IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727,
PP 31-35
www.iosrjournals.org

Web Defender

Mr. Shivam Singh[1], Mr. Dineshkumar Tanwar[2], Mr. Aman Singh[3],
Mr. Ashish Yadav[4], Mrs. Vaishali Nirgude[5]

[1]*(Department of Computer Engineering, Thakur College of Engineering & Tech. / Mumbai University, Student)*
[2]*(Department of Computer Engineering, Thakur College of Engineering & Tech. / Mumbai University, Student)*
[3]*(Department of Computer Engineering, Thakur College of Engineering & Tech. / Mumbai University, Student)*
[4]*(Department of Electronics & Telecommunication Engineering, Thakur College of Engineering & Tech. / Mumbai University, Student)*
[5]*(Department of Computer Engineering, Thakur College of Engineering & Tech. / Mumbai University, Assistant Professor)*

Abstract : *The project is all about Web Application security. The main aim of this project is to make your Web application more difficult to hack. Several vendors opt for Cloud based website hosting nowadays, as it is very trustworthy, easily scalable and reasonable hosting solution than traditional website hosting in various aspects. Many people wrongly assumed that hiring a smart developer or deploying commercial security product will magically make their site hacker proof. So, we will be implementing a virtual image of system implemented by top security features. The expected outcome of this Project will be Self-Configurable Virtual Machine Image which will be used by an enterprise to protect its own application by putting VM image in front of Application. Enterprise needs to implement an application which can demonstrate correct functionality. Non Functional Requirement of the application like Scalability and Security will be handled by Self Configurable Virtual Machine. Here VM will protect Enterprise-wide application through the various Hierarchical mechanism. VMs will allow Access level Protection along with Predictable Attack on Application.*

Keywords: *About Distributed Denial of service (DDOS), Domain Name System (DNS), Hypertext Transfer Protocol (HTTP), Kernel Virtual Machine (KVM), Open Web Application Security Project (OWASP), Reverse proxy, Virtual Machine (VM).*

I. Introduction

The Project will behave as Self Configurable Virtual Machine which can be used by an enterprise to protect its own application by putting VM image in front of Application. Enterprise need to develop application which can demonstrate correct functionality. Non Functional Requirement of the application like Scalability and Security will be handle by Self Configurable Virtual Machine. Here VM will protect Enterprise wide application though various Hierarchical mechanism. VMs will allow Access level Protection along with Predictability of Attack on Application. Now a day's internet is an insecure place and attacks keep occurring. One of the main reasons is the poor quality of the software used in systems and application software. So, we are coming up with the innovative idea for making web apps secure from hackers and other attackers. We are making a product which will be act as intermediate between the client web app and users (legitimated users / hackers). This product will fully have secured and configured by top 10 OWASP core rules. Now a day's internet is an insecure place and attacks keep occurring. One of the main reasons is the poor quality of the software used in systems and application software. So, we are coming up with the innovative idea for making web apps secure from hackers and other attackers. We are making a product which will be act as intermediate between the client web app and users (legitimated users / hackers). This product will fully have secured and configured by top 10 OWASP core rules. The end product will have the ability to secure any Web Application of client. Product will be flexible as per the requirement of virtual machine. Charges will be standard as of security services.

II. Literature Survey

The existing system include direct access of client and server, where in people are just relying on where they are hosting their website, or a platform where they are deploying their website which is completely dependent on the users concern.

The problems with the existing system are:
a) The existing system involves many types of attack.
b) Most of the attacks on enterprises networks are shifted towards cloud based application as more and more.
c) In the current system there are less security feature.

d) Very few existing systems provide a website security.
How did the idea originated?

- Security land scape changed from networking to web site.
- Current e-business and e-commerce area, security of web site is a major issue.
- Current process is about need a Cost effective solution
- So ideas is not to worry about changing the code to mitigate vulnerability instead create a cloud based services which will act as security as a service model.

What is specific problem and gapes we have identified:
Programmers frequently rely too much on frameworks to defend against dangerous inputs, or use application firewalls based on signatures that work by blacklisting the various attack vectors published by hackers in cross-site scripting (XSS) or SQL injection cheat sheets etc.
In today's world, where majority of people use website applications for their day to day needs, the application for advance marketing will be a success.

