# Improving Web Application Security by Eliminating CWEs

Weijie Chen, China

INFSY 6891 Software Assurance

Professor  Dr. Maurice Dawson

15 December 2015

## ABSTRACT

This study examined improving web application security by eliminating CWEs. With the developing of web application, securities are becoming more and more important and complicated. The article states the current situation of the web application security and provides possible solution method. By understanding the CWEs and eliminating CWEs can help web application developers to solve or minimizing the security problems. The study listed three CWEs as examples to show how to understand and eliminate CWEs. Problem-solving based method can help people quickly understand the advantage of using CWEs to improve web application. Cloud Computing is also covered in the study for the future research directions.

**Keyword:**

Web Application - Web application is where data is being operated between the WAN and internal network. A good web application control its vulnerabilities.

OWASP - Open Web Application Security Project is a standard organization for the web application security. It creates Top 10 lists to help improve security.

CWE - Common Weakness Enumeration help developer better understand the weaknesses and vulnerabilities of the web application.

CWE-602 - It shows that the application is relies on the client side to hold data.

CWE-79 - Known as XXS (Cross-site Scripting), means that this web application does not verify client side input or doesn’t have enough verification on the input.

CWE-601 - Phishing attacks, redirect user to an external websites.

Cloud Computing – Internet based computing, it is on demand and has three service models: PaaS, IaaS, and SaaS.

## INTRODUCTION

With the development of society and economy, web related services are becoming more and more important. Such as online shopping, social media, emails, cloud based computing, etc. However, when people are enjoying the conveniences and benefits from these service, they must realized that web related service securities are also becoming more and more important. There are many web applications behind these web rated services, so how to effectively managing these web applications and control its risks are very necessary. Improving web application security can help people better and safer to enjoy web services.

Web application security is not like common application security. For common application security, developer can use firewall or physical isolation to separate the application and the WAN (Wide Area Network). Web applications are different. Web applications are running in the server and open to the public network. The function of the web applications is for people to use on the web, so it has to be connected to the WAN. Because web application security is a new area, so it is a big challenge for people to solve or improve this area. A good solution should provide confidentiality, integrity, and availability to the web applications.

There are varieties of tools and methods to test a web application and help developer to improve web application security. Eliminating CWE (Common Weakness Enumeration) is one of the methods to achieve the goal. This article focus on eliminating CWEs to improve web application security. CWE can help developer better understand software weaknesses and vulnerabilities. Most web application threats can be defined by CWEs, includes cross-site scripting, SQL injection, cross-site request forgery path disclosure, and so on. After developer review and understand these CWEs for their web applications, they can improve the threats by eliminating CWEs.

To begin with, the following articles will provide clear view for web application security and CWEs in the background part. Secondly, showing people how to improve web application security by using CWEs. The security problems part will identify some web application security problems by listing CWEs, then using those CWEs as examples for the security solutions and recommendations part to show the possible ways to fix or improve the threats of the web applications. The future research directions part will discuss cloud computing, which is a new area for the security and web based.

## BACKGROUND

### Web Application Security

“As more and more applications find their way to the World Wide Web, security concerns have increased. Web applications are by nature somewhat public and therefore vulnerable to attack. Today, it is norm to visit Web sites where logins and passwords are required to navigate form one section of the site to another” (Cross, Michael). So web application is where data is being operated between the WAN and internal network. Web application should keep the data encrypted or protected between the WAN and internal network. But web application always have some level of unprotected area. Through these unprotected area, hacker can find the vulnerabilities of the application and use them to attack the application. A good web application shall block all the vulnerabilities so that hacker couldn’t attack it.

### OWASP

OWASP, the Open Web Application Security Project is the emerging standards organization for the web application security. OWASP was formed on September 9, 2001. OWASP provides web application developer a guide to check their web application security. OWASP also creates Top 10 for the web application security. Top 10 is a list of the 10 most important web application security risks. “These include SQL injection, used by hackers to target Vodafone Iceland; cross-site scripting (XSS), which left Microsoft Office 365 open to attack; open redirects, which presents issues for Facebook; and insecure direct object references, which saw Yahoo's servers open to root access.” (Journal of Engineering). Many large companies are using their awareness document to ensure their web application securities. Top 10 list can be included in CWEs. For example, CWE-928 is one of the CWEs, it contains the weaknesses in OWASP Top 10 from 2013.

