Efforts for Enhancing Security in Control Systems

Katsuhiro Takamatsu [*1] *Tsuyoshi Katou* [*1] *Hiroyuki Makabe* [*2]

To ensure the security of products, it is important to take measures in each phase of their development and also to implement security functions in the products themselves. This paper introduces the security development lifecycle that Yokogawa applies to each development phase of its system products. This paper also describes examples of security functions implemented in control systems. These are OS hardening, optimization of antivirus software, and security measures implemented in Vnet/IP to maintain the security of control networks even in an open network environment.

Security is also being enhanced in recent security certification programs. In particular, the ISASecure certification program aims to become an IEC standard and it is attracting attention. In an effort to keep up with this trend, Yokogawa has acquired the certifications of this program for its core products, CENTUM VP and ProSafe-RS.

INTRODUCTION

Security has long been required for control systems, and Yokogawa has been working on security issues. However, its efforts mainly focused on implementing apparently and immediately effective security functions and integrating third party security products.

Meanwhile, in information systems which have been exposed to security threats much earlier than control systems, they have been suffering from attacks aimed at their weak points. These weak points possibly abused for attacks are called vulnerabilities, and are disclosed widely to the public. These days, such attacks are expanding over control systems, and the disclosure of the vulnerabilities of control system products is also increasing.

For product security, efforts for developing products free from vulnerability is as important as implementing security functions; however, these efforts have been avoided because their effectiveness is not easy to evaluate and it often takes time.

To provide more secure products in accordance with the market trend, Yokogawa started efforts to not introducing vulnerabilities into its products, mainly controllers. This paper introduces the secure development lifecycle as efforts for not introducing vulnerabilities and then describes examples of efforts in terms of the security function aspect. These examples include OS hardening by an IT security tool and optimization of antivirus software as examples of security measures for platforms of products, and security measures in the Vnet/IP, a Yokogawa control network, as an example of security functions implemented in a product. Finally, this paper introduces Yokogawa's acquisition of the ISASecure Embedded Device Security Assurance (EDSA) certification owing to those efforts.

*1 System Integration D&E Dept.,
Global Development Center, IA Platform Business Headquarters
*2 Digital Hardware D&E Dept.,
Global Development Center, IA Platform Business Headquarters

EFFORTS TO NOT INTRODUCE VULNERABILITIES

Outline of Secure Development Lifecycle

To not introduce vulnerabilities into products, the development organization must introduce a secure development lifecycle and its engineers must acquire appropriate skills. The secure development lifecycle is a concept to take security measures in each phase of development processes. It aims to minimize vulnerabilities generated in deliverables of each development phase and detect them in as early phases as possible.

Although development may be conducted in various ways, this section describes an approach of the secure development lifecycle based on the development phases shown in Figure 1.

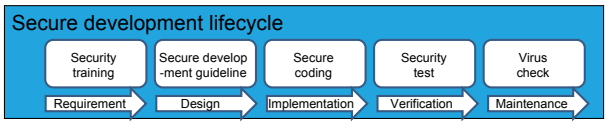

Figure 1 Procedures in secure development lifecycle

Requirement Phase

One of the main purposes of this phase is to help development engineers to understand what vulnerabilities are, how they are introduced into products, what measures are possible for them, and so on. For this purpose, security

training is given to engineers assigned to a development team. Another purpose of this phase is to establish a development scheme in which a third person participates to ensure that security is properly taken into account during the development process. As a third person, a security expert independent of development is assigned. He/she checks whether security requirements are properly followed throughout the project through security reviews in each phase, and others.

Design Phase

The main purpose of this phase is to minimize vulnerabilities introduced in designing as much as possible. For this purpose, a secure development guideline was created that describes security viewpoints to which developers must pay attention. All developers are required to follow this guideline. After functional design, threat analysis is conducted, in which interfaces between modules, in particular interfaces with the outside such as interfaces with networks and file I/O interfaces, are listed and possible threats are analyzed on them. The STRIDE model [(1)] is used for clarifying possible threats. The STRIDE model classifies threats into six types and examines the threats in each type to exhaustively find threats. All threats are rated based on the DREAD model [(1)]. When any possible threats in interfaces facing the outside are scored higher than the criteria specified for each product, measures are taken regarding them. The DREAD model scores threats from five viewpoints, minimizing variation among scores by different raters.

Implementation Phase

The main purpose of this phase is to reduce vulnerabilities introduced in implementation as much as possible. For this purpose, a coding guide has been created that describes viewpoints that need attention during coding. The guide also defines application programming interfaces (APIs) to be avoided. The implemented codes are reviewed by other engineers, checked by using a static code analysis tool, and corrected if any problems are found.

