Application Firewall System

Rohan Kelkar

B.E. Computer Engineering, K.J.Somaiya College of Engineering, Mumbai.

Abstract

The main idea behind implementing such a project is to develop a firewall system which is a network security system that doesn't permit the user to make any modifications on the application over the network. A firewall typically establishes a barrier between a trusted, secure internal network and another outside network, thereby protecting the system from malicious attacks that could cause damage or loss of data confidentiality. It will thereby prevent the system from attackers using several mechanisms to get access to the data stored in the system.

Keywords: firewall, network security, malicious attacks, data confidentiality.

I. INTRODUCTION

The developed firewall will detect and identify the attacks and block the attacks from affecting the application over the network and will make our system protected from such external threats.

Our system detects the following types of attacks:

- URL Attack
- Union SQL Injection Attack
- Santy worm introduction Attack
- XSS-Redirection Attack

It maintains a log of the attack in the log text installed over the system and determines exactly which type of attack has occurred.

II. SOFTWARE ARCHITECTURE THE SYSTEM

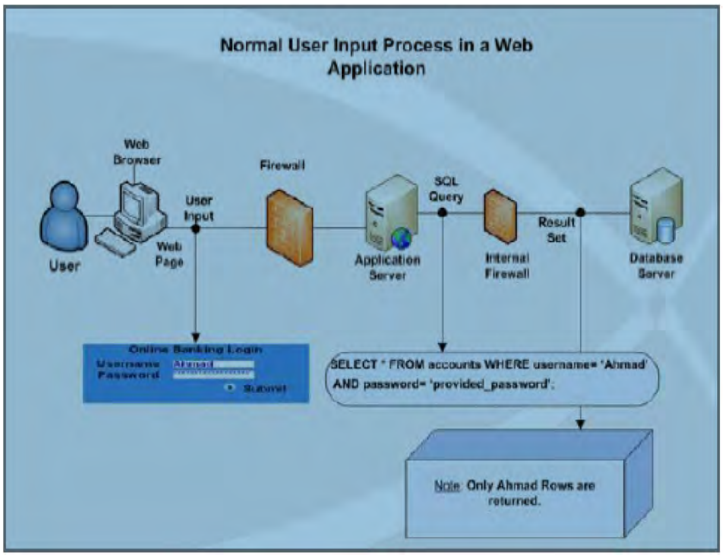
a. System architecture diagram

Attacks are simple in nature e.g. an attacker passes string input to an application in hopes manipulating the SQL statement to his or her advantage. The complexity of the attack involves exploiting a SQL statement that may be unknown to the attacker.

Open source applications and commercial applications delivered with source code are more vulnerable since an attacker can find potentially vulnerable statements prior to an attack.

The System architecture of our project is such that it includes a firewall in between thereby not allowing the intruder to not make any intended changes and thus prevent the system from being exploited.

Thus, the firewall provides as a security check protecting the application from not being modified, thus its database when put over a network.


Normal User Input Process in a Web Application
Web Browser
Firewall
SQL Query
User
User Input
Web Page
Application Server
Internal Firewall
Result Set
Database Server
Online Banking Login
Username
Ahmad
Password
Submit
SELECT * FROM accounts WHERE username= 'Ahmad' AND password= 'provided_password';
Note: Only Ahmad Rows are returned.

III. REQUIREMENTS

A. Software Requirements

- PHP
- XAMPP.

B. Hardware Requirements

Implementing the firewall just requires a computer having Windows OS. However, when the application is embedded into the application put on the real time network, this system needs to be included in its server side.

C. Functional Requirements

Attacks are a serious threat to any Web application that receives input from users and incorporates it into SQL queries to an underlying database.

Most Web applications used on the Internet, i.e. over the network or within enterprise systems work this way and could therefore be vulnerable to attack

Firewall will be able to detect and prevent system against several types of attacks such as:

- **URL manipulation Attack -**

  Manipulating URL to get unauthorized access to web pages.

  Union/SQL Injection Attack -

  Manipulating the query fired to get results as per choice or Altering or modifying the database content

- **XSS Attack -**

  Inject client-side script into web pages viewed by other users.

  This leads to direction of the certain web pages with certain format to another URL

- **Santy Worm Attack -**

  Inject client-side script into web pages viewed by other users.

D. Non –Functional Requirement

It should be able to perform well and protect the system against mentioned attacks and thereby provide secured environment.

- **Performance Requirements**

  It is expected that the program to defend an attack as per the rules appended in the firewall. It should detect attack as soon as possible.

- **Safety Requirements**

  The firewall should not alter any system code. It should not change.