### CWE

CWE, Common Weakness Enumeration is an application/software community project that focus on creating a catalog of weaknesses and vulnerabilities for the application/software. On another hand, it created a standard and measurable way for people to effectively understand the weaknesses of the application. CWE also provide a platform for developer to discuss their application securities, because they can use the same CWE number to identify their situations. There are many ways to get the CWEs for the web application. ZAP Proxy is one of the testing tools listed on the OWASP home page that can help web application developer to scan their application and allow them to find security vulnerabilities.

## SECURITY PROBLEMS

### CWE-602: Client-Side Enforcement of Server-Side Security

In the sample word, CWE-602 shows that the application is relies on the client side to hold the data. “Just as you shouldn't trust the client to hold sensitive state data, you shouldn't trust the client to tell you whether it's authorized to perform an action.” (Howard) Hackers can easily bypass the client side validation checks, allowing unexpected input to pass into the application. There are many ways to bypassing client side validation. For example, disabling the JavaScript, it will stop all JavaScript based controls and it is very easy to do. Another example can be using a proxy tool like Tamper Data to stop any submitted data and adjust it before sending it to the server.

### CWE-79: Improper Neutralization of Input during Web Page Generation (Cross-site Scripting)

This is a very common security problem, and it has some level similar to the SQL injection (SWE-89) security problems. Many hackers use these two vulnerabilities to attack the web site. CWE-79 is defined as Improper Neutralization of Input during Web Page Generation, but it is well known as XXS (Cross-site Scripting). For a hacker, XXS means that this web application does not verify client side input or doesn't have enough verification on the input. Then hackers may craft a client side script to make web application doing things incorrectly. Another formal to explain this hacker's behavior is they create untrusted data and enter into the web application from a web request, web application then execute the malicious script within those untrusted data.

### CWE-601: URL Redirection to Untrusted Site (Open Redirect)

URL Redirection to untrusted site is also well known as open redirect. This CWE is famous and from the TOP 25 CWE lists. Most people must have experiences with this CWE while they are using web application. For example, when people are checking their emails, if click an external site link from the email, they may be directed to a malicious websites. In the simple words, it is phishing attacks. It is really hard for web application to limit those phishing attacks, because web application cannot control human behavior.  A redirected malicious may contains malware or virus that can harm the user's computers and devices.

## SECURITY SOLUTIONS AND RECOMMENDATIONS

### Solutions for CWE-602: Client-Side Enforcement of Server-Side Security

Testing is can help developer to minimize the risks of this CWE. By using tools and techniques such as penetration testing that require manual analysis, the tester can record and adjust an active session. Recommendations for this CWE is to duplicate the security checks on the server side and making sure the inputs are coming from a trusted source. This method can support intrusion detection and provide feedback to the user for the expectations for valid input. It may also reduce the time for processing the unexpected input issues.

### Solutions for CWE-79: Improper Neutralization of Input during Web Page Generation ('Cross-site Scripting')

There are several methods to solve or reduce the risks for this CWE. There are two methods that will be discussed in this article, which is set the session cookie and filtering the inputs. This two methods are useful and effective for most web application. For the first method, it is very easy and simple, just need turn the session cookies to be HttpOnly, this behavior can prevent the user's session cookies from being accessible to malicious client side scripts. The second method is the filtering. By facing client side input, web application can filter those malicious client side scripts. For example, when the web application detect single quotation marks (that may cause malicious SQL commend), application can escape or filter those information. Filtering these malicious script can be defined as black list filtering. One disadvantage of black list filtering is it is impossible to cover all the malicious scripts. Hackers can always develop some scripts that pass the black list filtering. In that situation, wen application may use white list filtering. White list filtering means the web application only accept certain inputs and does not support other inputs. White list filtering may limit the range of the inputs, but it provide deeper protection than the black list filtering.

### Solutions for CWE-601: URL Redirection to Untrusted Site (Open Redirect)

The most effective method to alert people for this CWE is to create a pop up window. Every time when users want to direct to an external link that is not belong to this web application, the web application will have a disclaimer page to provide user a clear warning that they are direct to a different web application. So if people realized that they have been redirected to another web application, they can make a better decision without an accident click. Web application could also design a white or black list for the redirection link. It will help users avoid to visit some malicious websites.