III. Proposed Work

The application overcomes the problems in the current system in the following way:
Here below the whole architectural model is shown in fig.1
a) The idea is to create a fully equipped Open source based Virtual Machine, which will behave as A Reverse proxy for all web applications with innovative modules [3][4].
b) Hide the actual location of Web application: Unauthorized user try to access the website directly from the web server (actual location). So we are making use of reverse proxy server. A reverse proxy machine will take request from internet and will be forwarded to servers in internal network. Those who making requests to the proxy server may not be aware of internal network. You can see the references for this idea from [4].
c) Profiling Good access to Web application and create a Sets of permissible User inputs to web applications: Profiling (dynamic program or log analysis that measures the space (memory) or complexity of frequency and duration of function calls) of good access to web application and creating sets of permissible user inputs to web application [6][9].
d) Implementing standards OWASP Mod Security Core Rules Set [7].
This OWASP Mod security will have service feature as set of all generic attack detection rules that will be better as base level. The OWASP rules will be applied if following condition occur:

- Http Protocol Protection
- Real Time Black List Lookups

Architecture of system

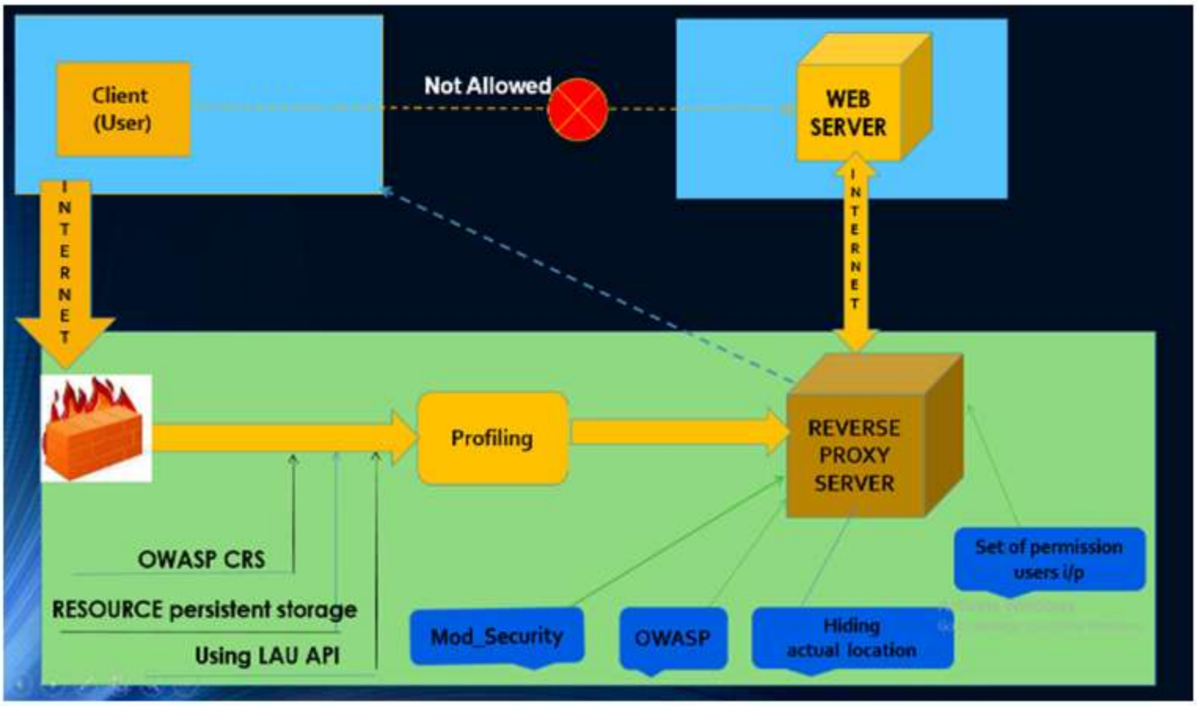


Fig.1 : Architectural model

IV. Methodology and Design/Algorithm

a) In this project, we as a team have thought of to proceed with Spiral model. The reason to choose this specific model is due to its changing system requirements and measurable progress of system.

b) Spiral model will follow a circular trail where once decision is made can change depending upon the requirement. The circular structure of spiral model is very favorable for our condition where client-interference will be at minimum and final assessment followed by presentation will be conducted at the end.

