A Web-based System for Prevention of Information Leakage

Yasuhiro Kirihata
Hitachi Software Engineering Co.,Ltd.
6-81, Onoe-cho, Naka-ku
Yokohama, 231-0015 Japan
kiri@ori.hitachi-sk.co.jp

Yoshiki Sameshima
Hitachi Software Engineering Co.,Ltd.
6-81, Onoe-cho, Naka-ku
Yokohama, 231-0015 Japan
same@ori.hitachi-sk.co.jp

ABSTRACT

In this paper, we propose a web-based system for prevention of the confidential information leakage caused by the person who is authorized to access. This system realizes the centralized access control to the distributed confidential information and supports the confidential pages generated dynamically by web applications. We show the design and implementation of this system that is transparent to users.

Keywords

Web, Security, Information protection

1. INTRODUCTION

As the Internet technology has become essential to office works, the office staff might deal with the confidential information such as customer information, personnel information, and designs of new products over the organization network. However, many incidents of confidential information leakage occur in organizations and most of these incidents are caused by the internal staff. Information leakage is one of the most serious security threats to the information security, but there are few effective methods to prevent it comparing with other conventional security threats such as eavesdropping of the network traffic.

In this paper, we propose the Data Leakage Prevention (DLP) system. This system provides protection of the confidential information stored in the web server against the information leakage such as bringing out the data by saving it as file, writing it to the media, and printing it out. Users can only read but cannot copy nor print the confidential information. The DLP system comprises four major components; Viewer, Encryption Proxy, Authentication Server, and Access Control Directory. Encryption Proxy, which is a proxy server interposed between client and web server, encrypts transmitted data of the confidential information on demand. Adopting this encryption method, it is not necessary to change the existing web server that stores the confidential information, and the DLP system supports the confidential information generated dynamically by web applications such as CGI or Java Servlet. In addition, whenever Viewer accesses the confidential information, Authentication Server authenticates users and controls the access. The system administrator can manage the confidential information with the configuration of Access Control Directory and centralized access control of the distributed confidential information can be realized.

2. ARCHITECTURE

We precise the architecture of the DLP system (Figure 1). Web server, Encryption Proxy, Access Control Directory and Authentication Server are located in the separated network segment so that no one can access them directly. Furthermore, those components authenticate each other and the communications among them are secure. No one can access those components without being authorized.

The web server stores confidential pages in plain form. Encryption Proxy is placed between the client and the web server to prevent the client from accessing to the web server directly via the network. It authenticates access users and encrypts the requested confidential page with the corresponding content key that is a secret key for encryption and stored in Access Control Directory. Viewer is installed in the client, which is an application to display confidential pages in this system. While Viewer is running, the user can neither take screen capture nor copy the displayed confidential information. Viewer has a function to save the encrypted confidential page.

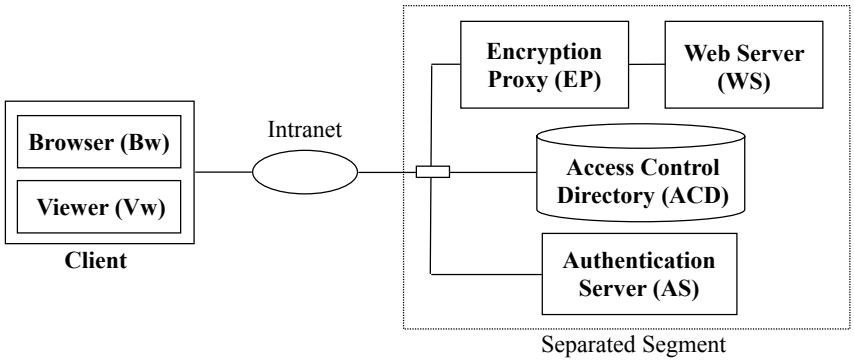

Figure 1. System architecture of the DLP system.

Access Control Directory is the database in which administrator registers the following tables; user table and secret data table. The user table records user accounts and passwords. The secret data table records data descriptors, confidential page URLs, content keys to encipher and decipher the confidential pages, and access control lists. The user table is used in user authentication process. To check the access permission to each confidential page, the system uses the secret data table. When the user opens a confidential page with Viewer, it requests the content key of the confidential page to Authentication Server. Authentication Server queries access control list and checks permission of the corresponding confidential page to Access Control Directory with ID, password and data descriptor, which identifies the confidential page.

In order to control the access to the distributed confidential information, we design the dynamics of the DLP system that is composed of two phases; download phase and open data phase. The download phase occurs when the web browser sends a request for a secret page stored in the web server to Encryption Proxy (Figure 2).

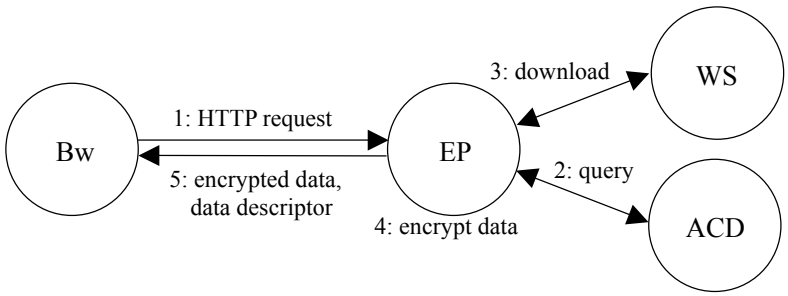

Figure 2. Download phase of the system dynamics.