Verification Phase

The main purpose of this phase is to verify that products have no known vulnerabilities and that implemented security measures are effective. Possible attacks identified during the threat analysis are actually tried on the products to verify the effectiveness of the measures. Abnormal packets of TCP or Ethernet frame are created and sent to products to verify that new vulnerabilities have not been created in this development. This test is called fuzzing. All plug-ins of Nessus, a vulnerability scanner, are used to verify that products are free from known vulnerabilities.

Maintenance Phase

The main purpose of this phase is to verify that products for shipping are not infected with viruses and to prepare respond guidelines and frameworks in case vulnerabilities are found after shipping. For shipping, digital signatures are added to the developed modules to indicate that programs are made by Yokogawa and are not malicious ones, and, three virus check software tools are used to check viruses in the final source code and the master media for shipping. To respond after shipping, Yokogawa has established a coordination center to monitor vulnerabilities found in the market and accept reports on vulnerabilities, and has recently made it possible for users to post those reports by using a Web browser. The coordination center acquires information regarding vulnerabilities from the Japan Computer Emergency Response Team Coordination Center (JPCERT/CC), and can perform its own surveys. For the vulnerabilities found in the market, countermeasures are decided depending on the scores derived from their threat analyses.

EFFORTS IN TERMS OF SECURITY FUNCTION ASPECT

Security Functions in Individual Products of Control Systems

These days, efforts to improve security of the entire system are required in which not only the security functions of products themselves but also those of the platforms of products, such of OSs and security software provided by third parties, are emphasized. As examples of security measures for platforms of products, this section introduces OS hardening by IT security tools and optimization of McAfee antivirus software. As an example of security functions implemented in a product, the security of Vnet/IP is introduced. This plays the most important role from the viewpoint of the security of controllers.

IT Security Tool

The Windows OS has various functions, but those not used for control system products can be disabled to block vulnerabilities in those functions. In addition, the proper setting of OS security functions can harden the system without affecting system operation. It is possible to set them on the tools provided by the OS without using a dedicated tool. However, required items are wide-ranging and the procedure is often complicated, easily causing setting errors.

Yokogawa's IT security tool offers three levels of security to meet various customers' operation needs. The legacy level emphasizes functional compatibility with the status quo. When the standard level is selected, the minimum security items applicable for most user environments are set. When the strengthened level is selected, the security items include those that must be selected due to the user environment. An example of the included items is the one for prohibiting log-on in the case a user fails to log-on a certain number of times. The security setting of the OS is completed only by selecting a security level.

Firstly, the system sets multiple security items to the most secure level. Then each product reduces the security levels for the security items to those needed for the product. This makes it possible for users to avoid software failures caused by security settings even when multiple control system products

are installed in one PC.

This IT security tool reduces setting errors and other human errors and eliminates vulnerabilities caused by these errors.

Optimization of Antivirus Software

Although antivirus software has been widely used, many of the folders in which control system products are installed are excluded from its scanning target. This is because control system failures due to misdetections and performance degradation due to virus scans are both unacceptable. However, many users want to lift this exclusion because viruses may slip into these folders at any time but they cannot be detected. Since there is a risk of misdetection or performance degradation, the exclusion cannot be lifted completely. Yokogawa has made a partnership agreement with McAfee, Inc. and is working hard to reduce the infection risk. Examples of the measures developed for this purpose are described below. These measures will be gradually applied to control system products.

Measures against performance degradation

McAfee's antivirus software has various functions, some of which are unnecessary for control systems or degrade their performance significantly. To fit these functions to the characteristics of control systems, their parameters were optimized. In addition, performance tests were performed to identify processes that degrade in performance. By defining special scanning specific to those processes, the exclusion range was reduced from folders to executable files, thus securing performance.

Measures against misdetection

For static files not changed during the execution of control system products, verification to find misdetections is conducted by performing virus scans using pattern files prior to McAfee's release. Misdetection can be reduced by removing those misdetections found. However, this method is not applicable to dynamic files changed during the execution of control system products, because they are changed after the verification mentioned above. Thus, the extensions of those dynamic files were registered to exclude them from the scanning target. This measure also reduces the exclusion range from folders to files with certain extensions, reducing the risk of misdetection.

Security of Vnet/IP

The Vnet/IP used in Yokogawa's production control systems and safety instrumented systems is a control network based on Ethernet technology. Use of the general-purpose open protocol enables the building of low-cost and highly extensible systems by using commercially available layer-2 switches, layer-3 switches and others. Meanwhile, applying a popular Ethernet interface tends to cause the systems to be targets of malicious attacks from the outside via networks. Among various security threats, spoofing, falsification and DoS attacks can significantly impact on controllers in the systems.

