Working Paper No. 9
Topic (iv) Authentication, security and privacy issues

UNITED NATIONS
STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE

EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE
EUROPEN COMMUNITIES
(EUROSTAT)

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Electronic Raw Data Reporting
(Geneva, 6-8 November 2006)

Security Considerations in Web Environment

Toni Räikkönen, Statistics Finland

Security Considerations in Web Environment

Toni Räikkönen
Dept. of Information Technology and Statistical Methods
Statistics Finland, Helsinki, Finland
toni.raikkonen@stat.fi

Joint UNECE / Eurostat Work Session on Electronic Raw Data Reporting
Geneva, Switzerland, 6-8 November, 2006

Abstract

It is quite common nowadays, that organizations are keen to offer their services online. While this is truly an admirable effort, it can also expose terrific security issues. There are five main principles in security, be it an online service or not. These are confidentiality, integrity, availability, authentication and accountability. Each of these must be elaborately processed on the organization level before the outside access to its assets can be made possible.

Categories and Subject Descriptors

ACM Computing Classification System (1998):

K.4.4 [**Computers and Society**]: Electronic Commerce – *Security*; K.6.5 [**Computing Milieu**]: Security and Protection – *Authentication*; D.4.6 [**Operating Systems**]: Security and Protection – *Access Controls*, *Cryptographic Controls*.

Keywords

Security, Authentication, Authorization, Security Threats, Security Strategy

1 Introduction

Security is all about protecting *assets* which in a web environment is typically the data related to the usage of a web application system; the data, the user gives to

the organization via the system. The well-known security aspects that also apply here are [8]:

- *Confidentiality*; unauthorized access to the data is prohibited.
- *Integrity*; unauthorized modification of the data is prohibited.
- *Availability*; the data is made available to the authorized users.
- *Authentication*; the user is appropriately identified.
- *Accountability*; the logging of all user activity.

The most common threats, targeted towards applications of web environment, are briefly covered in chapter two. Chapter three discusses about the most common ways of protecting an online application and gives some examples of their weaknesses. In chapter four, password protection scheme is discussed somewhat. A perfect security solution will probably not be found or at least, it will not be usable to the end users. Instead, a good security strategy should be adopted. The basics of security strategy will be covered in chapter five. In chapter six, the security approaches of Statistics Finland's online web data collection application are discussed.

2 Common threats

World Wide Web is full threats targeted against individual users and organizations. Even, during a short www-surfing session, the user might be exposed to several threats and infected by a number of malware. Even greater risk is, of course, targeted against organizational assets. In this chapter, some common threats present in the online web application environment, are briefly listed.

2.1 Command injection attacks

Command injection attacks are typically based on injecting malicious string values for underlying system. These string values are many times used, for example, for database queries. In fact, the most common command injection attack is an *SQL*

command injection attack [5, 12]. The application is vulnerable to the SQL injection if the attacker has a possibility to insert partial SQL statement into an existing SQL statement of the system. Typically, for example, the attacker injects malicious SQL statements as credentials in the login phase. If the system is not able to recognize this, it can, in the worst case, allow the attacker to login into the system or return highly confidential information like passwords, for example.

2.2 Cross site scripting

Cross site scripting (XSS) consists of a range of attacks where the attacker injects malicious JavaScript into the web application [4, 5, 6]. With cross site scripting, the attacker can steal or modify the information the user is handling with the web site. For example, the user could pass the credit card information for an online shopping cart and the XSS injection could forward this information to the attacker. In another approach, the user can be even redirected to a whole different malicious web site. In both cases this is totally transparent to the user, who does not have a clue that something ill-disposed is happening.

Cross site scripting attacks are typically very easy to implement with some bare knowledge of the target site. On the other hand, they are extremely difficult to notice and prevent, not to mention that they can cause significant damage to the organization and its credibility.

2.3 Eavesdropping

Eavesdropping is a method where someone, a hidden party, is illegally listening and intercepting the communication between the other two parties [1, 16]. Eavesdropping can sometimes also mean a *man-in-the-middle* type approach. In this scenario, there is an individual who is taking the requests from the client and forwarding them to the server. In this way, the eavesdropper can access, for example, the credit card numbers or any other confidential information the client is sending, and use it for his own purposes.

2.4 Social engineering

Social engineering is a great threat that is not commonly recognized as a real threat at all. Still, according, to Gartner, more than 70 percent of attacks targeted towards IT systems come from inside the organizations [15]. It can be someone who is providing confidential information about the system's behavior to the attackers or someone who is trying to raise his own reputation in the information highway community of wrong doers. Noteworthy is also that the "insider" may not be the organization's own personnel. It can be someone visiting the organization, a contractor, even the security guy at the reception.

