrorism attacks, the America government thoroughly investigated the infrastructure environment and found that the automated control system is a vulnerable point. In order to ensure the control system programming security, the malicious program issue must be investigated. Thus, our research focuses on the control system malicious code. This paper classified the malicious code based on control system characteristics. We identified the potential malicious code classifications, such as time-dependent, data-dependent, behavior-dependent, input-dependent, violate a certain theorem, deviation's error, isolated codes, change of privilege/authentication, anomalous file access, access of text segment as data, and so on. Finally, we develop an example to construct a malicious program on automated control system according to our classification. It demonstrates that the malicious code written by an insider and planted into the control system program is possible. So, this research is the first step towards detecting the malicious code on the control systems.

Keywords: control systems security, malicious code classification

1. Introduction

Since the 911 terrorism attacks in 2001, many critical infrastructure vulnerabilities are revealed. These critical infrastructures include nuclear, power, water, telecommunication, energy, and chemical industry, etc. Thus, the America government thoroughly investigated the infrastructure environment and found that the automated control system security is a weakness point. The control system includes many security vulnerabilities and it can be attacked easily by terrorists. Moreover, once the terrorism attack happens, it will lead to a serious consequence.

The reasons why the control system is the security vulnerability include two key points. First, the control system always executes in a closed-environment. It is hard to intrude the independent network form outside. Second, if someone wants to implement malicious actions, he or she must possess the domain knowledge to operate the control system. Comparatively the business information system security, the control system security has been always neglected in a long time. However, the 911 terrorism attacks represents that the terrorist resort to every conceivable means to achieve their goals.

During these years, many critical security events on infrastructures are happened[1,2,3,5,7,9,11,12]. Reed [11] wrote "In order to disrupt the Soviet gas supply, its hard currency earnings from the West, and the internal Russian economy, the pipeline software that was to run the pumps, turbines, and valves was programmed to go haywire, after a decent interval, to reset pump speeds

and valve settings to produce pressures far beyond those acceptable to pipeline joints and welds." The power-plant case [1] showed the control system programming of Inalina nuclear power-plant in Lithuanian was embedded the logic bomb by an insider who nursed a grievance against the government. Thus, it can be seen that the malicious code written by an insider is a severely vulnerable point on control system security. Therefore, the malicious code detection is an important issue for control system security which needs to be resolved immediately.

In this paper, our research focuses on the control system malicious code classification. There are many different characteristics between control system and general business system. The control system emphasizes the real-time implementation, and operates with feedback loop, sensor inputs, actuator actions, and operator commands to execute the operation. Thus, the potential malicious programs on control systems are different from those on other systems[4,10].

The rest of the paper is organized in the following way. Background will be briefly described first in Section 2. Section 3 presents the malicious code classification on control systems. Section 4 shows the case study that the malicious code on the nuclear power plant may happen probably. Finally, the future work will be given in Section 5.

2. Related Background

The research on malicious code classification includes McGraw [4] and R. W. Lo et al. [10]. McGraw and Morrisett [4] classified the malicious code into five categories: viruses, worms, trojan horses, back doors, and spywares. Combining two or more of these malicious code categories can lead to powerful attacks. R. W. Lo et al. [10] defined tell-tale signs and used them to detect the malicious code by program slicing [8] and data-flow technique. These tell-tale signs include three groups: tell-tale signs identified by program slicing, tell-tale based on data-flow information, and program-specific tell-tale signs. However, all of the research on malicious code classification deals with general programming code. It seems that there is no research so far to classify malicious code specifically for control systems. Thus, differing from these studies, our research focuses on control system malicious code classification.

Insider attack is an important issue on control systems. The primary threats to computer systems have traditionally been the insider attacks[6]. According to the U.S. Secret Service and CERT Coordination Center/SEI statistics[13,14], the insider threat can lead to serious damages. The report [13] shows that there have been the insider attacks in several critical infrastructure sectors, including baking and finance, information and telecommunication, energy, chemical industry and hazardous materials, water, etc. Table 1 represents the insider incidents during 1996 to 2002. Table 2 shows the percentage of organizations experiencing financial losses. Eighty-one percent of the organizations experienced a negative financial impact as a result of the insiders' activities. The losses ranged from a reported low of $500 to a reported high of "tens of millions of dollars."[13] They show that the insider threat can result in powerful attacks and severe loss. Thus, the insider attack consequence can not be neglected. So, to prevent the insider attack will be an important research for control systems security.