- **Availability**

  The availability of the system is easy.

- **Correctness**

  The results of the function are pure and accurate.

- **Flexibility**

  The operation may be flexible and one can edit or add more rule to the firewall.

- **Maintainability**

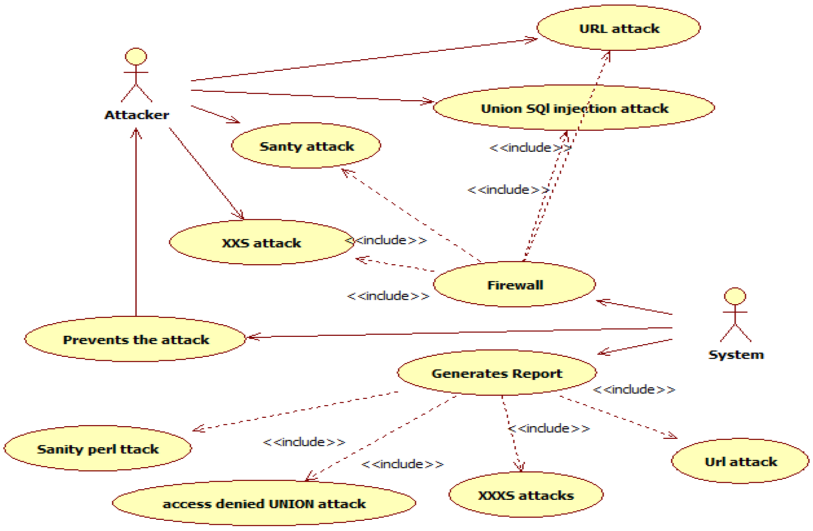
  After the deployment of the system if any error occurs then it can be easily maintained and corrected.

- **Timeliness**

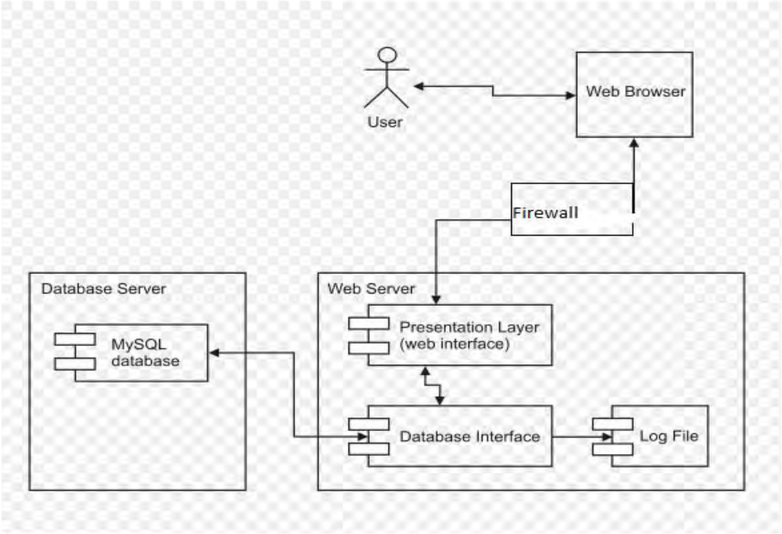
  The time limit is very important. It will save much time and provide fast detection of the attacks occurred.

