

# Design and Implementation of Open Source Web Application Security Tool

Chiung-Wan Chan and Chung-Huang Yang

Graduate Institute of Information and Computer Education,  
National Kaohsiung Normal University, Taiwan  
jocelynchan25@gmail.com, chyang@nknucc.nknu.edu.tw

**Abstract.** Most vulnerability of Web application program were caused by data which are not verified when they input to the system. That's why new attack to those vulnerability happens more often. Just because of this, enterprises should focus on the security of web during program development, this can make sure the vulnerability won't be detected because of program bug. This research is to utilize the security framework provided by the open source code HDIV, making use of API to prevent malicious Web attack. With the tool, administrator can take precautions and stop attacking maliciously in advance. With the security code generated by the tool automatically, the programmers do not need to spend extra time to understand the framework. With the operation of this tool, programmer can foresee the vulnerability during development phase and then take actions to prevent it in advance.

**Key words:** Web application security, web application security framework, HDIV

## 1 Introduction

It is extremely frequent that websites were attacked maliciously in recent years. If website administrator uses the traditional security safeguard procedures to avoid the external threat, it is not enough to deal with the vulnerability that program exists, and may put the enterprise in danger. To improve competition advantage and reduce operating cost for enterprise, more and more information systems were established. It may also bring other risk due to the exposure of those systems. You may hear that there are lots of websites were invaded by hackers, the threat of website safety is increased day by day.

In addition, if programmers feel the time pressure to release it, they can only focus on the accuracy of function, it is difficult for them to consider the design of the security in a short time. The most programmers perhaps are familiar with programming language, but not conscious of the security with this language. Their consciousness to the information safety is insufficient, it results in the negligence on security, it is unable to give consideration to software security. When they don't have systematic regulation about security, it's easy to develop unsafe website system. Set of safe development norms, it will be very easy to develop

the unsafe website system, offering more opportunities to be invaded in the future. [15]

According to the ten most critical Web application security vulnerabilities announced by OWASP 2007 [19], most attacks and vulnerabilities were found at application layer [17]. Most of those vulnerabilities were detected due to system wont verify those data in advance. The cost of hardware procurement is far higher than security program and software which can help verify data when receiving them. Hence, its recommended that enterprise should take action on severe data verification technology to protect the Web application program. For those data coming from the end users , server must check it first. Before sending out data from server, data should be also inspected to stop endless new attack [6] [16].

HDIV is a Java Web applicatoin security framework. To utilize this framework, user needs to understand its API's operation method in depth. For the programmer who would like to establish Web with the framework, he has to spend some time understanding that. This research is to make use of those security frameworks, design and build user friendly interface by oneself, foreseeing the most common attacks and security vulnerability, preventing it in advance during development. With HDIV security framework, this research makes the user interface to stop being attacked maliciously, and strengthens the function of that framework which is not offered at present HDIV, e.g. automation on data verification, Web management tool , pre-warning system, etc. What the tool provided in this research, it can help website administrator block hostile attacking immediately according to detailed log and alarm system without extra modification on old system structure. Administrators dont have to spend time studying HDIV framework, they can save money and time with the tool. Its easy for them to setup and monitor and achieve the purpose of the security.

## 2 Literature Review

### 2.1 The progress on Web application program [20]

Tim Berners-Lee proposed first concept about the global information network (World Wide Web) in 1990 [1], naming this platform web, this is the initial point of webpage development. Web page is only static picture at first till CGI was released in 1993, then webpage entered dynamic era. In the end, only less people use CGI because of the defect of its security question.

In addition to Java and internet network appearing in 1995, the introduction of Applet [22] also raised a new chapter of dynamic webpages, which also can follow strict safety inspection to stop potential safety risk. On the contrary, it requires Java virtual machine at user side to function properly, this also hampers the development of this program. At the same time , Netscape promotes a script language JavaScript [23], it also has the bottleneck that not every browser can support it, meaning with same script there will be different results shown. It also has vulnerability with the Javascript, and is not easy to debug for programmer. Eventually, this Javascript was used on creation of simple table application.

