CPIS-compliance Security Requirement Analysis for Software Development

Jiang Lei[1, a], Yuan Jing[1], Ren Weihong[1, b, *], Zhao Tai[1]

[1] 3rd Research Institute of MPS - Information Classified Security Protection Evaluation Center, Beijing, 100142, China

[a]email: jianglei@cspec.org.cn, [b]email:renweih@cspec.org.cn, [*]corresponding author

Keywords: Cybersecurity; Threat Modeling; Classified Protection of Information System(CPIS);

Abstract. China CPIS policy is the fundamental policy of China cybersecurity strategy. Therefore, CPIS-compliance software solution is highly demanded to help organizations comply with CPIS associated policies and standards. In this paper, a CPIS-compliance security requirements analysis method for software development is proposed, which utilizes *TAG* to assess the attack and then develop response plan according to priority. Using *TAG*, controls from CPIS Baselines are introduced into software during design phase, which efficiently enhances and improves software security. A case is studied to illustrate the practicality and effectiveness of *TAG* method.

1 Introduction

With the accelerated process of informatization of the society, information security has become an increasingly important issue. Research shows that a considerable number of security problems are due to security vulnerabilities in software development. Especially, the vulnerability introduced during design phase of software has been a major source of security issue. Due to the lack of semantic and contextual information of application, security measures available on the network and operating system level cannot prevent the applications from attacking. Therefore, it highlights the need to identify threats in the software design phase as early as possible and design appropriate response options based on the security assessment to enhance the security posture of applications. China Classified Protection of Information System(CPIS) policy, the fundamental policy of China cybersecurity strategy, emphasizes the importance of security requirement analysis for software development. However, no analytical method is given by it. In this paper, an CPIS-compliance security requirements analysis method based on threat modeling is proposed, which assesses cyber attack by threat-attack Graph(*TAG*) and develops response options and their priorities according to ***Baselines for Classified Protection of Information System***(Hereinafter referred to as "CPIS Baselines") to improve the security of software during design phase.

2 CPIS-compliance Security Requirements Analysis Method for Software Development

2.1 Necessity of Threat Modeling

CPIS policy is the fundamental policy of China cybersecurity strategy, which aims to protect the confidentiality, integrity and availability of information and information systems processing, storing and transferring the information. The CPIS Baselines point out security requirements for different levels of information system, requiring information systems have the appropriate basic protection capability. For example, the protection capability required for security-level 3 Information system in the CPIS Baselines states as "the ability to protect the system from organized groups from outside (such as a business intelligence organizations or criminal organizations, etc.), malicious attackers with resources(including staff capacity, computing power, etc.), severe natural disaster (catastrophe with great intensity, long duration, broad coverage, etc.), or equivalent threat sources(malicious insider, unintentional mistakes, serious technical failures, etc.). The ability to discover vulnerabilities and security incidents. The ability to quickly restore most functions after the system is compromised"[1]. These requirements actually reflect the ability of information systems against

threats and the resilience of information system after being destroyed. Information system with higher security level should be able to confront threats with greater intensity and withstand for longer time. Even for information systems with the same security level, due to different business mission and threat environment, different protection strategies are required. Thus, the fundamental premise of CPIS is to identify and analyze threats and further model threats. Because software plays the core role of information systems, it is safe to determine the security level of information system by security requirements of the software. Therefore it is significantly important to identify threats using threat modeling and design appropriate response options to enhance security posture of information system.

2.2 Process for CPIS-compliance Security Requirements Analysis

According to "***Implementation Guide for Classified Protection of Information System***" (hereinafter referred to as the "Implementation Guide"), the lifecycle of Classified Protection of Information System includes security classification, security planning, security design and implementation, security operation and maintenance, and information system disposal. The goal of security planning is to define the security requirements of information system and design CPIS-compliance security plan to guide the subsequent construction of information system based on the factors such as security level and business mission of the target information system. Implementation Guide also noted that security requirement analysis should first produce basic security requirements based on CPIS regulations and standards; then, risk analysis should be used to identify possible risks and propose special security requirements of the information system[2].

Definition 1. (Threat-Attack Graph, *TAG*) Threat-Attack Graph, a directed graph composed of root node, leaf nodes and directed links, is used to represent, analyze and assess threats and attacks the applications are facing, referred to as *TAG*, $TAG = (T, A, P, Cr)$, in which:

1) *T* is the set of threats, represented in *TAG* as root node. $T = \bigcup_{ci \in C} Tr_{ci}$, in which *C* is the set of system components including processing components (core process, dependent services, etc.), data components (persistent data and non-persistent data), and transferring components[3]. Tr_{ci} is the particular set of threats for a given component *ci*;
2) *A* is the set of attacks to achieve goals of threat sources, represented in *TAG* as leaf nodes;
3) *P* is probability of a particular attack, $P \in [0,1]$. *P* is represented in *TAG* as directed links;
4) *Cr* is the impact caused by attack, which represents the scope and severity of damage after successful implementation of the attack.