IV. SOFTWARE DESIGN

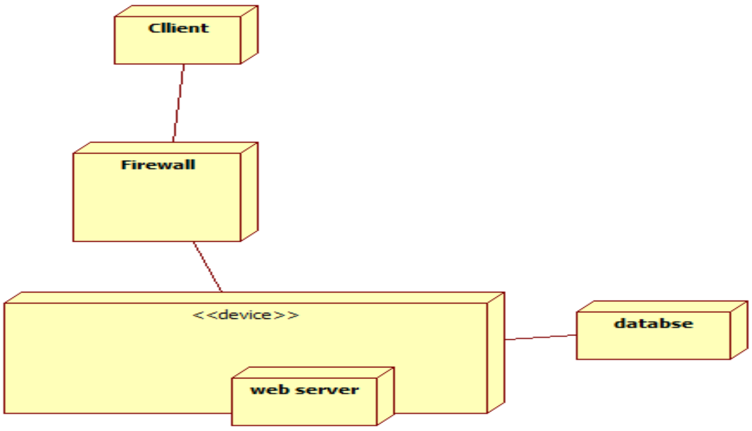
a. Use case diagram



b. Component Diagram

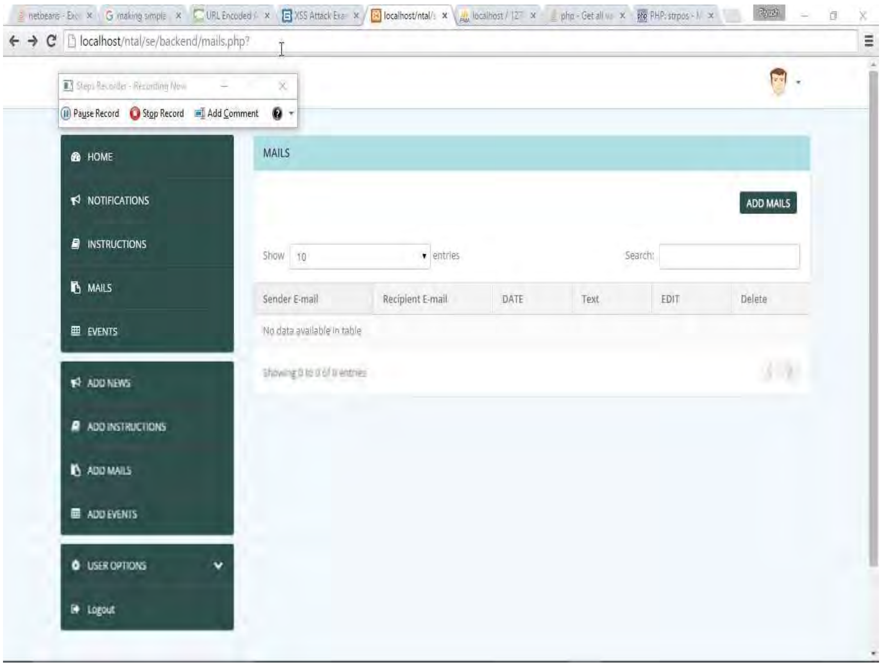


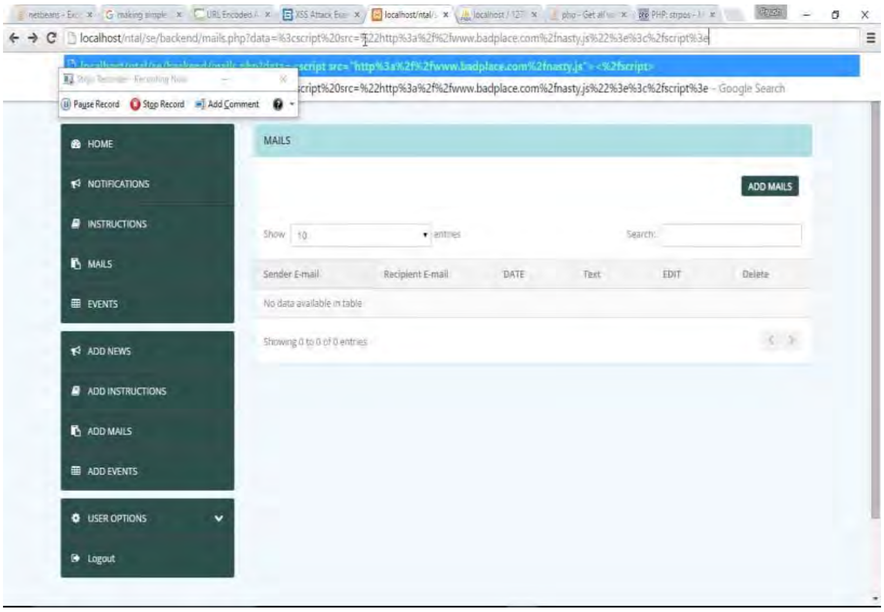
c. Deployment Diagram



V. IMPLEMENTATION DETAILS SCREENSHOTS OF IMPLEMENTATION

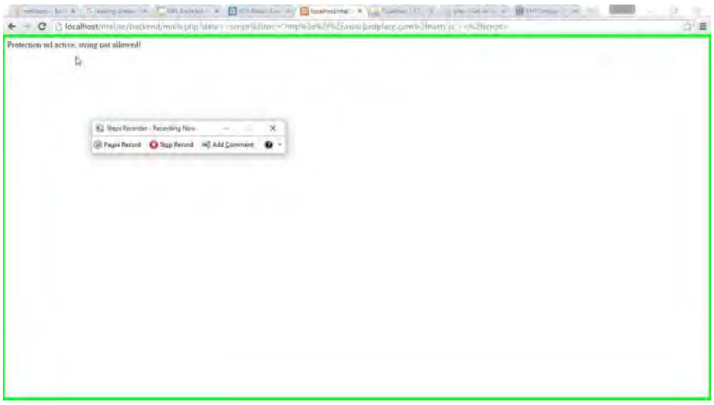
a. URL Attacks:



