The Probable Cyber Attack Concept
Exploiting Interrupt Vulnerability in Nuclear Power Plants

Taehee Kim, Soomin Lim, and Sangwoo Kim
Cyber Security Division
Korea Institute of Nuclear Nonproliferations and Control
Daejeon, Korea
email: {kimtaehee, s2min, kjoey}@kinac.re.kr

***Abstract*—So far, cyber threats to nuclear power plants have remained an unexplored area. No single cyber attack has been reported that successfully degraded the safety function of a nuclear power plant. However, it is not guaranteed that nuclear power plants are completely safe from cyber attacks. This paper proposes a probable attack concept, which can disrupt the real-time and deterministic nature of instrumentation and control systems.**

Keywords-nuclear power plant; cyber threat; cyber attack; I&C system; preemptive OS; interrupt.

I. INTRODUCTION

Conventionally, Instrumentation and Control (I&C) systems are designed to protect nuclear power plants in terms of safety. This design concept has been implemented at the early stage of nuclear power plants and reinforced for decades. Sometimes, lessons are also learned from accidents [1][2] and they are fed to the design as improvements. As a result, I&C systems can protect the plants from various events, such as failures, errors, and even disasters.

Recently, cyber threats to nuclear power plants have received increased attention. The threats include hacking, viruses, Distributed Denial of Service (DDoS) attacks [3] and they are well-known in general Information Technology (IT) environments. No single cyber attack, however, has been reported that successfully disrupted the safety control of a nuclear power plant. The well-known Stuxnet [4] succeeded to attack the Iran uranium enrichment plant, not nuclear power plants. The feasibility of a successful attack and appropriate countermeasures still remain uncertain in nuclear power plants. Despite this uncertainty, the participants in the nuclear power industry take an optimistic view about the safety of nuclear power plants. They believe detection methods for safety events must be effective for cyber attacks. Actually, the periodic Cyclic Redundancy Checking (CRC) [5] used for detecting memory errors may be useful for detecting the fuzzing attacks [6] messing up the memory.

During a survey on the detection methods for cyber attacks, the usage of I&C systems have attracted our attention. The usage is a factor for reflecting how busy an Operating System (OS) is and it is usually disseminated to the OS itself and surrounding I&C systems. The usage can also be utilised for detecting cyber attacks; a malicious task injected into an OS may increase the usage value. We, however, found a vulnerability that can be invoked by interrupts.

This paper is structured as follows. In Section II, we describe the background: assumptions that need to be described for further discussion. Section III proposes the probable attack concept derived by our research. The difference between the proposed concept and the similar safety event, that is the busy OS case, is given in Section IV. Then, the precautions that may protect the I&C system from the proposed attack and their limits is also given in Section V. Finally, this paper is concluded with conclusions and future works in Section VI.

II. ASSUMPTIONS

This section provides assumptions for further discussion. Although we name this assumptions, we believe the given assumptions are based on the common nature of I&C systems for nuclear power plants.

A. Relatively Low Performance

Reliability is the most expected virtue of I&C systems in the nuclear power industry. One of the typical methodologies to calculate reliability is analyzing each components as we can see in MIL-HDBK-217F [7]. The application history, however, is the most powerful proof of reliability. The proven I&C systems in the actual nuclear power plants will be preferred in other plants.

Therefore, most nuclear power plants tend to adopt proven I&C systems, although they have relatively low performance. Having a long application history means that I&C systems had been adopted and developed for a long time. On occasion, the systems may have been developed several decades ago, and their CPUs are operated at a low speed of few MHz. In general, the CPUs are slower than personal computers or even cell phones.

B. Preemptive Operating System

The safety controls of nuclear power plants should handle safety events timely, that is, real-time, and deterministic manner.

 ISBN: 978-1-61208-661-3

"Real-time" means that safety controls always respond within the requested time limit. A safety control with time limit of *n* milliseconds should respond within *n* milliseconds. "Deterministic" means that responding of safety controls are always predictable. Decision factors that may change responds of the safety controls should be known in advance. Different or random outputs from a same safety control are not permitted if decision factors are not changed.