3 Compromising security

There are several ways a web application could be protected. Unfortunately, these all contain some weaknesses, which makes them quite easy to compromise. This being a fact, no one method should be solely used for protecting a web site and its assets.

3.1 Authentication

Passwords are probably the most vulnerable part of the user authentication. Still, regular password protected accounts remain the most widely used mechanism of user authentication.

Password protected systems can be compromised in many ways, the most common probably being *brute force dictionary attacks*. This means that the account is being continuously hammered with a list of words, dictionaries, used as passwords. This can be done online and the server typically informs the login failure in some recognizable way that can be used as a pattern for the brute force algorithm.

A system can be prepared for dictionary attacks in two common ways [9]. *Delayed response* means that the server will use some extra time before it returns the information whether the authentication was successful or not. This small pause is usually enough to make the hammering obsolete for its purpose. Another usable

method is *account locking*. This means that the account is locked after a certain number of repeated login failures.

Although both methods could, in general, be seen as good methods against dictionary attacks, there are some weaknesses arising also. Delayed response is useless if the attacker is about to break any account. It does not matter if the response is delayed if several accounts are hammered in parallel. Account locking, on the other hand, can cause even more unpleasant effects. The system can be confronted with the *denial of service* (DoS) attack very easily if the hacker's only motivation is to lock out as many accounts as possible. In addition, the users themselves can lock their accounts by mistake when typing the passwords incorrectly several times.

3.2 SSL

The most typical way of securing communication between the client and server is using *Secure Sockets Layer* (SSL). SSL provides authentication, encryption and integrity to TCP/IP traffic. SSL is based on PKI and handles the authentication by exchanging digital certificates signed by trusted certificate authorities. The most typical way of using SSL is using HTTP over SSL (HTTPS).

Although SSL is considered to be secure enough for confidential online communication, it can be compromised quite easily [11]. As SSL is based on public key cryptography, it needs to exchange the keys between the browser and the server to establish the secured communication line. In such situation, it is enough if the man-in-the-middle attacker is able to acquire the server's digital certificate. The attacker can then intercept the communication and decrypt the data very easily.

Another risk arises from the fact that browsers' allow establishing an SSL secured connection even if the certificate is not signed by a well-known trusted certificate authority or the certificate is not valid at all, for example, has been expired. Now, this being the case, anyone can setup a malicious secure server and pretend to be the organization the client tries to access.

4 Password protecting assets

Like said, passwords are the most used mechanism of user authentication today. They pose a lot of risks but, still, they are the de facto standard for the time being. So what makes regular passwords bad and how could they be improved?

According to the study of Gaw and Felten people tend to use three or less passwords for all their needs [3]. To make this sufficient enough for all accounts, they reuse passwords, typically at least twice. People will reuse passwords; that's a fact. A bigger problem comes when they use the same password for several active accounts. It is enough for the attacker to get the password from one, perhaps from a poorly protected, site and use it to compromise multiple accounts elsewhere. Gaw and Felten mention that a regular user does not actually see the security as a priority aspect. In fact, all kinds of inconvenient security solutions, like passwords, are seen as a nuisance rather than a protection.

If no password policy is forced, people tend to use very simple passwords. Why? Simply, because the complexity of the password increases the amount of memory burden required to remember it. The more difficult the password is to remember, the more likely it is written down on a paper. So, in this sense, complexity is not the solution and actually violates the most recognized usability standard for computer systems [2]. On the other hand, according to the several empirical studies, there is clear need for stronger passwords, to prevent algorithmic attacks [14]. The studies indicate that a password must not be selected from a dictionary of any language, and the minimum length should be eight characters consisting of letters, numbers and special characters. To ease up the memory burden caused by this kind of password policy, *a pass-phrase* approach could be used. A pass-phrase is a natural language phrase that can be easily derived from a mnemonic. For example, a phrase "*My mother has three dogs and two cats*" could be something like "*mmh3d&2c*". In this case, only the first letter of each word is used and numerical meanings are presented as numbers and &-sign indicates the and-word.

5 Security strategy

A perfect technical security solution is just a nice dream, something that very probably can not be achieved, and actually should not even be sought after. No matter what a terrific invention one person makes, there is always someone to compromise it. This has been seen many times during the years with different copy protection schemes and encryption algorithms.

So, instead of wasting the resources on something that can not be achieved, a more suitable approach should be taken:

- Accept the fact that a security solution must be frozen on a certain level. There is no point going beyond that level.
- Create a *security strategy*.

A security strategy is stemmed from *analysis*, *planning*, *implementation* and *monitoring* [7].