Spoofing and falsification are attacks in which attackers act as an authorized person and perform unauthorized operations, and may cause unexpected control of the systems. One of the countermeasures against these attacks is authentication, in which a sender and receiver share a key known only to them by periodically exchanging it. An authentication code generated from this key and the communication data is encapsulated in communication packets to discriminate them from malicious packets sent by attackers. However, it takes a certain time until the start of communication after the end of the key exchange, and usual security technologies cannot prevent the interval unavailable for communication due to the periodical key exchange. This makes the communication response time periodically longer at every exchange process, and causes the lowering of real-time property. To resolve this problem, Vnet/IP uses a key exchange method that enables secure continuous communication even during periodic key updating processes.

In a redundant system, when an active controller fails, a standby controller takes over the control processing of the active controller, and when one communication channel is disconnected, communication is switched to another channel. In these cases, communication cannot be restarted quickly if the key is exchanged after the switchover. In the Vnet/IP, IP addresses are assigned to all ports of the controllers constituting a redundant system, and key exchanges are constantly performed with each port independently, making it possible to restart communication immediately after the switchover of the controller or communication channel.

A DoS attack is an attempt to disturb normal operation by flooding accesses to a target. If a controller is targeted, this attack overloads the controller and prevents it from executing normal plant control processing. These attacks are usually avoided by adding dedicated external devices or discarding packets at the application layer. However, either measure lowers real-time property, because external devices cause communication delay and discarding packets at the application layer increases the CPU load.

To resolve this problem, various measures are implemented. The controller is equipped with two CPUs: one is for control and the other is for communication, so that the load on the communication layer does not affect the control processing. Unnecessary packets are discarded at the lower levels of the communication layer to reduce the loads. If one of the duplexed channels receives more packets than predetermined amounts, the communication through its channel is stopped for a certain time, and communication is continued through another channel.

ACQUISITION OF ISASecure EDSA CERTIFICATION

Outline of ISASecure EDSA Certification

The ISASecure EDSA certification program certifies the security of embedded devices such as built-in controllers on the basis of the ISA/IEC62443 standard frameworks. A third-party organization assesses whether security is ensured

in embedded devices from the following three technical elements.

- Software development security assessment (SDSA): whether security is considered during the development process
- Functional security assessment (FSA): whether required security functions are implemented
- Communication robustness testing (CRT): whether communication robustness is secured

The ISASecure EDSA certification program offers three certification levels depending on required security strength and higher levels needed to satisfy more requirements. For CRT, however, all requirements included in it must be satisfied regardless of the level.

Yokogawa's Efforts for Acquiring ISASecure EDSA Certification

The ISASecure EDSA certification requires passing the audit by a certification body and tests including CRT requirements on actual devices. SDSA contains 130 requirements even at level 1. The examination in the audit checks whether the development process includes all requirements corresponding to the level and whether the development is carried out following the development process. During the examination of the development, the existence of minutes and other evidence is checked. For FSA, the examination checks whether all functions defined in the requirements are implemented through looking into specifications and design documents. Some functions are verified using actual devices. As for CRT, all requirements are tested regardless of the certification level. The output of the devices is required not to become improper values during and after the test.

Yokogawa's security efforts described in previous sections were effective for acquiring ISASecure EDSA certification. Efforts for the secure development lifecycle, security functions not limited to those described in this paper, and security of Vnet/IP have contributed to satisfy SDSA, FSA, and CRT requirements, respectively.

ISASecure EDSA Certified Products

Yokogawa acquired ISASecure EDSA certification for the two controllers of the production control system and the safety instrumented system shown in Table 1.

Table 1 ISASecure EDSA-certified products

Product name	Version	Security level
CENTUM VP	R5.03.00	ISASecure EDSA 2010.1 Level1
ProSafe-RS	R3.02.10	ISASecure EDSA 2010.1 Level1

CONCLUSION

This paper described Yokogawa's security efforts to not introduce vulnerabilities and for implementing security functions in its products. These efforts are not based on Yokogawa's own viewpoint alone but consensus in the market; and this fact is demonstrated by the acquisition of ISASecure EDSA certification.

Yokogawa will continue to provide security measures from a systems perspective to cope with ever-changing security threats.

REFERENCES

(1) Michael Howard, David LeBlanc, Writing Secure Code, Second Edition, Microsoft Press, 2002