Validation and Integrity Mechanism for Web Application Security

Saher Manaseer[1], Ahmad K. Hwaitat[2]

[1,2]Department of Computer Science, The University of Jordan, Amman-Jordan

***Abstract*—*Recently the world of web applications has witnessed a huge wave of attacks that caused a lot of web applications to get defaced and for a lot of businesses to lose financially and to lose the integrity between those companies and their users .since web hacking became common knowledge on the Internet and most of the defacements that happens on daily basis are going under a random way, the security specialists are trying to get new solutions that will help reduce the possibility of causing much damage to web applications even after the process of infiltration.*

In this research an enhanced web application solution was proposed, which is a validation and integrity component that can be easily installed on any web application firewall (WAF), in order to help in solving the problem mentioned and raise the level of trust between user's and web application hosts/owners also restore the lost data caused by hacking attempts in simple and systematic way.

Keywords*—*Attack Prevention, Application Firewall, Cyber security, Firewall component, Internet Attacks.

I. INTRODUCTION

Recently, the growth of web applications usage started to rise in a large way, which made the possibility of more web application vulnerabilities to appear and more hacking attempts to happen on daily basis [1]. The development of web applications witnessed a huge revolution along with the revolution of the Internet. Web applications are becoming very essential in the daily activities of persons and companies [1][2]. Some of these activities contain confidential information about the user such as credit card numbers, passwords, and money authorization transactions information.

The security of the user's information is a major concern for all companies' owners and administrators due to the successful attacks against web applications across the history. Many attackers may be able to compromise some web applications and access private data across the global net by exploiting several known and un-known web application vulnerabilities [3][4]. Such cyber-attacks can cause financial loss for many parties including private companies and any other type of infrastructure. They are also a main reason for users to lose trust and integrity in many private and governmental institutions. Therefore, there is a major need for developing researches and find methods for preventing and detecting any possible attack against such web based infrastructure, securing databases and help making the data more private for the users. Furthermore, there is a need to take pre-steps and create a method to restore lost data after the occurrence of the attack as fast as possible and help raise the integrity level of the content inside the web application.

II. RELATED WORK

Double Guard is an application used for checking the intrusions in multi-tier application. This application is used for back-end and front-end and its independent, it is also operated in dynamic and static servers in the web, these servers provide better protection for the application and information [5].

An approach based on learning was presented by [6], for securing web services against SQL and XPath Injection attacks. Valid requests patterns were learnt by the approach, and that is called the learning phase, then it was able of detecting and aborting requests which might harm the server, which is called the protecting phase. Some heuristics might be used to deal with suspicious cases when there isn't a possibility to have a finished learning phase. The technique was executed to keep TPC-App services safe, and for opening source service effect.

In [7], a mechanism was developed for the detection of SQL injection, by employing a Reverse proxy and MD5 algorithm to watch SQL injection in input. Using rules of grammar expressions for checking SQL injection in URL's. No changes are done in the application's source code by their method. Investigating and decreasing the attack is automatically done. The increasing in the number of proxy servers makes web applications able to handle any number of requests with no delay of time, and makes it able to protect the application from SQL injection attack.

A novel and an effective solution for the problem of XSS was suggested by [8], its purpose was to detect all SQLIA kinds. Their technique also checked the assigned value for a single quote, double dash and space given by the user through input sections. A space, single quotes or double dashes are ought to be used by the attacker in his input, when he is scripting an SQL injection.

The method of [9], is for combined static analysis and runtime validation. Legitimate queries are found in static analysis, it can be operated by the application, and it can also reform them into patterns of structure query. The query patterns resulted from it are kept in a separate respective tables, and that is for the reason of decreasing the runtime validation overhead the runtime query in the runtime validation is made into patterns and a comparison is made between them and the predetermined structure query patterns. The performance of the suggested technique has been evaluated by examining on weak web applications. The presented method can be performed on both web applications and applications which are linked to database

In [10], researchers have focused on one issue, namely the integrity of web content. It has been shown that given the limitations of SSL, a loss of web content integrity is possible because of the statelessness of HTTP. In an attempt to overcome this problem, a systematic web security framework was formulated to provide continued reliable and correct services to external users, even though a web data manipulation problem may have occurred. It was suggested that such a framework will offer an increased level of user confidence, since the framework provide a greater protection against web server subversion.