Table 1: Insider incidents by year of initial damage[13]

YEAR	Number of Incidents
1996	9
1997	5
1998	8
1999	7
2000	8
2001	6
2002	6

Table 2: The percentage of organizations experiencing financial losses[13]

Percentage of Organizations	Financial Loss
42	$1 - $20,000
9	$20,001 - $50,000
11	$50,001 - $100,000
2	$100,001 - $200,000
7	$200,001 - $300,000
9	$1,000,001 - $5,000,000
2	Greater than $10,000,000

3. Our malicious code classifications on control systems

The malicious code classifications are widely investigated on general business system. However, the critical control system security is always ignored. The potential insider attacks and the vulnerabilities of the control systems must be dealt with. In order to ensure the control system programming security, the malicious program issue must be investigated. In the control system, a single malicious action may not cause serious hazards. It always takes two or more instances of these malicious code categories to lead to a violent attack. Therefore, in this section, base on control system characteristics, we will analyze the potential malicious code. The following distinguishing features of control systems are identified:

- Real-time implement action
- Feedback loop
- Sensor inputs
- Actuator actions
- Keeping stable state
- Require sequential control
- Involved with operator

According to the above mentioned characteristics and existing research, we identified the following potential malicious code classifications for control systems.

(1) Time-dependent

Control systems always emphasize real-time operations. Hence, the timing-related problems are critical for them.

a. *Time-bomb*: the meaning of a time-bomb is that the malicious code is triggered at a certain time.

b. *Early or delayed message/action*: in generally, a message or an action directly influences the actual operations. An early or a delayed message/action will change the sequential operation of the control system. Thus, to keep the right control sequences is important.

c. *Race condition/Synchronization*: a race condition occurs when multiple processes access and manipulate the same data concurrently, and the outcome of the execution depends on the particular order in which the access takes place.

(2) Data-dependent (value-related)

Some of the control system operations depend on data setting. For example, for a car cruise control system, the speed set-point will influence the safety of a vehicle. The malicious code may change the critical data, and cause terrible damages.

a. *Display inconsistency*: the distances between the plants and the control room are always very far. Therefore, the monitor in the control room is a key reference basis to the operator. Once the malicious code counterfeits the actual result,

it will confuse the operator to perform right actions.

b. *Change alert set-point*: the control systems include several critical alert set-points. The malicious code may change that setting to invalidate the alert.

c. *Sending/receiving fake data*: on control systems the action depends on the sending/receiving data. Once the sensor and actuator send or receive the fake data, it would affect the proper actions.

d. *Getting fake feedback data*: the control system maybe be open loop systems or closed loop systems. Thus, to get the fake feedback data would cause the system in an unstable state.

e. *Independent/isolated data*: the anomalous data flow may include the potential malicious actions, such as use of undefined variable, use of extra data.

(3) Behavior-dependent (action-related)

Actuator actions affect the actual execution. Thus, the malicious actions will destroy the normal operation. Such actions include adding extra malicious actions, ignoring critical actions, executing wrong actions and mixing up execution sequences.

(4) Input-dependent

The malicious action maybe trigger when certain inputs are encountered. The insider may add abnormal actions in the program triggered by the particular inputs.

(5) Violate a certain theorem/rule/relation/invariant

The control system always keeps the system in a stable state. The system can not violate certain theorems, such as conservation of mass, the law of the conservation of energy, and so on.

(6) Deviation's errors

In the control system, the damages would not occur immediately, but they may result from accumulated deviation actions over a long period.

In addition to these categories, we may also include existing research results on the malicious code classifications. The **isolated codes, change of privilege/authentication, anomalous file access, and access of text segment as data** are also considered.

4. Case Study

As mentioned in Section 1, control system security has been always ignored over a long time. The reasons include that a control system always executes in a closed-environment and the hacker must possess the domain knowledge to operate it. However, in this section, we revealed that a control system may contain a malicious program written by an insider. The malicious actions may be triggered by malicious code of two or more of the above categories to make the accident happen.

In this section, we develop the malicious code on the Recirculation Flow Control System (RFC) of a power plant as a case study to demonstrate the feasibility of insider attacks on control systems. The RFC provides a mean to change the reactor power level by adjusting the recirculation flow over a range of power and flow at an established control rod pattern. The RFC sends the speed demand to control the physical processes. The operation results, core flow value and power value, are sent back to the console and displayed on the control monitor. The sketch is shown in Fig. 1.

In order to perform improper actions, the malicious codes are inserted into the control system

programs based on our classifications. We show a scenario combining two malicious code categories, which are *sending the fake data* and *display inconsistency. Sending the fake data* is performed to change the speed demand. As soon as the value of the speed demand is greater than 80%, then the fake speed demand value (110%) will be sent. When the physical processes receive the fake speed demand value (110%), this malicious action attempts to crash the physical operation. However, the values of the core flow and power will response to the reality situation. Therefore, the *display inconsistency* must be considered. It can change the actual values of the core flow and power, and display the calculated values on the control console. The fake display will swindle the operator, so that he is unaware of this abnormal circumstance. The sketch of the malicious actions is shown in Fig. 2. The step 1 and step 2 are *sending the fake data*, and the step 5 and step 6 are *display inconsistency.*

We inserted the malicious codes into the control system programs and executed them over a period. The simulated result showed that the anomalous situation happened, but the operator did not observe any irregular system behavior. According to our malicious code categories, the control systems can contain all kinds of vulnerabilities. There are many combinations which can be considered to destroy a control system. It can be seen that the malicious code written by an insider is severe threat to the control system security. Thus, potentially malicious code on control systems is an urgent topic to be resolved.