Until the introduction of JSP, PHP and ASP successively, MVC concept was implemented, it allows users not need to install each of huge software anymore, they can utilize the browser to execute the application program. AJAX( Asynchronous JavaScript and XML ) is a webpage application technology with interaction structure. This is created by Jesse James Garrett in 2005 [5], using the greatest advantage of Ajax to maintain the data without renewing whole page. This enables Web application program to reflect user's instructions promptly, similar as you double click the program shortcut on your personal computer desktop.

## 2.2 Study on tools to detect vulnerability of Web application

Most Web vulnerability issues are caused by the unverified data. Inside MOPS [3] and SPLint [4] static analysis technology announced in 2005, they were already used on the analysis of Web application source code. However, this technology got failure due to it didnt consider the behavior of Web in execution time of application program appropriately. Until 2003, Y.W. Huang published the black-box safety test tool for Web application program, so-called WAVES(Web Application Vulnerability and Error Scanning) [7], it can generate report to recognize the vulnerability of the Web when analyzing it.

The tool used on Web program safety analysis and inspection (WebSSARI) [8] [9] was launched in May, 2004. It can evaluate the safety level of Web application program automatically and get good result with real tests. Livshits and Lam also proposed one tool LAPSE [14] in Oct.,2005, it can integrate analysis result into Eclipse. It can help find out the vulnerability of incoming unverified data based on static analysis, further it looks for JAVA source code to inform user of all potential weakness. OWASP (Open Web Application Security Project) also issues WebScarab [18] projectAit also integrates the tools to inspect Web security, making the security check efficiently.

Lin and Chen [11] issued an automatic tool in 2006, it can automatically add verification function procedure to each server to get rid of malicious injection attack. They also published a tool [12] with the defense mechanism to hinder and filter malicious attack in IEEE international seminar in 2007, Besides, they presented a defense system [13] in the AINAW international seminar in 2008, this system includes the safety testing framework and verification function to reduce the malicious system attack.

## 2.3 Application software structure

Software structure is a semi-finished application program to reuse when the data input and output is almost similar. It provides a template for programmer to adopt the software structure for their software development. It allows programmer not to write duplicate language program to save money and time [2].

### 3 Web application software security framework

#### 3.1 Introduction of HDIV

HDIV [21] is safe extension framework authorized by Struts of Apache used on Java Web application. It aims at the protection of malicious Web application attack. HDIV can eliminate not only the top 10 network security vulnerabilities listed on OWASP in 2007, but prevent and solve it in advance under program development stage.

It's very easy for programmers to make mistake if they have to verify data column by column. They might forget verifying column data at some pages because it has to do this check for each column in traditional data verification mechanism. That's why HDIV provides generic rule to allow users to set up the data verification rules for their whole WEB. It can help filter the content of each request and hide important information to get rid of data being tampered to further prevent the attack caused by most application programs with vulnerability.

For example SQL injection attack, cross-site scripting and parameter tampering. It can apply to the application program that had already developed before, like Struts, Spring or JSTL. With the tool, you dont have to change your source code then you can use the framework on your structure. It enables programmers to save the development time. Enterprise can also save money because it is open source code with security framework. Web administrator can take advantage of the structure to control the security, establish Web protection to mitigate the treat from program attack with the isolation of system structure concept.

#### 3.2 State Objects

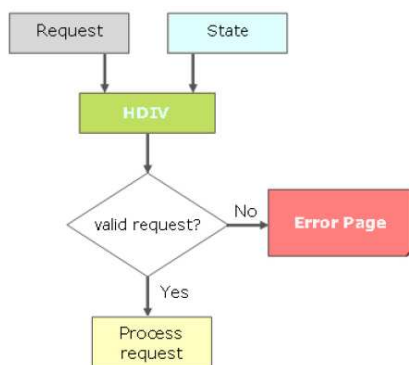


Fig. 1. HDIV data verification flowchart

HDIV adopts "State" to protect the integrity and confidentiality of data. Figure 1 is the data verification flow, it points out clearly that the frame will

generate one corresponding "State Object" at server and user side for every request from user. Every request will be verified by HDIV frame in comparison with the data from user side and server. The parameter was made arbitrarily including HDIV State to eliminate the attack from CSRF(Cross Site Request Forgery).

