International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.131.11

Web-Based Honeypot Analysis Tool

Prachi Shantaram Yewale[1], Anup Kumar Maity[2], Ravindra Sangle[3], Prasad Awere[4]
[1,2,3,4]*Department of Computer Engineering, Vidyalankar Institute Of Technology, Mumbai,India*

Abstract—In the current technological era of development and enhancements, it is very crucial to keep the system safe from known and unknown attacks. Today, there are thousands of Business, Entertainments, Media, Non-profit and the major one used by people is the E-commerce website. This kind of website stores personal information of the users like bank details, address, contact number which needs to be stored securely without any breach. It is a need to have a security system in place to deal with such kind of threats. Honeypot is a technology in which the data and the resources are placed in the network and is used to capture every detail of the user using the website, it indirectly acts as a dummy wherein if an attacker tries to gain information it will be detected by the system and thus the admin can take necessary action. This paper proposes a Web-based honey pot tool for the detection of an intrusion and to gain insights about the attacker and its types. The capabilities and limitations were tested and can be used as an intrusion detector so as to prevent malicious activities.

I. INTRODUCTION

The advantage and ease of connecting through the internet and the world wide web was accompanied by the various internet attacks and its serious hazards, a number of technologies have been implemented and are under development for prevention of such attacks and improvisation of the network security. To detect the black-hats society, it isnecessary to keep up-to-date with the hacker innovations. Inrecent times, two types of security scenario activitiesobserved namely, black-hats and white-hats. Black-hatsdestroys the network while white-hats protects the network.Honeypots were used for combat attacks. Honeypots can bedefined as “An attractive defence tool placed in a networkthat attracts the attackers towards it, detects them, andobserve them with the actual intention to know them” [1].

Attackers are now targeting the client system as they are more vulnerable and fragile as compared to the primary servers whereas in the past they targeted the main primary servers. It is very important to determine various modes of attacks, malicious code and scripts to make the system more secure. The web based client honeypot interacts with the server and websites using HTTP and FTP protocols.

This is a low interaction honeypot used in determining any malicious activity and keeping a track of it. It uses a signature based pattern matching algorithm to determine the types of attack and malicious code. The one drawback of this technique is it will not detect any unknown attack whose signature is absent in the database.

Here are several freely-available honeypot tools specialized for understanding SSH, web and malware attacks:

1. Kippo is an SSH honeypot that can log brute force attacks, where remote the remote attempts to guess logon credentials of an SSH server. Best of all, Kippo is able to record and replay the attacker’s interactions with the emulated shell on the fake SSH server.
2. Glastopf is a web application honeypot. It emulates often-exploited web vulnerabilities, such as remote and local file inclusion and SQL injection. Glastopf examines the attacker’s HTTP request and attempts to respond according to expectations to, for instance, download malicious files.
3. Dionaea is a honeypot for collecting malware. It emulates vulnerabilities in Windows services often targeted by malware, such as SMB, HTTP, TFP and FTP. Dionaea’s handling of the SMB protocol is particularly liked by researchers, as is its ability to emulate the execution of the attacker’s shell code.
4. Thug is a client-side honeypot (honey client) that emulates a web browser. It is designed to automatically interact with the malicious website to explore its exploits and malicious artifacts, often in the form of JavaScript.

II. LITERATURE SURVEY

There are a lot of honeypot tools has been around such as Project honey net, Specter, Honeyd, back officer and many more. The only reason the existence of honeypot is because the information beneath it that can help web administrator to understand any kind of attack and how to countermeasure it.

2.1 According to Lance Spitzner (2002) :

‘A honeypot as "a security resource whose value lies in being probed, attacked or compromised ". This means that whatever we designate as a honeypot, it is our expectation and goal to have the system probed, attacked, and potentially exploited. Keep in mind, honeypots are not a solution. They do not 'fix' anything. Instead, honeypots are a tool. How you use that tool is up to you and depends on what you are attempting to achieve. A honeypot may be a

system that merely emulates other systems or applications, creates a jailed environment, or may be a standard built system. Regardless of how you build and use the honeypot, it's value lies in the fact that it is attacked.
Research honeypots are honeypots designed to gain information on the black hat community. These honeypots do not add direct value to a specific
organization. Instead they are used to research the threats organizations face, and how to better protect against those threats. Think of them as 'counter-intelligence', their job is to gain information on the bad guys. This information is then used to protect against those threats.
Al Herrero, Ui Zurutuza, El Corchadohave proposed a system of honeyware to overcome the shortcomings of low interaction honeypot by using a client to make the multiple requests and then to use the Honeyware tool to scan the target machine. In this scenario, Honeyware would be able to compare both results and check if the web page used any techniques to mask its malicious behavior.[3]
It Kuwatly, Mi Sraj, Zr A1 Masri, Hi Artailhave combined different components like Data Logging Component, Data Analysis Component, Signature Extraction Component, Data Logging Component, Data Analysis Component, Signature Extraction Mechanism for attack pattern analysis and signature. [5]

III. PROPOSED APPROACH

This paper proposes an approach to have an easy access to the different kinds of potential threats using pattern matching. It uses the technique in which predetermined type so inputs which leads to the threat to the database or the website as a whole. It overcomes the limitations of the honeypot tool which requires file access to the user accessing the system. The only limitation of the proposed system is that the port number 80 of the client should be kept open.
This particular approach combines the concept of data collection, pattern matching and statistics. The major two parts of the website that is prone to an attack is the URL section and the Form fields. In this paper we have demonstrated both the limitations and performed attacks so as to see its particular reflection in the database, now using the pattern matching we will be getting the data of the user activities from GET, POST, SERVER, COOKIE.
There are two ways the browser client can send information to web server: Get method and Post method.