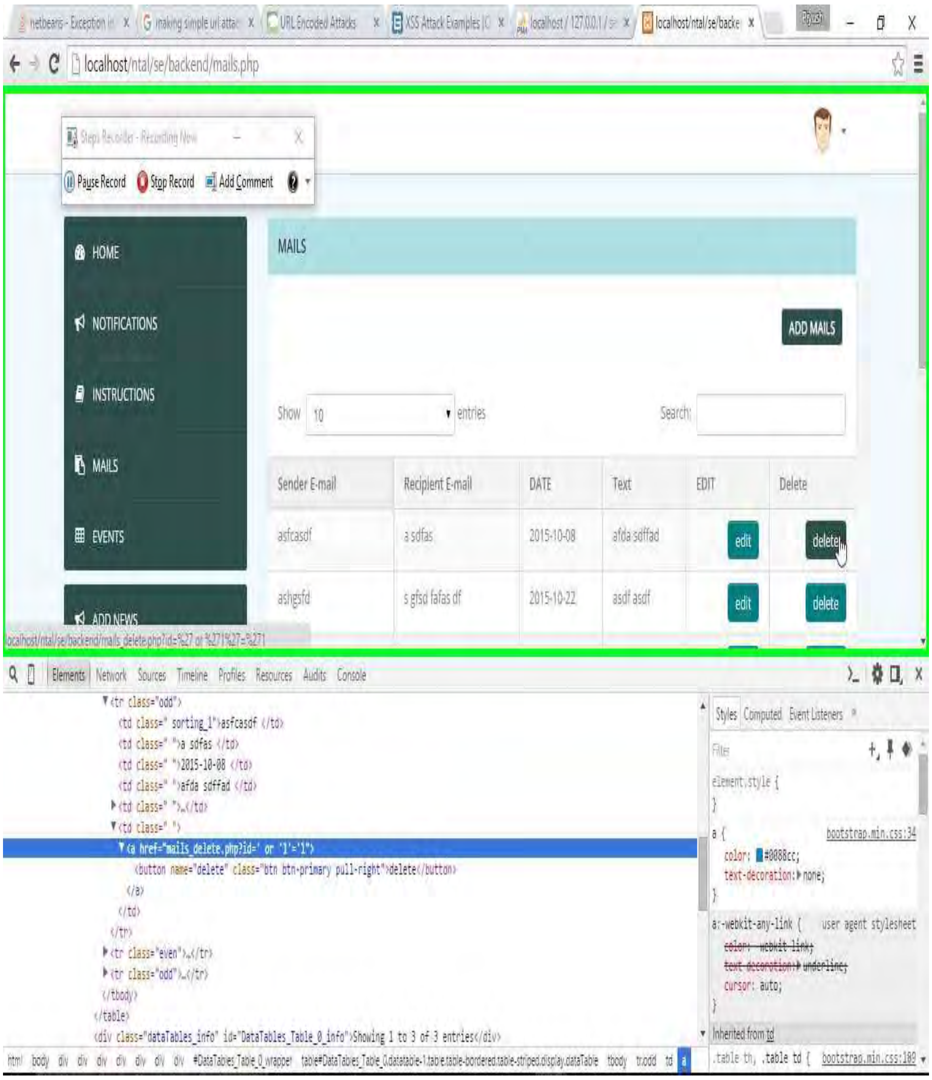

localhost/ntal/se/backend/mails.php?data=%3cscript%20src=%22http%3a%2f%2fwww.badplace.com%2fnasty.js%22%3e%3c%2fscript%3e

script src="http%3a%2f%2fwww.badplace.com%2fnasty.js"><%2fscript>

script%20src=%22http%3a%2f%2fwww.badplace.com%2fnasty.js%22%3e%3c%2fscript%3e - Google Search


Protection url active, string not allowed!
localhost/ntal/se/backend/mails.php?data=<script%20src="http%3a%2f%2fwww.badplace.com%2fnasty.js"><%2fscript>
Steps Recorder - Recording Now
Pause Record
Stop Record
Add Comment