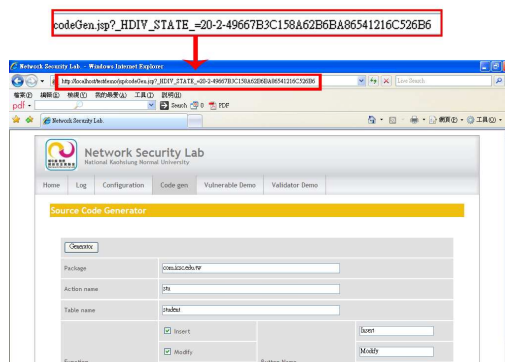


Fig. 2. HDIV State

## 4 System implementation

### 4.1 Environment creation installation and system development

This research makes use of Struts 2 structure as the security framework. Java language was chosen as development tool, system construction requirements shown at table below:

Environment	Description
Web Server	ApacheBTomcat
Database	support MySQL or MS-SQL
Java Virtual Machine	Java Runtime Environment (JRE)

### 4.2 System structure

The system consists of 5 major functions:

**Log.** Graphic user interface provides the log format setting and message to specific appender parameter. Output can be available from monitor, file, mail or database. Graphic log makes it detailed to administrator if they want to inquire the Web attack log.

2009/03/04	http-80-3	INFO	message HDIV_PARAMETER_NOT_EXISTS /testdemo/jsp/xss.jsp _HDIV_STATE_ null 192.168.0.100 192.168.0.100 user
------------	-----------	------	--

Fig. 3. Log message

**Configuration.** With graphic interface design, user can change the setting with it.

**Generate security code automatically with Code gen function.** With simple interface setting, it can generate security program code including web page and backend verification data to avoid malicious vulnerability attack.

**Vulnerable Demo.** With the application program generated by Code gen, it can defend the malicious vulnerability attack.

**Validator Demo.** It demonstrates valid verification for each data type.

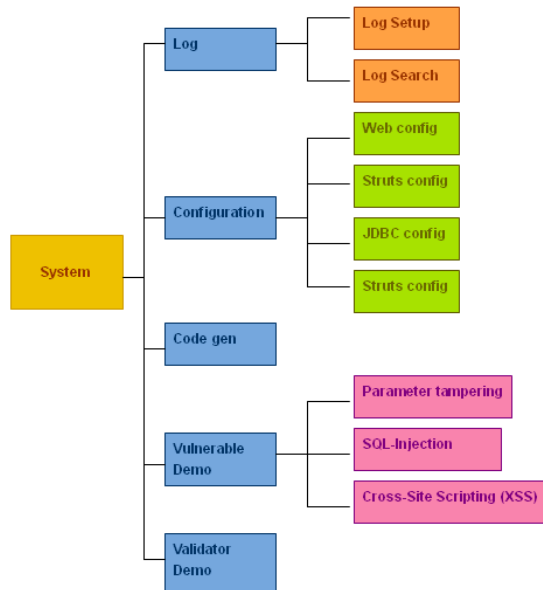


Fig. 4. System structure

### 4.3 Characteristic of system

**Graphic UI setting.** With graphic user interface setting, user can easily modify setting of system structure, its not necessary for them to know xml setting. Generate security program code automatically. It can generate security program code automatically for data verification. With simple interface setting, it can generate security program code including web page and backend verification data to avoid malicious vulnerability attack.

**Advantage to programmer.** It can be applied on Struts 1.x, Struts 2.x, Spring MVC and JSTL as well as the application programs developed before by enterprise. It can help save the development time and block most of the malicious attacks even without enough data security concept and know-how.

**Detailed attack log.** The Web management tool is easy to use, it can show alarm automatically with your setting. With the log, administrator can understand if there is malicious attack to the Web then he/she can take precaution to the next attack.

### 4.4 System demonstration

List several security vulnerabilities to system that can be avoided with the tool.

