Threat and Countermeasure Patterns for Cloud Computing

Takao Okubo
Graduate School of Information Security
Institute of Information Security
Yokohama, Japan
okubo@iisec.ac.jp

Yoshiro Wataguchi
Secure Computing Laboratory
Fujitsu Laboratoriues limited
Kawasaki, Japan
wataguchi@jp.fujitsu.com

Nobuyuki Kanaya
Secure Computing Laboratory
Fujitsu Laboratoriues limited
Kawasaki, Japan
kanaya.nobuyuki@jp.fujitsu.com

***Abstract*—Recently cloud computing markets have expanded, and there are various kind of services and their providers. However, Security is the primary concern of cloud users. However, service providers are unaware precisely of the types of security countermeasures required for their cloud servers. A method to define what the type security required for each operator using two security patterns are proposed. One is for typical threats and the other is for typical countermeasures. Using the two patterns with the relationship information among the functions, cloud components and stakeholders, the security coverage of the cloud and the security duties of each operator become clear.**

***Index Terms*—security; operation; cloud; service level agreement; security pattern;**

I. Introduction

Recently cloud computing markets have expanded, because cloud computing provides a convenient computing and maintenance environment for application providers. However, the chief concern for emerging users of cloud systems is security. The introduction of cloud systems and services brings with it several security risks. An example of these risks is security incidents caused by a lack of security countermeasures. Cloud computing is offered by various providers in several layers such as software-as-a-service (SaaS), infrastructure-as-a-service (IaaS), and platform-as-a-service (PaaS). Such an environment makes it unclear to identify the one responsible for certain security functions, and to determine whether the current security countermeasures are sufficient for securing the whole system. Cloud providers are unaware precisely of the types of security countermeasures required for their servers because there are no established relationships among functions and components. There is also commonly a lack of dependency information for security countermeasures.

To solve these issues, we propose two security patterns for cloud computing. One is threat pattern and the other is countermeasure pattern. Using the two patterns with the relationship information among the functions, cloud components and stakeholders, the security coverage of the cloud and the security duties of each operator become clear. These patterns would help making service level agreements between different cloud providers.

Our security patterns have been used for establishing the cloud security criteria provided by Fujitsu. These criteria describe the essential security operating functions for cloud providers. We have also performed a threat analysis of the criteria with the proposed security patterns and have confirmed that security can be assured under certain assumptions.

II. Background and Related Works

We expect there would be some typical patterns for security-related cloud operations. It is necessary to take into consideration security functions with human operations as well as functions that are implemented systematically. However, human-related security operations must be isolated from system security in order to clarify the problem.

A. Related Works

Fernandez et al. proposed several patterns related cloud computing and security [1] [2].

Hashizume et al. proposed several typical misuse patterns that may threaten cloud computing systems [3] [4] [5]. Although these patterns are useful for identifying threats and defining countermeasures, they do not focus on the dependencies among threats or actual functions deployment to the cloud components.

Open security architecture also provides cloud security patterns [6]. These patterns are associated with stakeholders such as IT managers, cloud service providers, developers and end users. Such a categorization of patterns is similar to our approach. However, the contexts and problems that these patterns provide are abstract and stakeholders applying them must carefully understand and realize these patterns. However, the security coverage and sufficiency of such patterns remains unclear.

The National Institute of Standards and Technology (NIST) has published a draft document for cloud computingsecurity [7]. The document model the general cloud computing systems. and offers security requirements for every stakeholders of cloud computing such as providers, brokers, consumers and auditors. Although NIST's approach is similar to us, there is no precise reasoning that the requirements are needed.

Our work is to reason the validity of such model and requirements with giving the set of threats and structured countermeasure set. Supakkul et al. proposed security patterns that mitigate the threat of credit card theft [8]. The patterns consist of threats and mitigations. Our motivation is to propose patterns with threats and countermeasures similar to [8] for cloud providers.

III. Cloud Security Patterns

We consider it would be helpful if every operators of cloud systems can recognize typical potential threats and required countermeasures for which they are responsible. However, it is actually difficult for them to identify threats and countermeasures without sufficient security knowledge. Therefore we utilise two types of pattern. One is the potential threats behind cloud systems. And the other is countermeasures against these threats. In addition, the countermeasure pattern contains relationships among countermeasures. I would be helpful for each operators to know the countermeasures required for themselves.

A. Cloud Threat Pattern

Typical security functions and responsibility for cloud security can be defined as a cloud countermeasures pattern.

Name: Cloud threat pattern

Context: Multiple cloud service providers provide different services such as SaaS and IaaS for a single cloud system. Several servers are deployed as the elements of the cloud system, and stakeholders, including providers and end users, play roles for the servers (i.e. as users or owners). A system element provides functionalities for the components of the cloud.