- In the analysis phase, the risks are evaluated by determining assets, threats and of course vulnerabilities. The determination of these is derived from the objectives and requirements. Once the assets and threats are enumerated, the next step is to quantify the value of each risk.
- In the planning phase, the security policies are defined. These policies indicate which threats are tolerable and which are not. A strategy for the policies is also defined. This strategy defines when the policy is to be enforced and for what particular reason. In addition, the policies are also exposed against auditing and reviewing practices.
- In the implementation phase, mechanisms and tools are to be selected for monitoring the threats targeted towards the assets. These mechanisms and tools are chosen based on the policies defined in the previous phase.
- In the continuous monitoring phase, the selected approaches are evaluated against the status quo. The evaluation indicates which methods and

technologies are successful and which are not, what new threats are exposed and what new techniques are invented. The monitoring phase is the single most important phase in the security strategy.

6 Security approaches in XCola

XCola (abbreviated from **X**ml based **col**lection **a**pplication) is an online web data collection application environment developed in Statistics Finland. It provides a feature rich engine for online surveys targeted towards business statistics' surveys. The approach of the engine is very generic; the questionnaires and survey logic are described in an XML format, thus making it possible to use the exact same application for all business statistics' surveys.

Security is an important issue in such an application and Statistics Finland is taking very seriously any potential threats that could arise. The communication route between Statistics Finland and the respondents is protected with today's de facto standards and the respondents are informed about the security measures Statistics Finland is applying in the data collection.

6.1 Authentication and authorization

XCola utilizes a *role-based access control* (RBAC) [13, 10] security framework for both authentication and authorization purposes. In our approach, it is not purely enough that the respondent of the survey is identified. He must also have an authorization to access the services and the information provided by the environment. Basically, this means that each and every user must have a role in the system. This is also a very useful approach in our case, since some of the survey logic can be coupled with the role thinking also.

The authentication mechanisms are based on directory services. This quite well prevents regular command injection attacks that could be targeted against the login functionality, although does not of course totally eliminate them. For dictionary based brute force attacks, the account locking method was considered but we found that not suitable for our purposes. We have some research data which indi-

cates that respondents are quite error prone when typing their credentials. So, the risk of incorrectly locked account would be too high. Instead, delayed response is used. With active monitoring, the hammering attempts can be detected anyway.

6.2 Credential policies

The empirical studies quite clearly indicate that the credentials generated with randomizing algorithms will probably be written down on a paper by the user [14, 2]. Despite this fact, we have chosen to use strong credential information with the minimum lengths corresponding with the recommendations of the studies and consisting of randomly generated alphanumeric characters. In addition, the user is not allowed to change any of the credential information. One would imagine that this leads to an uncontrollable amount of password request phone calls, but our experiences do not surprisingly enough indicate such behavior.

The credentials can only be delivered to the respondent by a letter. In case of forgotten password, it (only the password) can be given by a telephone call from Statistics Finland to a pre-defined contact person in the respondent organization. All other delivery methods are strictly prohibited.

Credentials are typically changed once in a year. This usually means that the respondents of monthly and quarterly inquiries are provided with new credentials by letter in the beginning of each year. For yearly inquiries the credentials are always generated again when the survey begins. If the respondent in the organization changes, new credentials are provided. In situations, when also the proprietorship of the organization changes, the information provided by the previous proprietor is erased.

6.3 Accountability

Identifying the user activity matters sometimes a great deal. Our approach is multithreaded. Basic logging facilities are used on the hardware level (firewalls) and on the daemon level (web server, operating system).

In addition, the XCola engine itself provides extensive accountability features. Every single user action in the system, from arriving to the login screen until the logout, is traced and logged. Of course, the detailed logging information is not exploited on a single respondent level, to preserve the common privacy, unless a specific reported malfunction needs to be solved. The logging information will be used on aggregated level though, to provide general information, for example, about usability issues or failure rates.

6.4 Integrity

It is extremely important for the credibility of the data collector that each user can only access and modify his own data. This can of course be quite well achieved by adequate user authentication and authorization techniques embedded with application level protection mechanisms. The feature developed for the latest version of XCola, is a user specific encryption technique. Once the user has been successfully authenticated and authorized, it is possible to provide the personal initialization vectors or secret keys for the use of encryption and decryption of the user data. While this seems to be the ultimate integrity solution, it still poses some great risks. What about if the management of keys somehow fails or the key storage corrupts? How can the data be recovered then? It simply can not.

In addition, heavy use of encryption could affect the performance of the system, thus weakening the usability. So, one really needs to think hard when to really use these extreme measures. For example, there is probably little need for encrypting the contact information or any other information that is publicly available anyway.