Novel approach for detecting SQL attacks which are based on information theory, was proposed by [11], the entropy of all queries, which exists in a program accessed before deploying a program, is computed during time of executing the program. This approach depends on the thought that dynamic queries with attack inputs result in a level of entropy that is decreased or increased. Three open source PHP applications which contain SQLI weaknesses, proved by report, validated the proposed framework. A prototype tool is implemented by them in Java for facilitating training and detection phase of the proposed technique. The result of the evaluation indicated that the technique checks all known SQLI weaknesses and might be an integral one to verify unknown weaknesses.

III. MATERIALS AND METHODS

The following figure (Fig.1) shows the main components of restoring the website.

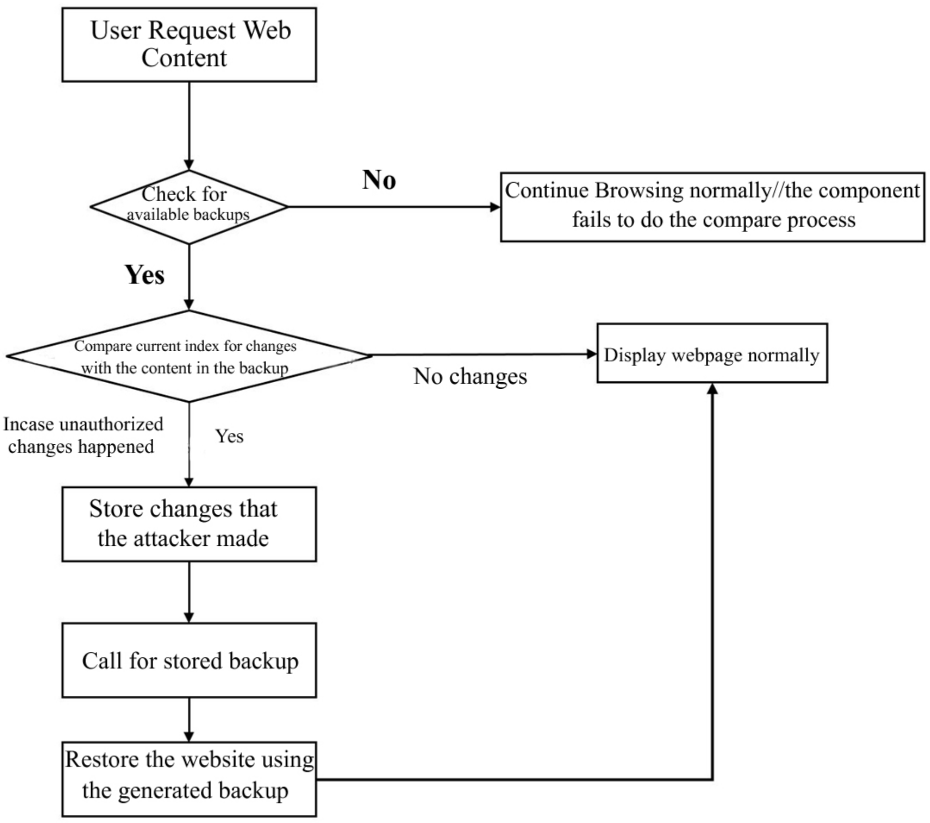

User Request Web Content
Check for available backups
No
Continue Browsing normally//the component fails to do the compare process
Yes
Compare current index for changes with the content in the backup
No changes
Display webpage normally
Incase unauthorized changes happened
Yes
Store changes that the attacker made
Call for stored backup
Restore the website using the generated backup

FIGURE 1: WEBSITE RESTORE COMPONENT

The proposed enhancement is triggered whenever a request is sent to the website, the enhanced component will check for available backups then start the process of comparison using a unique given hash for both the backup file and the website content [12], in-order to validate the current content and take actions in-case of any change on the website that wasn't permitted by the administrators [13]. The auto restore component will make sure to store all the changes that the attacker may have made after the infiltration also will make sure that the user gets the original website content and never endures the content that the attacker may have published on the server. this enhancement aims to protect the web application information and it also increases the level of integrity between the user and the web application owners and institutions, it also save the website from any misuse by the attacker if he/she aims to publish illegal, harmful or not suitable content through the hacked website, the proposed enhancement is a component that assures the integrity and safety of the content and only display the content that was only meant to be shown by the website's administrators.