Problem: Cloud providers are unaware precisely of the types of security countermeasures required for their server because there are no established relationships among functions and components. There is also commonly a lack of dependency information for security countermeasures.

Solution: The security countermeasures can be clarified if the typical threats against cloud systems can be identified and the owner and user information for the system element servers are described, as well as the relationships among servers and security countermeasures. This pattern provides the threat and countermeasure information. Combined with one of the 'cloud ownership patterns', cloud providers can find a solution to the problem.

Structure: The structure of this pattern is based on the threat tree [9] [10].

Figure 1 shows the structure of threats. It is described with threat tree. If the threat *Tc* is a child node of *Tp*, *Tc* represents the condition or method that achieves *Tp*. There is two types of relationship, 'and' or 'or'. This pattern contains only 'or' relationships. 'Or' means the parent threat is possible if one threat of the children is possible. 'Most of the threats appear at the threat classification by web application security consortium (WASC) [11]

B. Cloud Countermeasure Pattern

Typical security functions and responsibility for cloud security can be defined as a cloud countermeasures pattern.

Name: Cloud countermeasure pattern

Context: Multiple cloud service providers provide different services such as SaaS and IaaS for a single cloud system. Several servers are deployed as the elements of the cloud system, and stakeholders, including providers and end users, play roles for the servers (i.e. as users or owners). A system

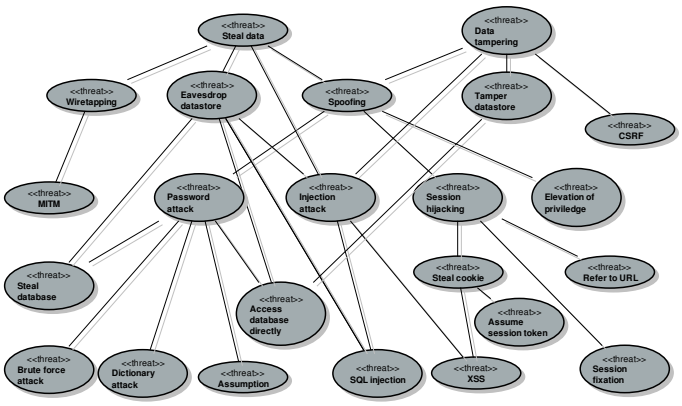

Fig. 1. Cloud threat pattern

element provides functionalities for the components of the cloud.

Problem: Cloud providers are unaware precisely of the types of security countermeasures required for their server because there are no established relationships among functions and components. There is also commonly a lack of dependency information for security countermeasures.

Solution: The security countermeasures can be clarified if the typical threats against cloud systems can be identified and the owner and user information for the system element servers are described, as well as the relationships among servers and security countermeasures. This pattern provides the countermeasure information. Combined with one of the 'cloud threat pattern', cloud providers can find a solution to the problem.

Structure: Figure 2 shows the structure of countermeasures. In Figure 2, classes with 'security' stereotype represents countermeasures, and classes with 'component' stereotype represents components. Each countermeasure class has property who is responsible for the countermeasure (presented by the 'responsibility' attribute of the countermeasure class). If the requirements of a countermeasure Ci depend on other countermeasures C1..Cm, Ci has links to C1..Cm. A countermeasure class also has the link to the cloud system component in which the countermeasure should be implemented.

For instance, to achieve user authentication, four additional security functions are required: role-based access control (RBAC), counter-brute force attacks, remote access protection and ruggedization of the server for disclosure. The 'protection of remote access' function is related to a 'remote access' component, which means that the 'remote access' component must have the 'protection of remote access' function. The 'protection of remote access' function also identifies the one responsible for the function. This function defines the 'owner' of the component that is responsible for the 'protection of remote access'. The dependency information of this pattern is based on the experiences of cloud security experts of Fujitsu Laboratories.

The dependency relationships is constructed based on the experiences of building and consulting numbers of cloud systems provided by the Fujitsu. Several security function

TABLE I
THREATS AND CORRESPONDING COUNTERMEASURES

Threat	countermeasure
DoS	ruggedization of server for disclosure
Eavesdrop data store	prevention of insider attack
	surveillance of usage
	log audit
	log acquisition
	force using unique account
Tamper data store	prevention of insider attack
	surveillance of usage
	log audit
	log acquisition
	force using unique account
Man in the middle attack	protection of remote access
CSRF	handling vulnerability(*)
Session hijack	handling vulnerability(*)
Refer session id from URL	handling vulnerability(*)
Password attack	user authentication
Data leakage from database	protection of internal servers
Session fixation attack	handling vulnerability(*)
Session token assumption	handling vulnerability(*)
Unauthorized access on DB	protection of internal servers
Brute force attack	counter brute force
Dictionary attack	multi-factor authentication
Password assumption	account lockout
XSS	handling vulnerability(*)
SQL injection	handling vulnerability(*)

elements are from security patterns. For example, 'account lockout' in Fig. 2 is from the 'Account lockout' pattern of the security patterns catalog [12].