UNION SQL Injection Attacks:



localhost/ntal/se/backend/mails.php

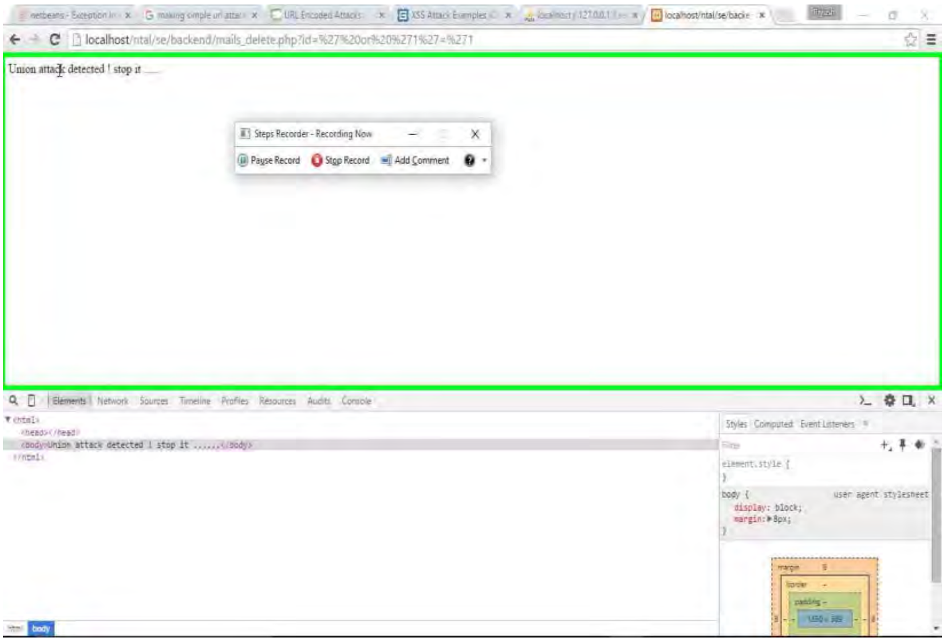
MAILS

Show 10 entries Search:

Sender E-mail	Recipient E-mail	DATE	Text	EDIT	Delete
asfcasdf	a sdfas	2015-10-08	afda sdffad	edit	delete
ashgsfd	s gfsd fafas df	2015-10-22	asdf asdf	edit	delete

localhost/ntal/se/backend/mails_delete.php?id=%27 or %271%27=%271

```
▼<tr class="odd">
   <td class=" sorting_1">asfcasdf </td>
   <td class=" ">a sdfas </td>
   <td class=" ">2015-10-08 </td>
   <td class=" ">afda sdffad </td>
  ▶<td class=" ">…</td>
  ▼<td class=" ">
   ▼<a href="mails_delete.php?id=' or '1'='1">
      <button name="delete" class="btn btn-primary pull-right">delete</button>
    </a>
   </td>
 </tr>
▶<tr class="even">…</tr>
▶<tr class="odd">…</tr>
</tbody>
</table>
<div class="dataTables_info" id="DataTables_Table_0_info">Showing 1 to 3 of 3 entries</div>
```

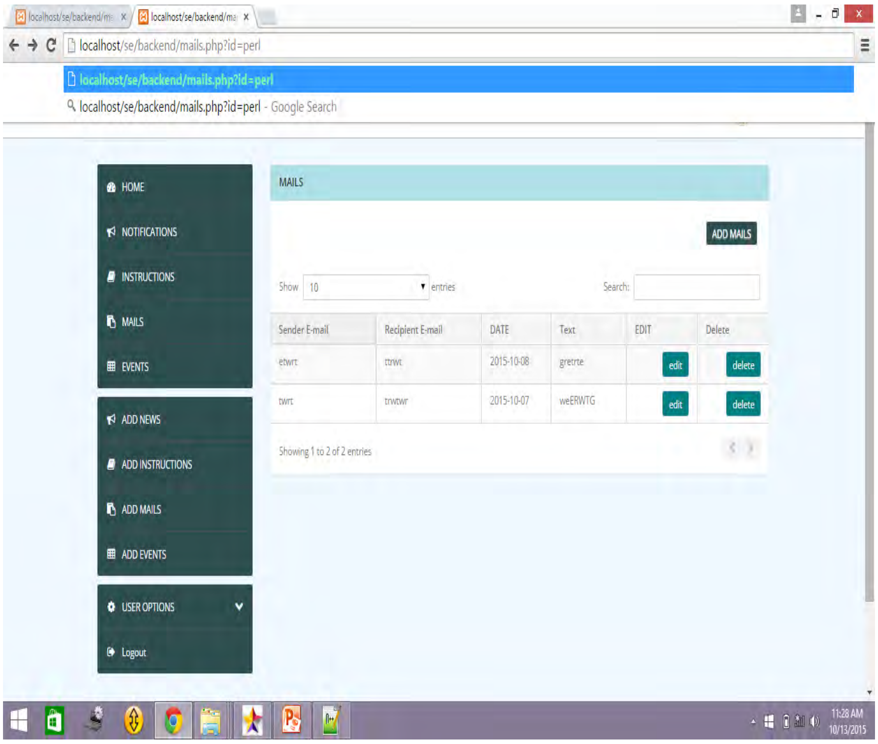


localhost/ntal/se/backend/mails_delete.php?id=%27%20or%20%271%27=%271

Union attack detected ! stop it ....

```
▼<html>
   <head></head>
   <body>Union attack detected ! stop it .....</body>
 </html>
```