In the first step, the web browser sends HTTP request for a confidential page to Encryption Proxy. Encryption Proxy analyzes the HTTP request and queries Access Control Directory if the request is for the confidential page or not. If the request is for confidential page, Encryption Proxy downloads it from the web server and encrypts it with the content key stored in the Access Control Directory. After that, Encryption Proxy adds data descriptor to the encrypted data and sends it to the web browser. In this phase, user cannot see the confidential pages because they are encrypted and stored in the local client. Hence, Viewer needs to obtain the content key to decrypt and display the downloaded confidential page.

Open data phase is occurred when Viewer opens confidential pages encrypted and stored in the local client (Figure 3). At first, Viewer sends ID, password, and data descriptor to Authentication Sever. After receiving those data, Authentication Server queries Access Control Directory in terms of ID, password, and data descriptor whether the user may access the confidential page or not. If user is allowed to see the data, Authentication Server sends content key to Viewer. Viewer decrypts confidential page data with the received key and displays it.

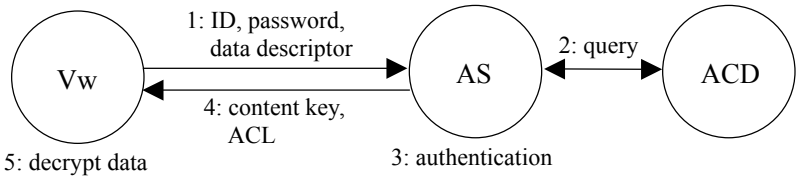

Figure 3. Open data phase of the system dynamics.

The most important process of the system dynamics is the two-way authentication between Viewer and Authentication Server. Actually, the result of user authentication is transmitted to Viewer precisely on the basis of the two-way authentication of those two components. Therefore, it is necessary for Viewer and Authentication Server to authenticate each other before the beginning of the session by authentication process such as the SSL mutual authentication. Applying this protocol, confidential page publisher can control the access to the distributed secret web pages, because users have to be authenticated and given access permission by Authentication Server whenever they open the confidential page with Viewer.

3. DESIGN AND IMPLEMENTATION

We built the prototype of the DLP system. The design concept of the prototype is transparency to users (no change in URLs and browser behavior). Viewer is implemented as ActiveX control so that Viewer is embedded into Internet Explorer and the user is unaware of its running. If a component of frame page is a confidential page, Viewer control displays only a part of frame page as a confidential page on Internet Explorer. It restricts copy and print functions to prevent users from bring out the displayed confidential information.

Viewer control prohibits screen capture while it is running. To realize the prohibition of screen capture, it hooks and cancels the Win32 API methods by which capture tools take hardcopies. We can use the techniques to inject an arbitrary DLL into another process [3] to hook the Win32 API methods. Using DLL injection technique, it is possible to manipulate the import section defined in each application and hook the Win32 API used in it. However, if the attacker could overwrite the import section of a capture tool after the injection of hooking DLL, it is possible to avoid the hooking routine and cancel the Win32 API hook. This is the imperative problem.

The prototype of Encryption Proxy is the modified Squid proxy, to which we added the authentication function as plug-in of the Squid. On the arrival of the HTTP request issued by Internet Explorer, the Encryption Proxy analyzes the Proxy-Authentication attribute in the HTTP header field to authenticate the access user and returns a web page for Internet Explorer to activate Viewer control. Confidential pages' URLs and user accounts are previously registered in the Access Control Directory and the Encryption Proxy queries it with LDAP.

4. RELATED WORKS

IBM research proposes a system for web content protection [1][2]. This system provides digital rights management to off-the-shelf Web browsers and browser plug-ins. It verifies the browser code with the digital signature scheme and prevents users from performing actions that are not allowed such as print, save as, and so on. It distinguishes the protected contents with the specific protocol, and Internet Explorer invokes the Trusted Control Handler with these method names. Therefore, it is necessary to change the existing web pages to introduce this system. In addition, usage rights information is stored in the client and there is no mechanism to change the usage rights dynamically.

5. CONCLUSION

We proposed the DLP system that realizes the protection of confidential information. By the application of the super-distribution system architecture [4] to the DLP system, it realizes the access control of the confidential information after its distribution. Furthermore, the on-demand encryption in the proxy realizes the support of confidential pages generated dynamically by web applications. We think that the DLP system is the essential architecture to deal with confidential pages and brings large effects to securities of the web system on the business.

6. ACKNOWLEDGEMENTS

We would like to thank Tsutomu Matsumoto, Professor of Yokohama National University, for helpful comments and discussions on design and evaluation of the DLP system.

7. REFERENCES

[1] M. Mourad, J.Munson, T. Nadeem, G. Pacifici, M. Pistoia, A. Youssef. WebGuard: A System for Web Content Protection. In Post Proc. of the Tenth International World Wide Web Conference, May 2001.

[2] A. Youssef. WebGuard: Making Copyright Protection Easy for Web Publishing. **http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/1829**.

[3] Jeffrey Richter. Programming Applications for Windows Forth Edition. Microsoft Press.

[4] Superdistribution: The Concept and the Architecture. **http://www.virtualschool.edu/mon/ElectronicProperty/MoriSuperdist.html**.