## FUTURE RESEARCH DIRECTIONS

### Cloud Computing

Cloud computing has some level of similarity to the web application, both of them are related to the Internet. There are many definitions of cloud computing, but one of them stand out by NIST (National Institute of Standards and Testing), which is an agency of US to develop standards and measures. “The NIST Cloud Computing definition has three sections: 1. Essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.); 2
Service models (software-as-a-Service, platform-as-a-Service, and infrastructure-as-a-Service); 3. Deployment models (private cloud, community cloud, public cloud, and hybrid cloud).” (Carstensen, 26) Technologies are the foundation of the cloud computing, such as virtual machine technology, hardware technology, massive data management technology, etc. A simple way to understand cloud computing is that using by demand, adjust by demand.

As mentioned above, cloud computing has three service models: Infrastructure as a service (IaaS), Platform as a service (PaaS), and Software as a service (SaaS). IaaS is the most basic model, users can get the services from those basic computing through the Internet. PaaS provide a software development platform to the users, help users to develop, run and test their software on a cloud environment. PaaS is one of the applications of the SaaS model. PaaS can help SaaS to grow faster, especially in the application development area. SaaS provide a way for user to use the software through the Internet without purchase the software. Software supplier can rent out their software to their users. The following figure 1.1 provide some examples for each service models.

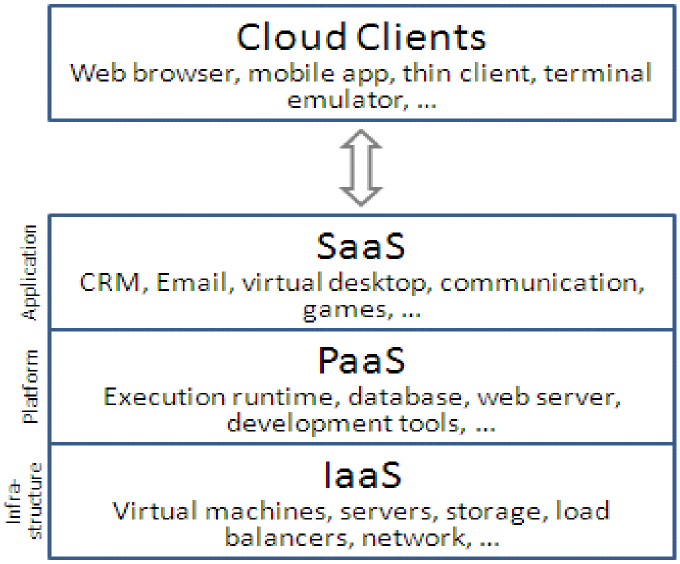
Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
SaaS
CRM, Email, virtual desktop, communication, games, ...
PaaS
Execution runtime, database, web server, development tools, ...
IaaS
Virtual machines, servers, storage, load balancers, network, ...
Application
Platform
Infra-structure

**Figure 1.1**

Nowadays, cloud computing is growing very fast due to its characteristic: virtualization, reliable, cost saving, versatility, and so on. At the same time, cloud computing security problems are also becoming bigger and bigger. “Cloud computing and web services run on a network structure so they are open to network type attacks.” (Danish & Hassan) Both cloud computing security and web application security are part of the cyber security, they are all open to network type attacks. The future discussion for the cloud computing security will also becoming more and more important like the web application security.

## CONCLUSION

Web application security is a huge area that everyone who is using Internet must face. There are many ways to improve this area, but using CWE is one of the best way to minimize the risks. From the CWEs examples on the study, improving web application security by eliminating CWEs is effective and useful. By understanding those CWEs, developer can have a clear view of the vulnerabilities for their web applications. Solution to those vulnerabilities can be followed by eliminating those CWEs. The future research directions can be focus on the cloud computing, since it is also Internet related and has some level of similarity to the web application.

## REFERENCES

Cross, M., ebrary, I., & Books24x7, I. (2007). Developer's guide to web application security. Rockland, MA: Syngress Publishing.

OWASP puts focus on growing web application security risks at AppSec europe. (2014). Journal of Engineering, 935.

Howard, M. (2009). Improving software security by eliminating the CWE top 25 vulnerabilities. IEEE Security & Privacy, 7(3), 68. doi:10.1109/MSP.2009.69

Carstensen, J., Golden, B., Morgenthal, J., & Books24x7, I. (2012). Cloud computing: Assessing the risks. Ely, Cambridgeshire: IT Governance Publishing.

Bikeborg. (2012). Cloud computing layers.png, Retrieved from URL (https://commons.wikimedia.org/wiki/File:Cloud_computing_layers.png)

Danish Jamil, & Hassan Zaki. (2011). Cloud computing security. International Journal of Engineering Science and Technology, 3(4), 3478-3483.