Spiral model in our project:

As our project is never ending process. So we have chosen the spiral model for product development. The various phases for spriral model is describe below:

Communication: We communicated to many industrial people, startups and found out some of the security services gaps.

Planning: After doing the literature survey, we came up with our unique idea. Accordingly we have planned the complete project implementation.

Modeling: Our complete project model is divided into three phases which are describe as below

1-Protecting DOS attacks.

2-Implementing the OWASP Top 10 security vulnerabilities.

3-Real-time Profiling.

Construction: We implementated whole project on AWS (Amazon web service) and on linux environment.

We have used shell script as a language for developing varoius module of our system. We created a web interface for client to get our best security services for his/her websites.

Deployment: After implementation we have deployed our project on Amazon web services.

Pseudo Code Design:

Web Defender implements four module:

- Detect denial service of attack
- Hide the actual location of web server
- OWASP modsecurity
- Set of permissible user inputs

In first phase of our project we have successfully implemented first module (Detect denial service of attack. Currently we are working on second module.)

Detect denial service of attack:

The mod evasive apache module sometimes called as mod DoS evasive, it protects against Denial of service attack, Distributed DoS and brute force attack on the web server. It can also protect during attack and report to be abuses via mail and syslog facilities. We have implemented mod evasive in our module, which will detect and deny the blacklisted ip address. If following condition occur:

- Requesting the same page multiple times within some time frame.
- Making more than 50 concurrent requests on the same child per second
- Will be making any requests while temporarily blacklisted.

If any one of the above criteria match, a 403 type of response is sent and the that particular IP address will be blocked. Optionally, a notification will be sent to owner of the server or that particular ip are blocked.

V. Implementation

Here we are showing small implementation of our product, you can see the result of implementation.

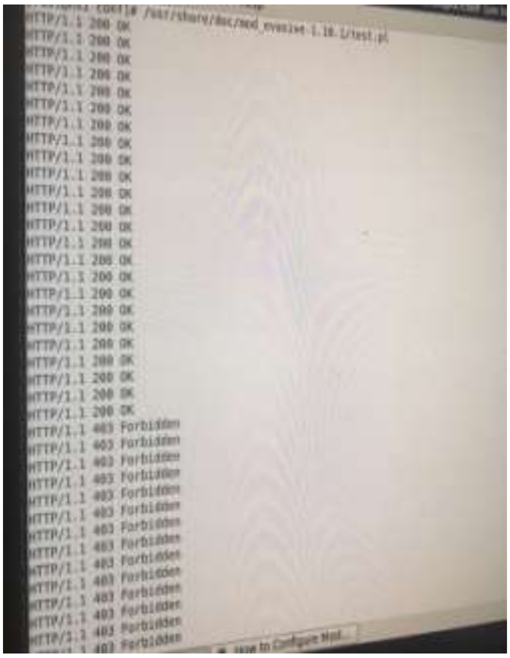
Algorithm (Perl Script for DOS attack):

Below is given code for DoS attacks and the code is taken from web site as mention [8].

```
#!/usr/bin/perl
# test.pl: small script to test mod_dosevasive's effectiveness
use IO::Socket;
use strict;
for(0..100)
{
  my($responses);
 my($SOCKETS)= new IO::Socket::INET(Proto   => "tcp",
                   PeerAddr=> "target.com:80");
 if (! defined $SOCKETS)
             {
        Die $!
       }
print $SOCKETS "GET /?$_ HTTP/1.0        \n\n";
$responses = <$SOCKETS>;
print $responses;
close($SOCKETS);
}
```