3.1 Get Method

Before the browser sends the information, it encodes it using a scheme called URL encoding. In this scheme, name/value pairs are joined with equal signs and different pairs are separated by the word semcl;
uname1=pass1semcl;uname2=pass2semcl;
The GET METHOD produces a string that appears in the server logs.
The string length is restricted to 1024 characters only.
The PHP provides $_GET associative array to access all the sent information using get method.

3.2 Post Method

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING.
The POST method does not have any restriction on data size to be sent.
The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
The PHP provides $_POST associative array to access all the sent information using POST method.

3.3 Server Method

$_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.
The following methods used:
$_SERVER['SERVER_ADDR']
$_SERVER['SERVER_NAME']
$_SERVER['REQUEST_METHOD']
$_SERVER['REQUEST_HOST']
$_SERVER['REMOTE_ADDR']

3.4 Cookie

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.
The isset() function is used to check whether the cookie is set or no.

IV. HOW THE SYSTEM WORKS

a. The system first presents the login screen to the any user. If the user is an admin, he has the total administrative control over the system such as control the server, view the logs and statistics, etc. If anyone other than the admin uses this telnet service his activity is logged onto the server.
b. On the other hand, all the traffic entering the network in the form of TCP, IP, UDP or ICMP form and their details including their payloads in logged onto the server.
c. The administrator first enables the two servers by starting them on. Then the processing of the logged data can be done.
d. The traffic captured through the HTTP service is processed as follows: The logs containing the malicious IPs are displayed along within their cause, number of hits on the database and date of first and last hit. To categorize any internet protocol address as malicious we have written pattern matching algorithm and tracking script. This algorithm and script analyzes, unusual source or destination address, reserved ip address entering the network, illegal combination of TCP flags in the TCP header or illegal contents in the UDP and ICMP headers. The attackers generally used specially crafted packets with these illegal settings. Any IP that gets trapped into these algorithm and script is termed as malicious and will be reflected on the php site that is running.
e. Then the admin may click on any malicious IP to get more information about it. This enables the IP Address Inspector screen to be opened.
   This contains the following contents:
   1. The overview about the IP: It has the IP address and its URL listed.
   2. The Settings service: It shows he User Agent, Connection and Host settings like browser, Operating System, etc.
   3. Trace route: This facility shows the hops required to reach the malicious IP from the source that is it traces the route from source to destination In the same way the admin can look up the user logs consisting of the information like username, password, session time in, session time out, IP address used, files visited, packets downloaded etc.
f. Then we present different the statistics to the admin on the basis of the gathered data. That is we can filter the incoming stats on basis of the IP address having performed some unusual activities, day to week logs, based on types of attack performed etc. This graphical and statistical views may help him to take decision or devise methods to improve the current security system.

Pattern Matching:
The algorithm will consider the following patterns for the attacks which is generalized and only websites that are vulnerable will respond to the same.

1. SQL Injection
   1. “SELECT” + ”;”;
   2. “SELECT” + “(“;
   3. “SELECT” + “)”;
   4. “SELECT” + ”=“;
   5. “SELECT” + “*”;
   6. “DELETE” + ”(“;
   7. “DELETE” + ”)“;
   8. “DELETE” + “=”;
   9. “UPDATE” + “(“;
   10. “UPDATE” +”)“;
   11. “UPDATE” +”=“;
   12. “DELETE” + ”;“;“

2. Inclusion
   1. “SELECT” + ”;”;
   2. “SELECT” + “(“;
   3. “SELECT” + “)”;
   4. “SELECT” + ”=“;
   5. “SELECT” + “*”;
   6. “DELETE” + ”(“;

7. “DELETE” + ”)“;
8. “DELETE” + “=”;
9. “UPDATE” + “(“;
10. “UPDATE” +”)“;
11. “UPDATE” +”=“;
12. “DELETE” + ”;“;“

3. Code Injection

1- "htmlentities()";
2- "mysql_real_escape_string";
3- "htmlspecialchars()" ;
4- "strip_tags()";
5- "addslashes()";