Santy Worm Attacks:



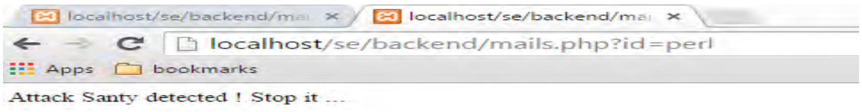
localhost/se/backend/mails.php?id=perl

localhost/se/backend/mails.php?id=perl - Google Search

MAILS

Sender E-mail	Recipient E-mail	DATE	Text	EDIT	Delete
etwrt	ttrwt	2015-10-08	gretrte	edit	delete
twrt	trwtwr	2015-10-07	weERWTG	edit	delete

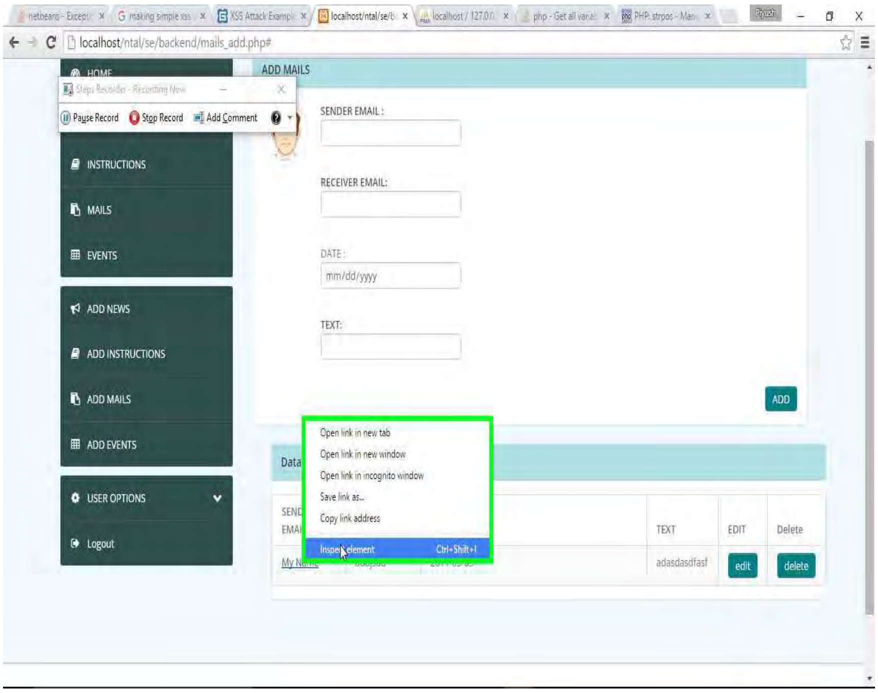
Showing 1 to 2 of 2 entries



localhost/se/backend/mails.php?id=perl

Attack Santy detected ! Stop it ...

XSS Attacks:



localhost/ntal/se/backend/mails_add.php#

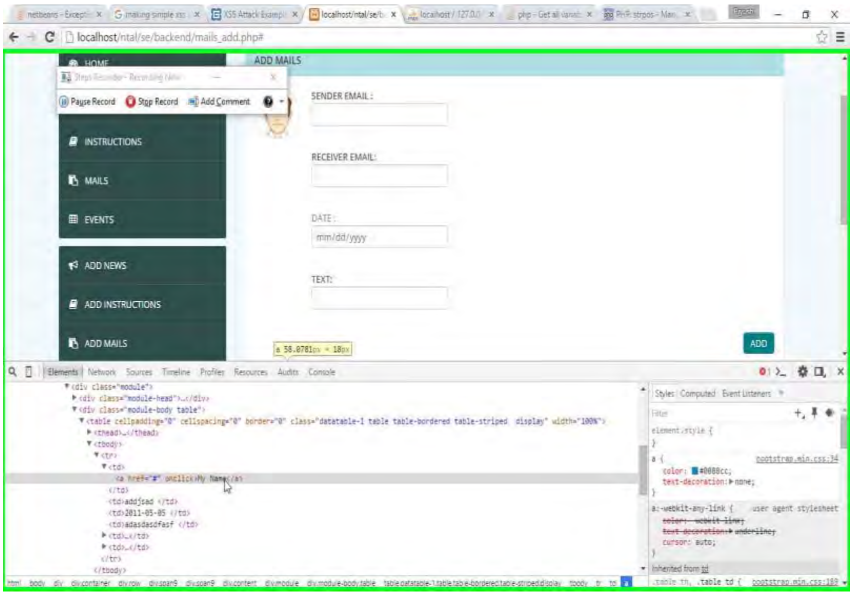
ADD MAILS

SENDER EMAIL :