The relationships among threats and countermeasures are shown in Table II. Using Table II, developers can verify whether the countermeasures are sufficient to mitigate all threats identified against the target cloud systems.

Known uses: Fujitsu cloud systems follow this pattern.

IV. HOW TO USE PATTERNS

Cloud providers are able to identify the threats and countermeasures using the threat pattern and the countermeasure pattern. However, each cloud provider would not able to know precisely what countermeasures they have to do. It would be possible if the patterns are used with the information of relationships among stakeholders and system components The following pattern is helpful for providing such information.

A. Cloud Stakeholder Pattern

A cloud stakeholder pattern can be defined based on the metamodel. The format is derived from the body knowledge of security patterns [13].

Name: Cloud stakeholder pattern (SaaS / IaaS)

Context: Multiple cloud service providers provide different services such as SaaS, IaaS for a single cloud system. Several servers are deployed as the elements of a cloud system, and stakeholders, including providers and end users, play roles for servers (i.e. as users or owners). A system element provides functionalities for the components of the cloud. Target of this pattern is providers providing SaaS and IaaS. PaaS providers are out of the scope of this pattern.

Problem: The responsibility of security is not clearly defined because there is no stakeholder relationship among stakeholders and system elements.

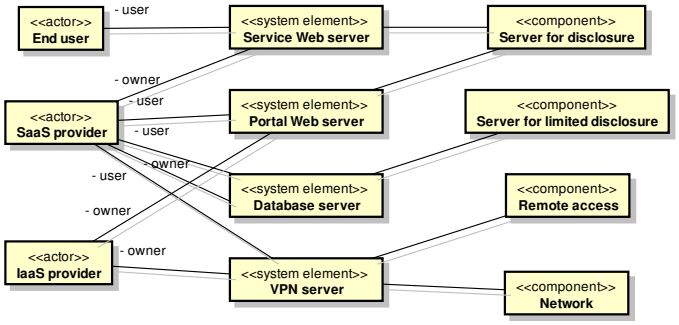

Fig. 3. Cloud stakeholder pattern

Solution: The responsibility would be clearified if the relationships among stakeholders and system components are defined, along with the relationships among components and countermeasures. Such a pattern provides the former information. Combined with one of the 'cloud countermeasure patterns', cloud providers can determine the solution to the problem.

Structure Figure 3 shows the structure of this pattern. There are four types of server for a cloud system: service web, portal web, database and VPN. Each server type has two types of actors, a user and an owner. For example, the user of a portal server is a SaaS provider and the owner of the portal server is an IaaS provider. Each server also plays one or more roles for components. For example, a VPN server plays both 'remote access' component and 'network' component roles.

Known uses: Fujitsu cloud systems follow this pattern.

With the cloud stakeholder pattern and the cloud threat and countermeasure pattern, we can identify the required security functions for each cloud provider and/or user. For example, security functions for IaaS providers can be identified using the following steps.

1) Required security functions related to components identified are listed with the cloud threats and countermeasures pattern.
2) If a component of (1) has the 'user' responsibility attribute and it is in CU, then the SaaS providers are responsible for the component.
3) If a component of (2) has the 'owner' responsibility attribute and it is in CO, then the IaaS providers are responsible for the component.

From the above steps, the security functions for which IaaS providers are responsible are as follows.

- Ruggedization of server for disclosure
- Protection of remote access
- Protection of internal server
- Server for limited disclosure
- User authentication
- Log acquisition
- RBAC
- Account lockout
- Multi-factor authentication
- Port scan