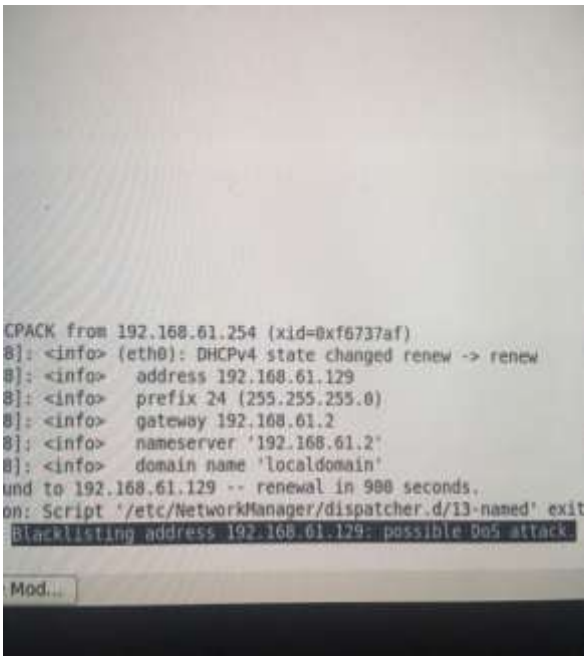
VI. Results and Discussion

Currently, we have successfully completed our module of our project i.e. As shown in below some proof of work.



```
conf]# /usr/share/doc/mod_evasive-1.10.1/test.pl
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
```

Fig.3: Result of DOS attack(a)



```
CPACK from 192.168.61.254 (xid=0xf6737af)
8]: <info> (eth0): DHCPv4 state changed renew -> renew
8]: <info>   address 192.168.61.129
8]: <info>   prefix 24 (255.255.255.0)
8]: <info>   gateway 192.168.61.2
8]: <info>   nameserver '192.168.61.2'
8]: <info>   domain name 'localdomain'
und to 192.168.61.129 -- renewal in 900 seconds.
on: Script '/etc/NetworkManager/dispatcher.d/13-named' exit
 Blacklisting address 192.168.61.129: possible DoS attack.
```

Fig.4: DOS blacklisting report(b)

In Fig.3 (a) we have shown the result of Denial of service attack. As we can see in the figure there are many http requests which has been sent by the above code. After some request it will deny accesses for particular IP. As forbidden requests have been shown in fig.(a). It means that Particular IP cannot be able to access the resources.

After denying the request that particular IP will be recorded and will be shown as a log analysis. Then that ip will not be able to access the web resources for some times.

VII. Conclusion

Our product will help to provide VM based Security as a service model. Initially VMs will be hosted on Public cloud to make easily available to various enterprises. Any enterprise even an individual whose business is running online can take a product and acquire the best services needed for his/her web application. This project will majorly attract the people those running his /her business online and want to grow his business with full boost. In initial state it may not stand with the today’s security service provider like GoDaddy and CloudFlare and many more. But it will be best product in future. In market there are many security service provider as competitor.

References

[1] Ryan c. Barnett, “Web Application Defender’s cookbook: Battling hackers and protecting users” Published by john Wiley & sons Inc., 2013.

[2] mod_evasive https://www.digitalocean.com/community/tutorials/how-to-protect-against-dos-and-ddos-with-mod_evasive-for-apache-on-centos-7 aws php documentation, https://aws.amazon.com/blogs/developer/provision-an-amazon-ec2-instance-with-php/

[3] Reverse Proxy with mod_proxy on CentOS 7, https://www.digitalocean.com/community/tutorials/how-to-use-apache-as-a-reverse-proxy-with- mod_proxy-on-centos-7 BugTraq, http://www.securityfocus.com/archive/1

[4] Auger, R., Barnett, R.: Web Application Security Consortium: Threat Classification Version 1.0. Web Application Security Consortium (2004), http://www.webappsec.org

[5] The Open Web Application Security Project (OWASP) considers, in its 2016 top ten list. Available: https: / /www.owasp.org/index.php/ToPI02016 – TOP 1O.

[6] https://www.digitalocean.com/community/tutorials/how-to-protect-against-dos-and-ddos-with-mod_evasive-for-apache-on-centos-7.

[7] Ivan Ristic: ModSecurity Handbook: The Complete Guide to the Popular Open Source Web Application Firewall, 2010 Feisty Duck Ltd Edition ISBN: 1907117024.

[8] The OWASP mod security project Available:

[9] https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project

[10] NAXSI Project (Nginx Anti Xss Sql injection) May 2014. Available: https://www.owasp.org/index.php/OWASPNAXSlProject.