RECEIVER EMAIL:

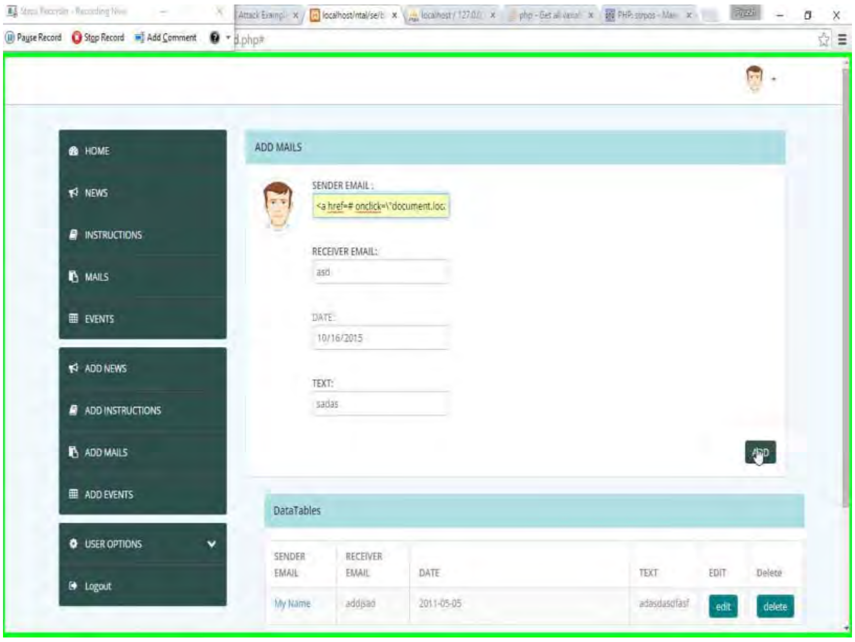
DATE : mm/dd/yyyy

TEXT:



```
▼<div class="module">
  ▶<div class="module-head">…</div>
  ▼<div class="module-body table">
    ▼<table cellpadding="0" cellspacing="0" border="0" class="datatable-1 table table-bordered table-striped  display" width="100%">
      ▶<thead>…</thead>
      ▼<tbody>
        ▼<tr>
          ▼<td>
              <a href="#" onclick>My Name</a>
            </td>
            <td>addjsad </td>
            <td>2011-05-05 </td>
            <td>adasdasdfasf </td>
          ▶<td>…</td>
          ▶<td>…</td>
          </tr>
        </tbody>
```

DATE : mm/dd/yyyy



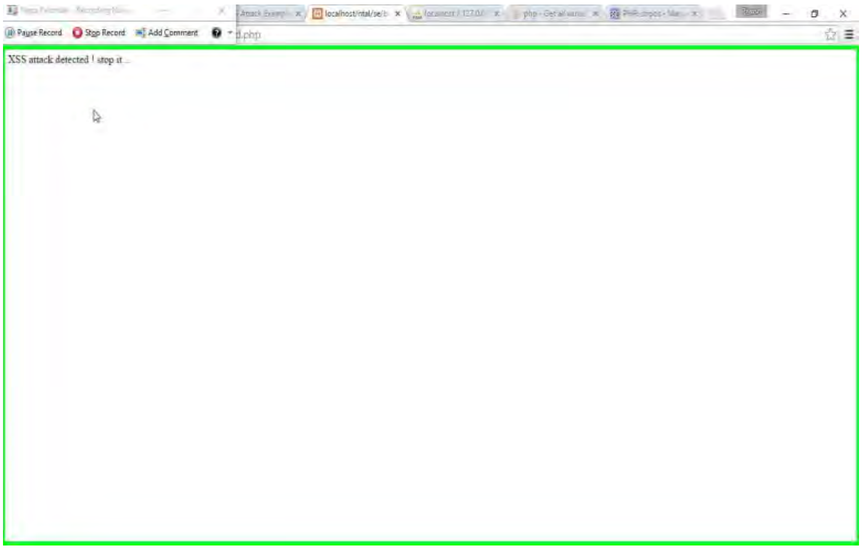
SENDER EMAIL : <a href=# onclick=\"document.loc