**Cross-Site Scripting Attack.** Hackers are good at inserting illegal and malicious program codes into Web with the trust from web master. When user logs on the webpage, the malicious program will transfer users ID and password to remote Web that hacker specifies. In the future, the hacker will log on the Web on behalf of you to steal the private information to gain interest from that.

With malicious script command input, like " %3c%73%63%72%69%70%74%3e%20%61%6c%65%72%74%28%20%27%20%58%53%53%20%76%75%6c%6e%65%72%61%62%69%6c%69%74%79%20%27%20%29%20%3b%20%3c%2f%73%63%72%69%70%74%3e%20" in figure 5, it translates "<script>alert('XSS vulnerability'); </script>" into hex value. This help block XSS attack by inspecting the encoding commands with error message shown.

**SQL injection.** For the security vulnerability incurred at application program and database, hacker inserts malicious SQL commands when requesting the data transfer. If the application program neglects the verification and accept the commands, the server will regard it as normal commands and response it. It results in leaking out confidential data from server. With the malicious SQL injection commands attack, e.g. "'; DROP TABLE tbAccount--", it will block the SQL injection attack with error message.

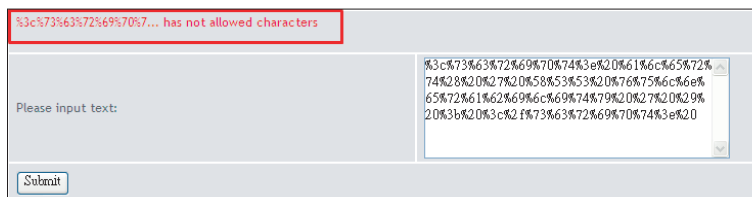


Fig. 5. Error message

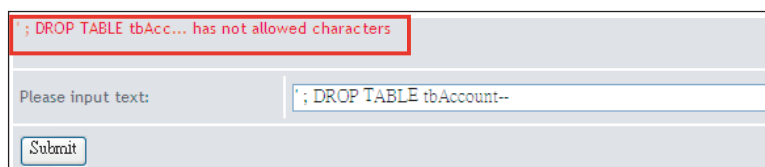


Fig. 6. Error message

**Parameter tampering.** Parameter tampering is one basic type of attack. When server transfers data to user, hacker tampers the input parameter to get customer confidential information. Once higher level authority was available, hacker will perform serious destruction.

The following example enables user to get user data with hidden field. You can see the user J2EE data from source code in figure 7 but actually its username was hidden and replaced by another value (figure8) to make sure the confidentiality to get rid of parameter tampering attack.

```
<s:form action="tampering">
  <s:hidden name="name" value="j2ee" />
  <s:submit value="Parameter tampering" align="true" />
</s:form>
```

Fig. 7. Hidden field source code

**Information Leakage and Improper Error Handling.** If error messages generated by application program provide too much useful information, hackers may make use of them to find the vulnerability of the system. The Web with security framework will direct to error page (figure9) when errors were detected, hackers cant judge the error messages to plan new attack.

```
<form id="tampering" name="tampering" onsubmit="return true;"  
action="/testdemo/jsp/tampering.action" method="post">  
<table class="wwFormTable">  
<input type="hidden" name="name" value="" id="tampering_name"/>
```

Fig. 8. Source code in browser



Fig. 9. Page without error message

## 5 Conclusion

In recent years, Web sites suffer from malicious attack more frequently. Security threat to system is getting serious with improper control method on Web application program. If administrator takes traditional defense actions to protect the system from invading, its not enough to deal with the vulnerability caused by program itself. This will also put enterprise information system in danger.

Its passive for enterprises to solve security issue of the Web application program at present. Defense with firewall is only to give temporary relief not to solve it at all, vulnerability on program itself is not improved. Thats why enterprises should put the security problem with higher priority during application program development. However, programmer may not have the sense on that due to they lack of security awareness or related know-how. They are not well aware that those program vulnerabilities may be used by hackers to penetrate into enterprise information system for their illegal stealing. In addition, under the urgent pressure of time to launch the program on time, they only can focus on functionality not vulnerability. To be honest, its not easy to take safety logical design into consideration if they are not specialized in information security protection. To solve this kind of problem, programmers should also pay attention to the side effect if their application programs were developed with vulnerability. They can also take advantage of the tools recommended in this research to establish safety Web rapidly and solve the issue within program. They dont have to change the structure of current system then they can block the most known Web attacks. It can help programmer to save the time and meet the requirement to establish Web with security even though they dont have knowledge on the information security know-how.