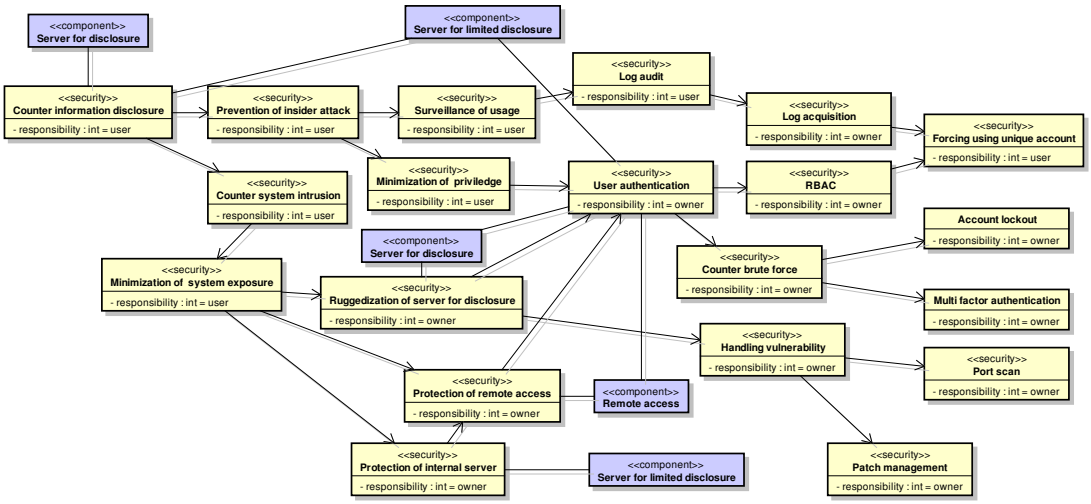

Fig. 2. Cloud countermeasure pattern

- Patch management

We established cloud security whitepaper provided by Fujitsu [14][1] using the proposed patterns. The criteria described, define the essential security operating functions for cloud providers.

B. Limitation

In Table I, CSRF, Refer session id from URL, Session fixation attack, Session token assumption, XSS and SQL injection are related to the countermeasure 'handling vulnerability'. However, 'handling vulnerability' does not state any concrete implementation for mitigating the attacks. The countermeasure pattern does not provide secure coding guidance or pattern. Thus users of the cloud countermeasure pattern must find countermeasure methods. It is the limitation of the proposed method.

V. Conclusion

We have developed two security patterns, the cloud threat pattern and cloud countermeasure pattern for typical cloud systems. Various cloud providers such as SaaS providers and IaaS providers are able to identify required countermeasures for which they are responsible combining the proposed patterns with relationship information among the countermeasures, cloud components and stakeholders. We have succeeded to establish the security criteria of the Fujitsu cloud using the proposed patterns.

In the future, we intend to examine the proposed security patterns using other cloud systems. Although we assume that there would be no software vulnerabilities on the cloud systems, this assumption is not assured. We should consider the case in which the system contains vulnerabilities.

Another future work is to combine our work with the NIST 500-299 document [7]. We are going to verify the NIST model and requirements with our model and patterns.

[1]http://jp.fujitsu.com/solutions/cloud/concept/security-whitepaper/ (in Japanese)

Acknowledgment

The authors would like to thank to Dr. Eduardo Fernandez and Dr. Sam Spakkul who gave a lot of suggestions for improving our research.

References

[1] E. B. Fernandez, N. Yoshioka, and H. Washizaki, "Patterns for cloud firewalls," *Procs. of AsianPLoP (Pattern Languages of Programs) 2014*, 2014.

[2] E. Fernandez, R. Monge, and K. Hashizume, "Two patterns for cloud computing: Secure virtual machine image repository and cloud policy management point," *Procs. of 20th Conf. on Pattern Languages of Programs (PLoP 2013)*, 2013.

[3] K. Hashizume, E. B. Fernandez, and M. M. Larrondo-Petrie, "A pattern for software-as-a-service in clouds," *ASE/IEEE International Conference on BioMedical Computing (BioMedCom)*, vol. 0, pp. 140–144, 2012.

[4] D. G. Rosado, D. Mellado, E. Fernandez-medina, and M. Piattini, *Security Engineering for Cloud Computing: Approaches and Tools*. Information Science Reference, 2012.

[5] K. Hashizume, E. B. Fernandez, and nd Nobukazu Yoshioka, "Misuse patterns for cloud computing," *wenty-Third International Conference on Software Engineering and Knowledge Engineering (SEKE 2011)*, 2011.

[6] OSA. (2014) Sp-011: Cloud computing pattern. [Online]. Available: http://www.opensecurityarchitecture.org/cms/library/patternlandscape/251-pattern-cloud-computing

[7] NIST, *Cloud Computing Security Reference Architecture*. NIST Special Publication 500-299, 2013.

[8] S. Supakkul, T. Hill, L. Chung, and E. A. Oladimeji, "Goal-oriented security threat mitigation patterns: A case of credit card theft mitigation," in *PLoP09*, 2009.

[9] F. Swiderski and W. Snyder, *Threat Modeling*. Microsoft Press, 2004.

[10] M. Howard and S. Lipner, *The Security Development Lifecycle*. Microsoft, 2006.

[11] WASC. (2014) Threat classification. [Online]. Available: http://projects.webappsec.org/w/page/13246978/Threat%20Classification

[12] D. M. Kienzle, M. C. Elder, D. Tyree, and J. Edwards-Hewitt. (2002) Security patterns repository version 1.0. [Online]. Available: http://www.scrypt.net/~celer/securitypatterns/

[13] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns: Integrating Security and Systems Engineering*. Wiley, 2006.

[14] M. Okuhara, T. Shiozaki, and T. Suzuki, "Security architectures for cloud computing," *Fujitsu Science Technology Journal*, vol. 46, no. 4, pp. 397–402, 2010.