RECEIVER EMAIL: asd

DATE: 10/16/2015

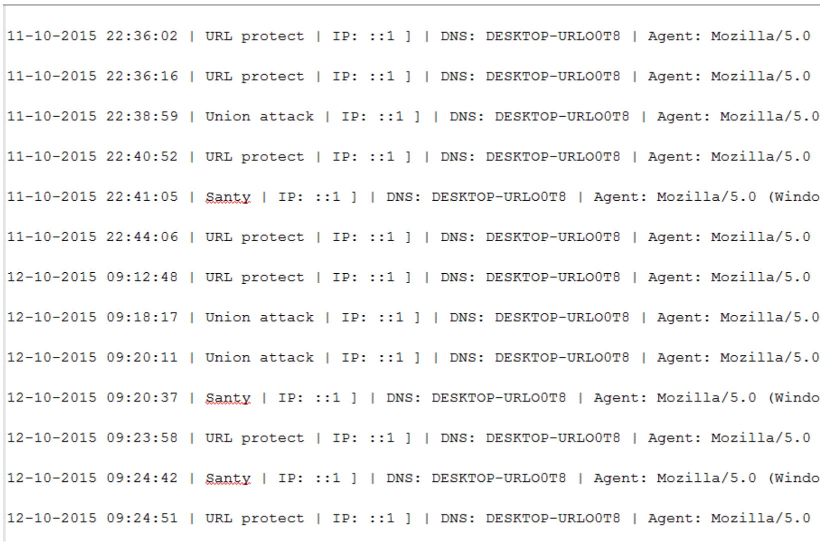
TEXT: sadas

SENDER EMAIL	RECEIVER EMAIL	DATE	TEXT	EDIT	Delete
My Name	addjsad	2011-05-05	adasdasdfasf	edit	delete



XSS attack detected ! stop it ..

LOG FILE:



```
11-10-2015 22:36:02 | URL protect | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0

11-10-2015 22:36:16 | URL protect | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0

11-10-2015 22:38:59 | Union attack | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0

11-10-2015 22:40:52 | URL protect | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0

11-10-2015 22:41:05 | Santy | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0 (Windo

11-10-2015 22:44:06 | URL protect | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0

12-10-2015 09:12:48 | URL protect | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0

12-10-2015 09:18:17 | Union attack | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0

12-10-2015 09:20:11 | Union attack | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0

12-10-2015 09:20:37 | Santy | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0 (Windo

12-10-2015 09:23:58 | URL protect | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0

12-10-2015 09:24:42 | Santy | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0 (Windo

12-10-2015 09:24:51 | URL protect | IP: ::1 ] | DNS: DESKTOP-URLO0T8 | Agent: Mozilla/5.0
```

VI. ADVANTAGES

1. The firewall prevents attacks like URL Attack, SQL Injection Attack, Santy Worm Attack and XSS Attack and also records the type and time of attack in the log.
2. It can inspect the contents of traffic, blocking specified content, such as certain websites, viruses, or attempts to exploit known logical flaws in client software.
3. They can permit or deny specific applications or specific features of an application given a great degree of granular control.
4. Application firewalls can also authenticate users directly. This means, for example, that they can allow or deny a specific incoming telnet command from a particular user, whereas other firewalls can only control general incoming requests from a particular host.

VII.FUTURE SCOPE

The application firewall system has a lot of advantages. However, there is a lot of room for improvement too. The disadvantages of the above system that can be improved by the systems developed in the future are as follows:

1. The main drawbacks to Web application firewalls are cost and performance.
2. Performance is often an issue because these tools inspect all incoming and outgoing traffic at the application layer.
3. This level of examination, often referred to as deep packet inspection, examines the actual payload of a packet and provides far better content-filtering capabilities than traditional packet-filtering firewalls.

VIII. CONCLUSION

Thus, a secure system was achieved against the considered attacks by implementing a firewall system, thereby preventing loss of data confidentiality.

P

IX. ACKNOWLEDGEMENT

I would like to thank the other members in my project group Mr. Piyush Vishwakarma, Mr. Sumeet Jain, Ms. Aishwarya Walimbe and Mr. Chintan Thakkar for their hard work and timely efforts so that the project implementation could be executed in an accurate and timely manner. I would also like to thank my parents who constantly encouraged and supported me in my endeavour.

X. REFERENCES

[1] www.stackoverflow.com
[2] www.wikipedia.com
[3] www.technet/microsoft-en.aspx
[4] www.geekstuff.com
[5] www.geeksforgeeks.com