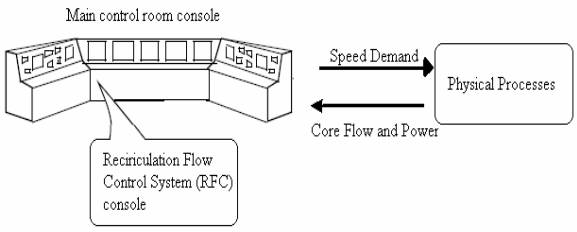

Fig. 1 The relation between RFC and physical processes

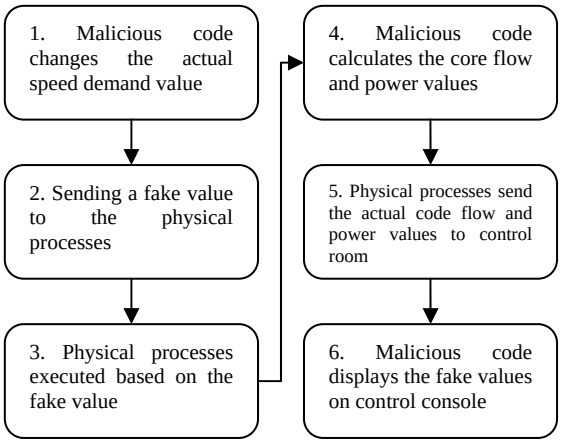

Fig. 2 The sketch of the malicious actions

5. Future Work

This research focuses on the control system malicious code classifications base on its characteristics. Our classifications include time-dependent, data-dependent, behavior-dependent, input-dependent, violate a certain theorem, deviation's error, isolated codes, change of privilege/authentication, anomalous file access, access of text segment as data, and so on. In Section 4, it demonstrates that the malicious code planted into a control system program written by an insider is possible. Therefore, in order to enhance the security of the control systems and reduce their vulnerability, the final goal is to discriminate the malicious code from the benign ones. Our research is the first step towards detecting the malicious code on control systems.

As to our future work, we will develop the malicious code detection patterns base on our classifications. Furthermore, we plan to combine static and dynamic analyze to detect the malicious code. In the static analysis, we will detect the malicious code using the program slicing and data flow techniques by defined detection patterns. On the other hand, some malicious actions can not be easily found in the static analysis.

Thus, dynamic analysis is needed in run-time monitoring. Finally, we plan to construct practical tools that address the automatic detection of the malicious code on the control systems.

Acknowledgement

This work was supported in part by graduate student scholarship by Atomic Energy Council, Taiwan.

Reference

[1] Lithuanian nuclear power-plant logic bomb detected, *Software Engineering Notes, Vol. 17, No.2*, 2006.

[2] Major power outage hits New York, other large cities, 2003.

[3] Calton, Taipei Subway Computer Crash, *The Risks Digest, Vol. 18, No. 17*, 1996.

[4] G.McGraw and G.Morrisett, Attacking malicious code: Report to the Infosec research council, *IEEE Software, 17(5)*, pp. 33-41, 2002.

[5] Kevin Poulsen, Slammer worm crashed Ohio nuke plant network, *SECURITYFOCUS NEWS, Security Focus*, 2003.

[6] Lawrence E.Bassham and W.Timothy Polk, Threat Assessment of Malicious Code and Human Threats, *National Instisut of Standards and Technology Computer Security Division*, 1994.

[7] Loney Matt, US software 'blew up Russian gas pipeline, *ZDNet, UK*, 2004.

[8] Mark Weiser, Program Slicing, *Proceedings of the Fifth International Conference on Software Engineering*, pp. 439-449, 1981.

[9] Mikhail B.Ignatyev, Analysis of the Threat of Cyber attacks to Major Transportation Control Systems in Russia Terrorism: Reducing Vulnerabilities and Improving Responses, *U.S - Russian Workshop Proceedings*, 2002.

[10] R.W.Lo, K.N.Levitt, and R.A.Olsson, MCF: A Malicious code Filter, *Computers and Security, 14(6)*, pp. 541-566, 1995.

[11] Thomas C.Reed, At the Abyss: An Insider's History of the Cold War, *Ballantine Books*, 2004.

[12] Tony Smith, Hacker Jailed for Revenge Attack, *http://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage*, 2001.

[13] U.S.Secret Service and CERT Coordination Center/SEI, Insider Threat Study: Computer System Savotage in Critical Infrastructure Sectors, 2005.

[14] U.S.Secret Service and CERT Coordination Center/SEI, Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector, 2004.