TAG represents all known attacks corresponding with threats. Assessment of threats based on implementation of attack is the foundation to determine the response options and their priorities. For a given implementation of attack *a*($a \in A$), the priority of response option should be proportional to the probability of a successful attack, and also proportional to the impact of the attack, which is:

$$res = p_a cr_a \quad (p_a \in P, \ cr_a \in Cr) \tag{1}$$

From a certain aspect, priority of response option reflects the implications of the attacks on the security posture of application. The larger the value of priority, the more urgent to implement the response option.

The following is the details of the process for CPIS-compliance security requirements analysis in the form of pseudo-code, according to Implementation Guide.

```
Input: set of system components, set of threat category
Output: response options in descending order according to priority
  Begin
     componentSet = getComponents();            //    Get system components
     threatSet = getThreatSet ();               //    Get threat category
     foreach (threat in threatSet) {
          foreach (component in componentSet){
               if (threat exists for component){
                    TAG(threat, attack, probability, criticality);        // Generate TAG
                    sortRespondsInDesOrder();   // Generate response options in descending order
               }
          }
     }
  End
```

3 Case Study

To further clarify CPIS-compliance security requirements analysis and verify its effectiveness and practicality, the following is study case in which CPIS-compliance security requirements analysis is applied to Institute X's Content Management System(CMS).

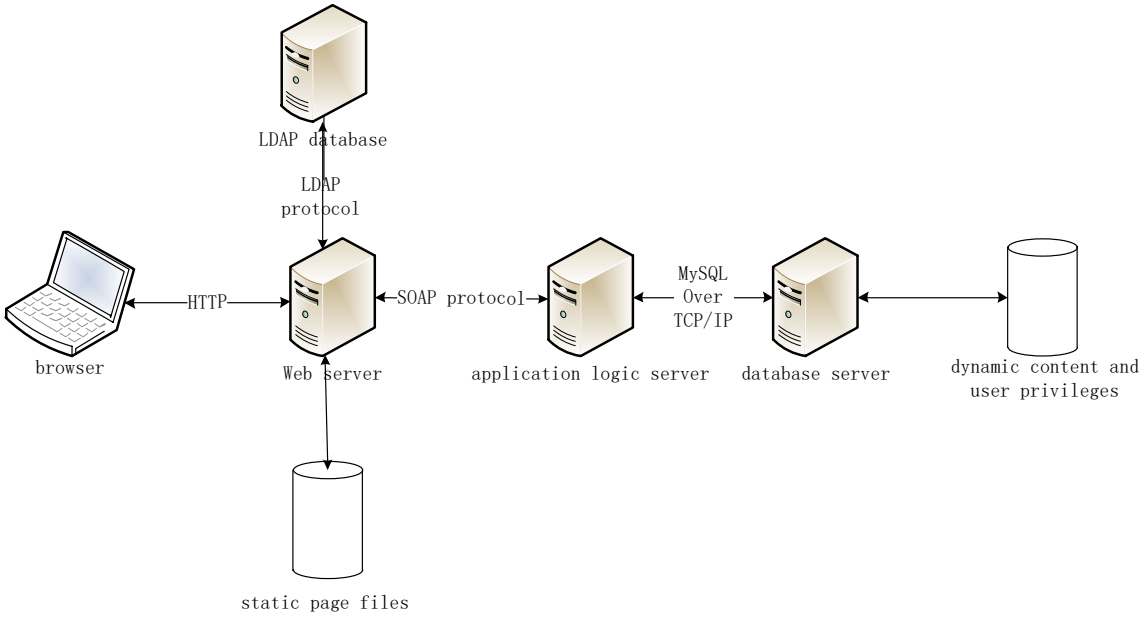

Figure 1. Overview of Institute X's Content Management System(CMS)

3.1 Identify System Components

Figure 1 depicts an overview of Institute X's Content Management System. The functions of CMS include user authentication, information dissemination, document flow (including entry, approval, etc.) and privilege management. According to CPIS-compliance security requirements analysis, System Components are as follows:

1) Processing components: including user's browser, Web server, application logic server, database server;
2) Data components: including static page files, LDAP database, dynamic content files and user privileges;
3) Transferring components: including HTTP protocol, LADP protocol, SOAP protocol, MySQL connection protocol.

3.2 Categorize Threat

Given behavior-based threat category enumerated by national standard, ***Information Security***

Technology-Risk Assessment Specification for Information Security, Table 1 shows threats applicable to software[4].