When a change in the website's content is discovered by the component the process of validation will start checking for previous backups of the website on the server in order to restore the website content automatically, the enhanced component will use the last updated backup that has been generated previously to be restored.

A mechanism for generating auto backup has been added to the proposed component, which will be enabled by the administrator to create a backup for all the contents of the website at any time.

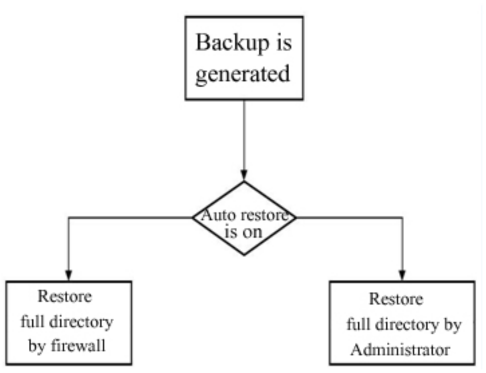

Backup is generated
Auto restore is on
Restore full directory by firewall
Restore full directory by Administrator

FIGURE 2: BACK UP IS GENERATED

3.1 Website restore component options

Two options for the proposed component has been added, which are the manual and automatic website restore. The manual restore option is left for the admin to decide to restore the changes or take whatever suitable actions in the case of infiltration. On the other hand the auto restore option will directly take automatic actions and fully restore the original website contents using the already stored backup files as explained in the fig. 2 above.

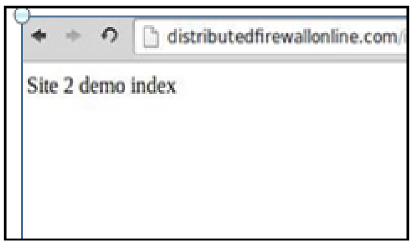
3.2 Store the changes that the attacker made

The added component has a feature to record and store all the changes that the attacker made to the web content, and make a report of these changes, this is done by comparing the code of the original web content with the content that the attacker have changed.

IV. RESULTS OF TESTING THE WEBSITE RESTORE COMPONENT

4.1 Testing restore component

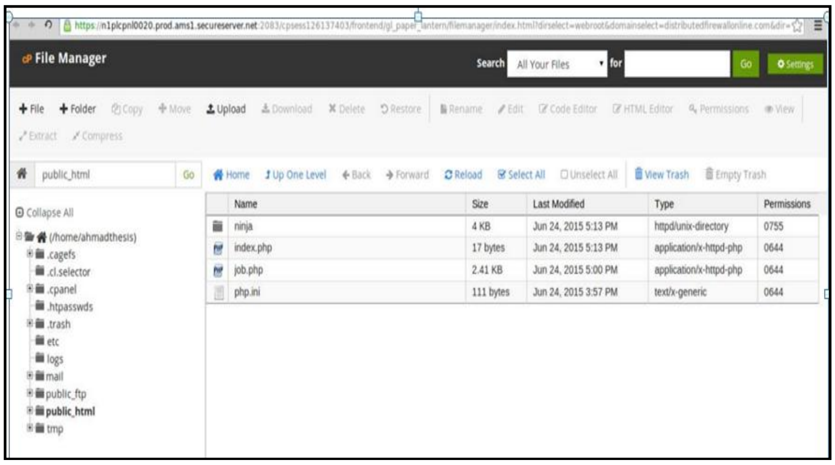
To test the restore component we have created a sample website for testing purposes consist of few web pages and an index page that is shown in fig.3 in one domain:



Site 2 demo index

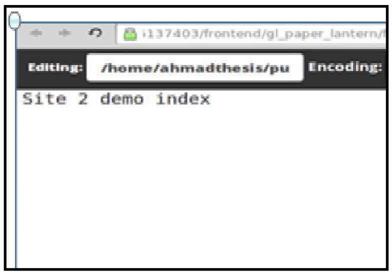
FIGURE 3: ORIGINAL SAMPLE INDEX PAGE OF THE DEMO WEBSITE

The option is set to auto restore in-order to restore the full directory in-case of change automatically see fig. 4,then a new index page is uploaded using the server hosting panel, as soon as the new index page is replaced the auto restore component gets triggered and restore the original full directory from the backup file that we have generated before see fig.5.


https://n1plcpnl0020.prod.ams1.secureserver.net:2083/cpsess126137403/frontend/gl_paper_lantern/filemanager/index.html?dirselect=webroot&domainselect=distributedfirewallonline.com&dir=
File Manager
Search All Your Files for Go Settings
File Folder Copy Move Upload Download Delete Restore Rename Edit Code Editor HTML Editor Permissions View Extract Compress
public_html Go
Home Up One Level Back Forward Reload Select All Unselect All View Trash Empty Trash
Collapse All
(/home/ahmadthesis)
.cagefs
.cl.selector
.cpanel
.htpasswds
.trash
etc
logs
mail
public_ftp
public_html
tmp

Name	Size	Last Modified	Type	Permissions
ninja	4 KB	Jun 24, 2015 5:13 PM	httpd/unix-directory	0755
index.php	17 bytes	Jun 24, 2015 5:13 PM	application/x-httpd-php	0644
job.php	2.41 KB	Jun 24, 2015 5:00 PM	application/x-httpd-php	0644
php.ini	111 bytes	Jun 24, 2015 3:57 PM	text/x-generic	0644

FIGURE 4: THE PROCESS OF CHANGING THE CONTENT FROM THE SERVER PANEL

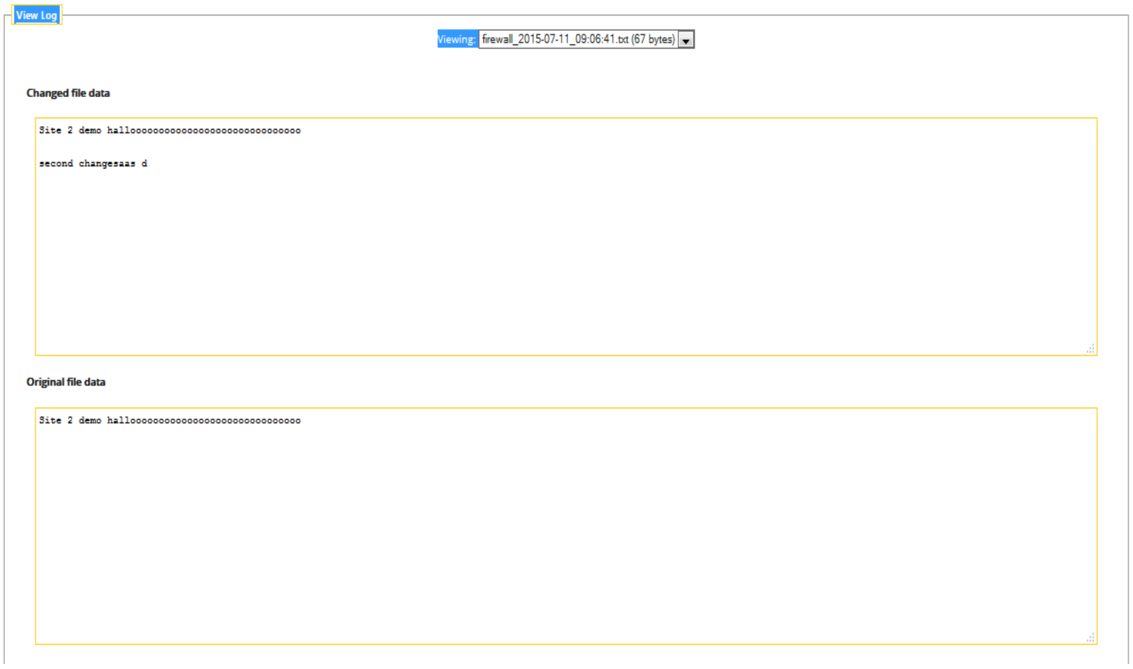

i137403/frontend/gl_paper_lantern/
Editing: /home/ahmadthesis/pu
Encoding:

```
Site 2 demo index
```

FIGURE 5: THE RESTORED WEB PAGE

4.2 Testing the feature of Providing a report for the changes that the attacker have made

The enhanced component provides a feature for storing the changes that the attacker has made to the web application a GUI of the report section with the changes that has been done during the testing attack. It shows the changes that have been done. The administrator can view both the original and the changes by clicking the log option as shown in fig. 6.