## References

1. Berners-Lee, T.: The WorldWideWeb browser, <http://www.w3.org/People/Berners-Lee/WorldWideWeb.html> (1990)
2. Cavaness, C.: Programming Jakarta Struts, 2e. O'REILLY (2004)
3. Chen, H., Wagner, D.: MOPS: an Infrastructure for Examining Security Properties of Software. ACM Conference on Computer and Communication Security (2002)
4. Evans, D., Larochelle, D.: Improving Security Using Extensible Lightweight Static Analysis. IEEE Software, 42-51 (2002)
5. Garrett, J. J.: Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/ideas/essays/archives/000385.php> (2005)
6. Holz, T., Marechal, S., Raynal, F.: New Threats and Attacks on the World Wide Web. IEEE SECURITY & PRIVACY (2006)
7. Huang, Y.W., Huang, S.K., Lin, T.P., Tsai, C.H.: Web Application Security Assessment by Fault Injection and Behavior Monitoring. 12th Intl World Wide Web Conference (2003)
8. Huang, Y.W., Yu, F., Hang, C., Tsai, C.H., Lee, D. T., Kuo S.Y.: Securing Web application code by static analysis and runtime protection. Proceedings of the 13th International World Wide Web Conference (2004)
9. Huang, Y., Yu, F., Hang, C., Tsai, C.H., Lee, D.T., Kuo, S.Y.: Verifying Web applications using bounded model checking. Proceedings of the 2004 International Conference Dependable Systems and Networks (2004)
10. Husted, T., Dumoulin, C., Franciscus, G., Winterfeldt D.: Struts in Action: Building Web Applications With the Leading Java Framework. Manning Publication Co. (2002)
11. Lin, J.C., Chen, J.M.: An Automatic Revised Tool for Anti-malicious Injection. IEEE International Conference on Computer and Information Technology (2006)
12. Lin, J.C., Chen, J.M., Wong H.K.: An Automatic Meta-revised Mechanism for Anti-malicious Injection. Proceedings of Network-Based Information Systems, 98-107 (2007)
13. Lin, J.C., Chen J.M., Liu, C.H.: An Automatic Mechanism for Adjusting Validation Function. 22nd International Conference on Advanced Information Networking and Applications, IEEE Computer Society, 602-607 (2008)
14. Livshits, V. B., Lam, M. S.: Finding Security Vulnerabilities in Java Applications with Static Analysis. The 14th USENIX Security Symposium (2005)
15. Meier, J.D.: Web Application Security Engineering. IEEE SECURITY & PRIVACY, 16-24 (2006)
16. Nichols, E. A., Peterson, G.,: A Metrics Framework to Drive Application Security Improvement. IEEE SECURITY & PRIVACY, 88-91 (2007)
17. Ollmann, G.: Writing secure code. Network Security, 16-20 (2007)
18. OWASP.: WebScarab Project, <http://www.owasp.org/>
19. OWASP.: The Ten Most Critical Web Application Security Vulnerabilities, [http://www.owasp.org/images/e/e8/OWASP\\_Top\\_10\\_2007.pdf](http://www.owasp.org/images/e/e8/OWASP_Top_10_2007.pdf) (2007)
20. Tilley, S.: Five Years of Web Site Evolution. Proceedings of the Fifth IEEE International Workshop on Web Site Evolution (2003)
21. Velasco, R., Rabadan, A., Enrique, M., Vicente G.: Protect your web applications against attacks Reference v2.0.4 (2005)
22. Wikipedia.: Java applet, [http://en.wikipedia.org/wiki/Java\\_applet](http://en.wikipedia.org/wiki/Java_applet)
23. Wikipedia.: JavaScript, <http://en.wikipedia.org/wiki/JavaScript>