7 Summary

More and more application logic is offered nowadays in the World Wide Web. It is not a safe environment. In fact, it is full of hostility expressed in many nasty ways. Typical threats targeted towards web applications are command injection attacks, cross site scripting, even eavesdropping, not forgetting the threats stemming from social engineering areas.

There are several ways of preventing the hostilities aimed for the online applications. The communication can and usually must be secured when dealing with confidential information. The users must be appropriately authenticated and the unauthorized use of confidential data must be prevented by any means.

Even though there are many security solutions, most of them contain some weaknesses by nature. It is probably not even worthwhile to seek a perfect security solution; presumably one does not exist or is not very user friendly and lacks tremendously in usability. The more important aspect for the organization is to conduct a security strategy. In such strategy the assets, threats and vulnerabilities are recognized and tolerable levels are defined. The most important part of the security strategy is the continuous monitoring of the ever evolving hostile environment.

References

[1] Bhimani A., Securing the Commercial Internet. *Communications of ACM* 39, 6, 1996, ACM Press, pages 29-35.

[2] Braz C., Robert J-M., Security and Usability: The Case of the User Authentication Methods. Proc. *18th International Conference on Association Francophone d'Interaction Homme-Machine*, Montreal, Canada, April 2006, ACM Press, pages 199-203.

[3] Gaw S., Felten E. W., Password Management Strategies for Online Accounts. Proc. *2nd Symposium on Usable Privacy and Security*, Pittsburgh, PA, USA, July 2006, ACM Press, pages 44-55.

[4] Huang Y-W et al., Securing Web Application Code by Static Analysis and Runtime Protection. Proc. *13th International Conference on World Wide Web*, Edinburgh, Scotland, May 2004, ACM Press, pages 40-52.

[5] Kals S., Kirda E., Kruegel C., Jovanovic N., SecuBat: A Web Vulnerability Scanner. Proc. *15th International Conference on World Wide Web*, Edinburgh, Scotland, May 2006, ACM Press, pages 247-256.

[6] Kirda E., Kruegel C., Vigna G., Jovanovic N., Noxes: A Client-Side Solution for Mitigating Cross-Site Scripting Attacks. Proc. *ACM Symposium on Applied Computing 2006*, Dijon, France, April 2006, ACM Press, pages 330-337.

[7] Lao G., Wang L., Application of E-Commerce Security Management Strategy in Banking. Proc. *7th International Conference on Electronic Commerce*, Xi'an, China, August 2005, ACM Press, pages 627-632.

[8] Li C., Pahl C., Security in the Web Services Framework. Proc. *1st International Symposium on Information and Communication Technologies*, Dublin, Ireland, September 2003, Trinity College Dublin, pages 481-486.

[9] Pinkas B., Sander T., Securing Passwords against Dictionary Attacks. Proc. *9th ACM Conference on Computer and Communications Security*, Washington, DC, USA, November 2002, ACM Press, pages 161-170.

[10] Park J. S., Sandhu R., Ahn G-J., Role-Based Access Control on the Web. *ACM Transactions on Information and System Security* 4, 1, 2001, ACM Press, pages 37-71.

[11] Rahman S., Nguyen T. A., Yang T. A., Developing Certificate-Based Projects For Web Security Classes. *Journal of Computing Sciences in Colleges* 21, 5, 2006, Consortium for Computing Sciences in Colleges, pages 28-37.

[12] Su Z., Wasserman G., The Essence of Command Injection Attacks in Web Applications. *ACM SIGPLAN Notices, Conference Record of the 33rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, Charleston, SC, USA, January 2006, ACM Press, pages 372-382.

[13] Taylor K., Murty J., Implementing Role Based Access Control for Federated Information Systems on the Web. Proc. *Australasian Information Security Workshop Conference on ACSW Frontiers*, Adelaide, Australia, January 2003, Australian Computer Society Inc., pages 87-95.

[14] Tari F., Ozok A. A., Holden S. H., A Comparison of Perceived and Real Shoulder-surfing Risks Between Alphanumeric and Graphical Passwords. Proc. *2nd Symposium on Usable Privacy and Security*, Pittsburg, PA, USA, July 2006, ACM Press, pages 56-66.

[15] Viega J., Messier M., Security is Harder than You Think. *Queue* 2, 5, 2004, ACM Press, pages 60-65.

[16] Xia H., Brustoloni J. C., Hardening Web Browsers against Man-in-the-Middle and Eavesdropping Attacks. Proc. *14th International Conference on World Wide Web*, Chiba, Japan, May 2005, ACM Press, pages 489-498.