I&C systems with a preemptive OS can support real-time and deterministic nature [8][9] of the safety controls. The preemptive OS periodically scans all tasks and stops current task on executing if urgent task is waiting to be executed. The urgent task does not wait the termination of the other non-urgent tasks, and gets the right to be executed although the other non-urgent tasks are waiting. These periodic scans and exchanges of tasks are called as context switches. By these context switches, time limit is met and we can predict the execution order of tasks.

C. Interrputs

An interrupt is one of the well-known methodologies for data exchange between a CPU and peripherals. With an interrupt manner, a CPU executes tasks and does not care peripherals before they inform the CPU. On the other hand, with a polling manner, a CPU periodically stops executing tasks to check peripherals whether they want to exchange data with the CPU.

By their nature, an interrupt is more efficient method than a polling. Time to check peripherals caused by polling is wasted if the peripherals do not have data to exchange. Therefore, most digital systems, such as personal computers adopt an interrupt manner for exchanging data with peripherals. A mouse and a keyboard are representative examples.

Inside of I&C systems used for nuclear power industry, interrupts are preferred methods for data exchange, such as urgent switchover between redundant CPU modules for seamless operation and asynchronous serial communication for downloading tasks.

In this paper, we assume that I&C systems support several interrupts with their own priority.

D. Interfaces

The main purpose of I&C systems is to receive inputs from field devices, to process inputs, and to send outputs to where they are needed at. For this reason, I&C systems essentially have various interfaces; analog and digital, input and output.

Serial interfaces for Human and Machine Interfaces (HMIs), and Engineering Work Stations (EWSs) are representative examples in nuclear power plants. Basically I&C systems can execute their tasks of themselves, but they still need HMIs for observing operational values and manipulating configuration settings, and EWSs for managing control logics.

III. ATTACK CONCEPT

In this section, we propose the probable attack concept that exploits the interrupt vulnerability of I&C systems.

The attack concept is simple: to break the real-time and deterministic nature of I&C systems. In other words, disrupting tasks to be executed within time limits is the proposed attack concept. The following is a summary of how the tasks can be disrupted.

The I&C systems we assume in this paper have various interfaces. In general, these interfaces work in an asynchronous manner for a safety purpose. Unused devices are not connected with interfaces because the devices might be touched by an operator accidentally and then send unintended instructions to I&C systems. Electrical surges from unused devices might also cause malfunctions in I&C systems. This asynchronous nature means that interfaces are based on an interrupt manner. This asynchronous nature brings two implications. The first implication is interfaces are not occupied and are waiting devices. The second implication is that interfaces are driven with an interrupt manner.

The first step for the attack is connecting devices to those unoccupied and interrupt-driven interfaces. For any device, it is possible to enable continuous interrupts. Typical example is a bad USB device. A pair of a dongle and the laptop with software having ability to send serial data automatically is another example.

The second step is enabling interrupts with high priorities continuously to the targeted I&C system through the connected interface. Data contents and an application layer protocol for enabling interrupts do not matter. Every data will be delivered to interrupt handlers whether they are valid or not. An interrupt is a just hardware-level signal used for informing an OS and it cannot interpret data contents. Therefore, even invalid data cannot be filtered and should be delivered to interrupt handlers. These interrupt handlers will consume precious time and disrupt other interrupts with lower priorities.

At the final step, a time tick, that is a kind of interrupt, is delayed by interrupts enabled by the attack. The main purposes of a time tick is measuring the time flow and calling context switches periodically. Therefore, delayed time ticks cause delayed context switches. It means time limits of tasks cannot be met and we cannot predict the execution order of tasks in advance, as written in Section II.B. Finally, the I&C system with delayed time ticks will not work in a timely and deterministic manner.

IV. THE BLIND SPOT OF THE USAGE

The busy OS case given in Section IV.A may be confused with the proposed attack because tasks suffer from delays in that case. However, it is totally different from the proposed attack concept in terms of intention. The proposed attack is a hostile action with intention, while the busy OS case occurs with no intention. It is basically closer to programming errors, such as infinite loops or congestions, not filtered during tests. Based on this difference, the busy OS case can be detected by the usage and thus it is detectable while the proposed attack concept cannot be detectable.

To explain the difference above mentioned, the usage calculation process is described in Section IV.B. Then, it is

 ISBN: 978-1-61208-661-3

followed by Section IV.C which describes the blind spot hiding the proposed attack from the detection.