4. Cross Side Scripting

1-“<“+”>”+”</”+”%>”+”<%”+”href”+”echo” + “select”+”(”+”)”+” *”;
2- “insert”+ “< “ + ”>”+”</”+ ”%>”+ ”<%”;
3- “delete”+ “< “ + ”>”+”</”+ ”%>”+”<%”;
4- “select”+ “< “ + ”>”+”</”+ ”%>”+ ”<%”;
5- "<?" + "$_GET" + "?>";
6- "<?" + "echo" + "$_GET" + "?>";
7- "<?" + "$_POST" + "comment";
8- "<?" + "echo" + "$_POST" + "comment";
9- "<script>" + "</script>";
10-"?>" + "<$" + "$>";

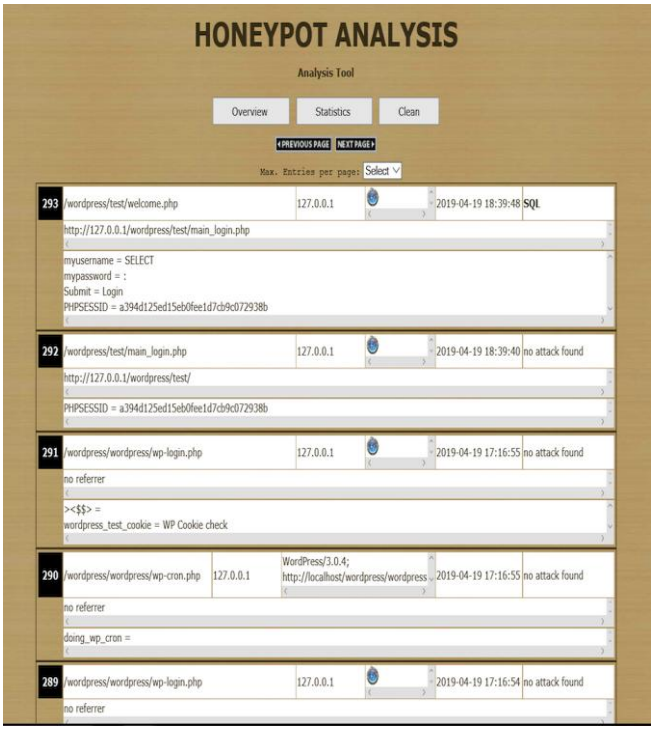

Fig-1 Overview of the network Traffic

Loudest IPs:

127.0.0.1	293 Hits	[100%]

Hits per module:

phpMyAdmin	293	[100%]

User Agents:

Number of distinct user agents:	4

Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.17) Gecko/2009122116 Firefox/3.0.17 (.NET CLR 3.5.30729)	248 Hits	[84.64%]
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/46.0.2486.0 Safari/537.36 Edge/13.10586	30 Hits	[10.24%]
WordPress/2.9.1; http://localhost/projects/wordpress	12 Hits	[4.1%]
WordPress/3.0.4; http://localhost/wordpress/wordpress	3 Hits	[1.02%]

Attacks:

Total number of detected attacks:	58	[19.8%]
Number of distinct attack types:	6	

Attacks per module:

phpMyAdmin	58 Hits	[100%]

Attack Types:

INCLUSION	36 Hits	[62.07%]
XSS	24 Hits	[41.38%]
SQL	4 Hits	[6.9%]
DIR-Change	3 Hits	[5.17%]
INJECTION	3 Hits	[5.17%]
CURL	3 Hits	[5.17%]

Most often used attack patterns:

http://localhost/projects/wordpress//wp-admin/	13 Hits	[22.41%]
<script>alert('hello')</script>	6 Hits	[10.34%]

Fig-2 Statistics of the traffic

V. CONCLUSION

This paper proposes a tool useful for the analysis of various types of attack on a vulnerable website. It also talks about the how different types of attack identified by their signature using tracking script and pattern matching algorithm. This tool helps in preventing any further attack from a potential attacker.

VI. ACKNOWLEDGMENT

For publishing this paper, we had to take help and guidelines of some respected persons, who deserve our greatest gratitude. We would like to show our gratitude to Dr.Arun Chavan, Head of Computer Department, V.I.T. for his useful insights. We would also like to thank Prof. Ravindra Sangle for giving us guidelines throughout numerous consultations.

VII. REFERENCES

[1] International Journal of Advance Research in Computer Science and Management Studies "Web Application Honeypot" Volume 2, Issue 3, March 2014
[2] International Journal of Scientific and Research Publications, Volume 3, Issue 8, August 2013 1 ISSN 2250-3153 "Web based honeypot systems"
[3] Al Herrero, Ui Zurutuza, El Corchado, A Neural-Visualization IDS for Honey net Data, International Journal of Neural Systems, 22 (2012]
[4] IO Chang, Ti Yi-Lang, Design of virtual honeynet collaboration system in existing security research networks, in: 2010 International Symposium on Communications and Information Technologies (ISCIT), 2010, pp 798-803
[5] It Kuwatly, Mi Sraj, Zr A1 Masri, Hi Artail, A dynamic honeypot design for intrusion detection, in: Pervasive Services, 2004' ICPS 2004i IEEE/ACS International Conference on, IEEE, 2004, pp 95-104
[6] 2016 Internaylional Conference on Computing for Sustainable Global Development (INDIACom) "Honeypot-Based Intrusion Detection System: A Performance Analysis "