Changed file data

```
Site 2 demo hallooooooooooooooooooooooooooooo

second changesaas d
```

Original file data

```
Site 2 demo hallooooooooooooooooooooooooooooo
```

FIGURE 6: GUI THAT SHOW THE ORIGINAL AND THE CHANGES THAT THE ATTACKER MADE

V. CONCLUSION

This research presented a component that aims to enhance the performance of web application firewalls and reduce the possibility of causing damage after the attack occurrence.

The enhanced component and phases are explained to show the process of interaction between the administrator and the installed component.

This research has made the following contributions:

- Provided a solution to strengthen the integrity between the users and the web-application owners.
- Prevent major content losses inside the web application after the infiltration.
- The proposed component can produce a report for the changes that the attacker made in-side the web application.
- The proposed component provides an easy and flexible solution with an easy to use GUI.

REFERENCES

[1] Lodha S, Dhande S.**Web Database Security Techniques.**International Journal of Advance Research in Computer Science and Management Studies. 2014; 2 (3): 300-305.

[2] Dodge R. **Information Assurance and Security in the ACM/IEEE CS2013.** Ronald C. Dodge; Lynn Futcher. 8th World Conference on Information Security Education (WISE), Auckland, New Zealand. Springer, IFIP Advances in Information and Communication Technology, AICT-406, 2017, Feb, pp.48-57.

[3] J. Gupta A, Kaur K. **Vulnerability Assessmentand Penetration Testing.** International Journal of Engineering Trends and Technology. 2013; 4 (3): 328-333.

[4] Subramanian T.K, Deepa B. **Security Attack Issues And Mitigation Techniques In Cloud Computing Environments**. International Journal of UbiComp (IJU), 2016; 7 (1): DOI:10.5121/iju.2016.7101.

[5] Reddy D. et al. **Detecting Attacks and Protecting From single To Multi Level application.** International Journal of Advanced Technology in Engineering and Science. 2015; 3 (1): 478 – 484.

[6] Laranjeiro N, et al. **A Learning-Based Approach to Secure Web Services from SQL/XPath Injection Attacks.** Pacific Rim International Symposium on Dependable Computing IEEE. 2010; 10 (24): 191-198.

[7] Hidhaya S, Angelina G. **Intrusion Protection against SQL Injection Attacks Using a Reverse Proxy.** Recent Trends in Computer Networks and Distributed Systems Security Communications in Computer and Information Science. 2012; 335: 252-263.

[8] Bangre S, Jaiswal A, et al. **SQL Injection Detection and Prevention Using Input Filter Technique**. International Journal of Recent Technology and Engineering (IJRTE). 2012; 1 (2): 145-150.

[9] Win W, Htun H. **Detection of SQL Injection Attacks by Combining Static Analysis and Runtime Validation.** International Conference on Advances in Engineering and Technology. 2014; 3 (20): 95-99.

[10] Aljawarneh S, Laing C, Paul Vickers. **Verification of Web Content Integrity: A new approach to protecting servers against tampering**. School of Computing, Engineering & Information Sciences North Umbria University, Newcastle upon Tyne.2013; 1-6.

[11] Shahriar H, North S, Chen W. **Early Detection Of sql Injection Attacks**. International Journal of Network Security & Its Applications. 2013; 5 (4): 53 -65.

[12] Ogini N, Ogwara N. **Securing Database passwords using a combination of hashing and salting techniques**. IPASJ International Journal of Computer Science (IIJCS). 2014; 2 (8): 52-58.

[13] Rewatkar L, Lanjewar U. **Implementation of Cloud Computing on Web Application. International Journal of Computer Applications**. 2010; 2 (8): 28-32.