A. Busy OS Case

Tasks executed by I&C systems have various branches and each branch has its own work flows. Some branches may have simple operations while other branches have heavy operations. For example, a task may just observe a certain value before it exceeds thresholds, but the task may write the trend of the value on a slow flash memory with very dense interval time for a future audit. Depend on which branch is being executed, a CPU may be busy or not busy.

If the busyness of an OS, or the usage, reaches 100%, the I&C system cannot afford additional work imposed by a task jumping to a heavy branch. In this case, the safety controls supported by I&C systems cannot work in a timely and deterministic manner. In other words, they are compromised.

B. The Usage Calculation Process

I&C systems keep their own value called the usage, to detect compromised I&C systems by the busy OS case. The usage is the factor reflecting the busyness of an OS. Nested I&C systems observe the usage of each other and I&C systems with a high usage value will be regarded as compromised.

The usage can be calculated by measuring how long time the idle task is executed within the given time. This calculation can be implemented as follows.

1) Calculating G: In initializing phase, an OS does nothing except increasing a variable whthin the given time T and keeps it at the global variable G.

2) Start the OS: After the OS completes initializing phase, a local variable L in the idle task is set to zero and scheduling is started in earnest.

3) Calcuating L: For the given time T, L is continuously increased when the idle task is on execution, while it is not increased when the other tasks are on execution.

4) Comparing L and G: After the given time T, by comparing G and L, the OS can know how long time the idle task was executed within given time. Then L is reset to zero.

5) Repeat: the OS repeats 2) ~ 4).

The above calculation can be expressed by

$$\text{Usage for } T = (1 - (L / G)) \times 100. \quad (1)$$

The usage value will stay low, if the idle task is executed longer, while it will become high, if the idle task is executed shorter. When it is 100%, a I&C system is fully busy and cannot afford additional work.

C. Blind Spot of Usage

The calculation given in Section IV.B seems quite reasonable and clear. The serious trap, however, is lying on (1) because T cannot be measured. T stands for actual and absolute time, but the OS does not have the tool to measure such time in general. Instead, the OS counts time ticks to measure T. According to this, P is pre-calculated by (2) and hard-coded into the OS.

$$P = T / \text{Interval Time between Time Ticks} \quad (2)$$

Then, (1) should be updated by

$$T' = \text{Interval Time between Time Ticks} \times P, \quad (3)$$

$$\text{Usage for } T' = (1 - (L / G)) \times 100. \quad (4)$$

The proposed attack concept given in Section III extends the interval time among time ticks and makes T' longer by (3). The extended length is same with time spent by the interrupt handlers called by the attack. L and G stay same regardless of attack, because L is not increased in interrupt handler and G is calculated before attack. As a result, the usage value becomes lower than the actual busyness of the OS by (4). This is the obvious blind spot of the usage.

More seriously, nested I&C systems described in Section IV.B cannot detect compromised I&C systems because the usage value will stay low due the blind spot.

V. Precautions and Limits

In this section, a few existing precautions are given. These precautions may be useful to protect I&C system from the proposed attack. They, however, cannot provide complete protection.

A. Blocking Interrupt

In the attack steps given in Section III, the time spent by interrupt handlers may be short in well-designed I&C systems. Furthermore, continuous invalid data received in a short period may be considered as noises or attacks. Then, they will be discarded without an interpretation. This may help to mitigate the attack but cannot provide the complete protection.

The only complete solution is to block the interrupt channel connected with the interface is being attacked. However, it may also block the other essential devices for operation and a future forensic procedure. Once they have been blocked, the targeted I&C systems may need factory reset, which initializes inside of I&C systems and destroys evidences

B. Watchdog

A watchdog [10] is a kind of timer for increasing reliability. It waits to be kicked (to receive a signal from outside) for the pre-defined time limit. If it is not kicked within the time limit, it releases the warning signal. A watchdog is driven by the inside time source and independent from time ticks and interrupts.

In terms of response time, however, a watchdog cannot provide the complete protection. The I&C systems we assume are operated by a preemptive OS, which works in "within time" manner, not "on time". It means that the time limit of a watchdog should have enough margin.