Table 1 Behavior-based Threat Category

Category	Description	Sub-category
Privilege Escalation or Abuse	Act of exploiting a bug, design flaw or configuration oversight in an operating system or software application to gain elevated access to resources that are normally protected from an application or user; Damaging the information system by improper use of authorized permission.	Unauthorized access to network resources, unauthorized access to system resources; abuse privileges by unauthorized modifying the system configuration or data, disclosing confidential information, etc.
Network Attacks	Attack or intrude information system by means of tools or techniques.	Network probing and information gathering, vulnerability detection, sniffer (accounts, passwords, permissions, etc.), user identity forgery and fraud, theft and destruction of the user or business data, system unauthorized control and destruction, etc.
Information Leakage	Leak information to unauthorized users.	Internal information leakage, external information leakage, etc.
Tamper	Illegally modified information, undermine security of information system by corrupting integrity of information or making system unavailable.	Tampering with network configuration, tampering with system configuration, Tampering with security configuration, tampering with the user identity information or business information, etc.
Repudiation	Deny the operation or transaction that has already been done.	object repudiation, subject repudiation, third-party repudiation, etc.

3.3 Generate TAG

Apply the threat items in Table 1 to system components and generate *TAG* whenever component is affected. By analyzing this case, it is found that the CMS is facing multiple threats as follows:

1) Tampering:
   a) Users tamper with data components, such as: users unauthorized modification of the static page file;
   b) Information has been tampered with during transmission over transferring components, such as: authentication information transmitted from the browser to the Web server is illegally modified.
2) Privilege Escalation or Abuse:

a) Privilege user's authentication and response information is unauthorizedly tapped by non-privilege user during transmission between the transferring components(Web server and LDAP server);

3) User Identity Forgery and Fraud:
   a) Counterfeit identity after user data component(LDAP database) is compromised;
   b) Forged identity after authentication information is tapped during transmission between the transferring components(browser and Web server);
4) ……

Figure 2 exemplifies *TAG* generated from a particular threat, "User Identity Forgery and Fraud", in which the implementation of attack and the value of probability and impact are obtained by Delphi method and statistical information from cybersecurity incidents. In this case, the probability of a successful attack is divided into 10 levels in the range from 0.1 to 1.0 while impact ranging from 1 to 10 for 10 levels.

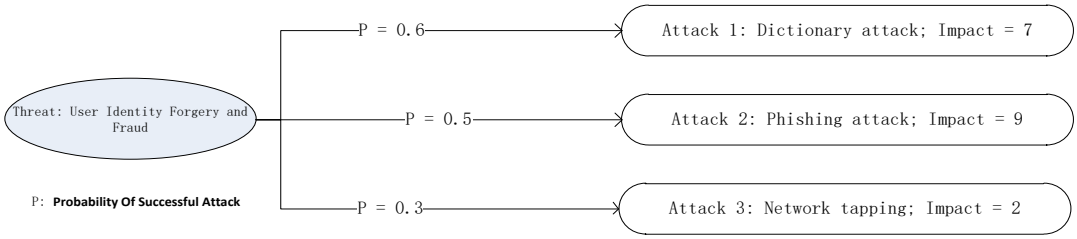

Figure 2 TAG generated from Threat "User Identity Forgery and Fraud"

3.4 Develop Response Plan

Based on CPIS Baselines, response plan is developed to counter attack and options picked up from CPIS Baselines are prioritized by product of probability and impact of the attack, as shown in Table 2.

Table 2 Response Plan in Descending Order of Priority

Attack	Response Plan	Priority
Phishing Attack	1) Deploy multi-factor authentication to control access;	4.5
Dictionary Attack	1) Provide the login failure processing function by session termination, limiting login attempts or automatic application withdrawal, etc. 2) Provide user with unique identification and check authentication information to ensure that the application does not duplicate user identification, and authentication information cannot be fraudulent.	4.2
Network Tapping	1) Verify communicating parties by use of cryptography during connection initialization. 2) Encrypt or take other effective measures to preserve the confidentiality of system management data, identification information and other business data during transmission; 3) Encrypt entire packets or session of communication.	0.6

Priority of the response option is proportional to the implications of the attack on the system. From Table 2, although the probability of phishing attack is not the biggest, it has highest priority. Thus, it means greatest implications on the system and should be on the top of list.

4 Summary

China CPIS policy is the fundamental policy of China cybersecurity strategy. Therefore, CPIS-compliance software solution is highly demanded to help organizations comply with CPIS associated policies and standards. In this paper, a CPIS-compliance security requirements analysis method for software development is proposed, which utilizes *TAG* to assess the attack and then develop response plan according to priority. Using *TAG*, controls from CPIS Baselines are introduced into software during design phase, which efficiently enhances and improves software security. A case is studied to illustrate the practicality and effectiveness of *TAG* method.

References

[1] GB/T 22239-2008 Information security technology-Baseline for classified protection of information system[S].

[2] GB/T 25058-2010 Information Security Technology-Implementation guide for classified protection of information system[S].

[3] Michael Howard, David LeBlance. Writing Secure Code[M]. Microsoft Press.

[4] GB/T 20984-2007 Information security technology-Risk assessment specification for information security[S].

[5] He Ke, Li Xiaohong, Feng Zhiyong, Object-Oriented Threat Modeling [J], Computer Engeering. 2011 Vol.37 No.4: 21-23