 ISBN: 978-1-61208-661-3

Furthermore, attacks may be designed to delay time tick by *n-1* milliseconds, when a watchdog is set to wait *n* milliseconds.

C. Real-time Clock Component

I&C systems may have other common time sources, such as real-time clock components [11]. It can measure actual time flow independently with time ticks.

Nevertheless, they cannot provide the complete protection because their time scale, that is hour, minute, and second, is not precise enough for context switches in a preemptive OS. The time scale should to be few milliseconds at minimum for efficient tasks scheduling. Because of this, real-time clock components are preferred for displaying the current time.

D. Physical Acess Control for Interfaces

Well-known regulations [12][13] compel nuclear power plants to protect interfaces from unauthorized accesses. The actual protection strategy, however, is implemented by periodic security audits rather than technical security controls. This strategy is inevitable for many legacy systems in nuclear power plants because they were not designed with security considerations. Therefore, interfaces are left to be attacked between audits.

VI. Conclusions and Future Works

In this paper, we proposed the probable attack concept exploiting interrupt vulnerability. The proposed attack delays time ticks first, and then context switches of a preemptive OS. As a result, the real-time and deterministic nature of I&C system are not guaranteed. Furthermore, nested I&C systems for safety controls cannot detect the attack even if the targeted I&C systems are compromised due to the usage blind spot. Existing precautions and their limit analyses were also given.

This paper does not include actual experiments and the feasibility is not proven. However, we believe that nuclear power plants should be protected from any possibility to degrade the safety level.

In the future, we will perform experiments on our test bed. If it is observed that the proposed attack has any influence on the test bed, we will try to find mitigating measures.

References

[1] United States of America, Nuclear Regulatory Commision, *Backgrounder on the Three Mile Island Accident*, 2013. [Oline]. Available from: https://www.nrc.gov. [Accessed: Jul. 2018].

[2] United States of America, Nuclear Regulatory Commision, *Backgrounder on NRC Response to Lessons Learned from Fukushima*, 2015. [Oline]. Available from: https://www.nrc.gov. [Accessed: Jul. 2018].

[3] S. T. Zargar, J. Joshi, and D. Tipper, "A Survey of Defence Mechanisms Against Distributed Denial of Service Flooding Attacks," IEEE Communications Surveys & Tutorials, vol. 15, issue. 4, pp. 2046-2069, 2013.

[4] R. Langner, "Stuxnet: Dissecting a Cyberwarfare Weapon," IEEE Security & Privacy, vol. 9, issue. 3, pp. 49-51, 2011.

[5] A. B. Marton and T. K. Frambs, "A Cyclic Redundancy Checking Algorithm," *Honeywell Computer Journal,* vol. 5, no. 3, 1971.

[6] M. Sutton, A. Greene, and P.amini, Fuzzing: Brute Force Vulnerability Discovery, United States: Addison-Wesley Professional, 2007.

[7] J. W. Harms, "Revision of MIL-HDBK-217, Relaibility Prediction of Electronic Equipment," Proc. IEEE Symp. Annual Reliablity and Maintainability, IEEE Press, 2010, dio:10.1109/RAMS.2010.5448046

[8] International Electrotechnical Commision, *Nuclear power plants – Instumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions*, 2006.

[9] Institute of Electrical and Electronics Engineers, *IEEE Standard Criteria for Programmable Digital Devices in Safety Systems of Nuclear Power Generating Stations*, 2016.

[10] Maxim Integrated, "MAX6814 5-Pin Watchdog Timer Cuircuit," 2014. [Online]. Available from: https://www.maximintegrated.com. [Accessed: Jul. 2018].

[11] Maxim Integrated, "DS1685/DS1687 3V/5V Real-Time Clocks," 2012. [Online]. Available from: https://www.maximintegrated.com. [Accessed: Jul. 2018].

[12] United States of America, Nuclear Regulatory Commision, *Reguatory Guide 5.71: Cyber Security Programs for Nuclear Facilities*, 2010. [Oline]. Available from: https://www.nrc.gov. [Accessed: Jul. 2018].

[13] Republic of Korea, Korea Institue of Nuclear Nonproliferations and Control, *Reguatory Standard – Security for Computer and Information System of Nuclear Facilities*, 2016

 ISBN: 978-1-61208-661-3