IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 10, October 2015.
www.ijiset.com
ISSN 2348 – 7968

Securing Web Application By Detect and Prevent Input Validation Attacks Using Swart Tool

Deepa V Subramaniam M.E.,CSE(Pursuing)
Computer Science & Engineering,Thakur College of Engineering & Technology,
Mumbai,Maharashtra,India.

Mr.Kiran Bhandari M.E.,Ph.d(Pursuing),Associate Professor,
Computer Science & Engineering,Thakur College of Engineering & Technology,
Mumbai,Maharashtra,India.

Mrs.Veena Kulkarni M.E.,Assistant Professor,
Computer Science & Engineering,Thakur College of Engineering & Technology,
Mumbai,Maharashtra,India.

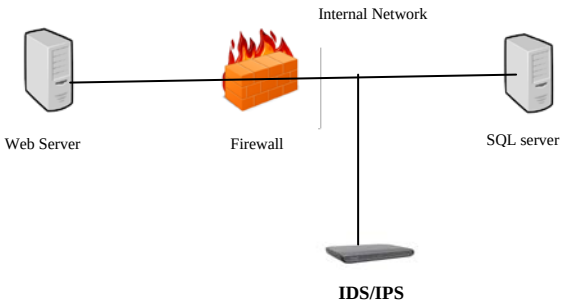
Abstract

Web applications are used to provide e- services such as social networking over the internet, the attacks over the web applications have also increased. Many systems are currently present for detecting and preventing web attacks, they are often limited in scope and functionality. Many existing tools can only respond to certain types of attacks. Most of the systems are also platform specific. These techniques are earlier used for purpose of network security but with recent advancement in application threats these tools are now used for securing application level attack. SWART is an Application Intrusion Detection System tool which secures web application by providing early warning to the attacker or the malicious user then it might be possible that the application is not further exploited for finding the loopholes. The proposed approach with PHP based web application and also perform Chi Square test to validate the assumptions. The Chi square Hypothesis testing is used to prove the importance of response tool for the web application.

Keywords: Input Validation Attacks,Cross-site Scripting Attack, Brute Force Attack, Chi Square Testing,SQLinjection Attack.

1. Introduction

Nowadays, Attacks mainly focus to exploit vulnerabilities at application level. Intrusion Detection and prevention Systems is the process of monitor the network and analyzing for signs of possible flaws occurring in the web application. An IDPS able to detect traffic indicative of the new attack. An intrusion detection system (IDS) is automates the intrusion detection process. The proposed approach detect vulnerabilities by using Signature-based Intrusion Detection System(SIDS). A Signature-Based Intrusion Detection System compares the signatures or attack patterns, and compares the patterns against the well-known attack patterns that are stored in the database. As the attack pattern matches to the packed patterns, The Signature-Based Intrusion Detection System(SIDS) generate and sends report to the administrators.An intrusion prevention system (IPS) do all the process of an intrusion detection system. An Intrusion detection and prevention systems(IDPS) identify flaws in the web application, log information about the intruder, attempt to stop the attack and produce reports for security administrators. Fig(1) shows the block diagram of Intrusion Detection and Prevention System IDPS.



Fig(1):Intrusion Detection and Prevention System(IDPS)

2. Input Validation Attacks

An attacker intentionally send unusual input in the hopes of confusing the web application known as Input Validation Attack(IVA). If an attacker discovers that the application makes unsupported expectations about the type, length, format, or range of input data.. While the network level entry points are fully secured; The input to the web application used to test the system and a way to execute code on an attacker's behalf. If the application blindly trust the input of an attacker, it may be susceptible to the input validation attacks.These attacks provide vulnerability to the web application.The proposed approach is explained about types of input validation attacks.

ISSN 2348 – 7968

2.1 SQLInjection Attack:

SQL injection attack also known as code injection attack, it is the most prominent attack used by the hackers through data-driven web applications. SQL injection allows the hacker to create, delete, update, read and alter data stored in the back-end database. The malicious attacker may restructure the actual SQL statements and execute vulnerable code exposing to the web application. SQL injection attack can occur when a web application utilizes the user input data without proper validation or encoding the query. Tautology-based SQL injection attacks are bypass user authentication and extract data while inserting a tautology in the WHERE clause of a SQL query. The query convert the original condition into a tautology, that causes all the rows in the database table are open to an unauthorized user. SQL tautology Query is, "or <comparison expression>", where the comparison expression uses one or more relational operators to compare the operands and generate an always true condition.

SQL injection Example:

If an unauthorized user input user id as **admin** and password as **anything' or '1'='1** then the resulting query will be:

```
select * from user _details where user_ id = 'admin' and passwor
d = 'anything' or '1'='1'
```

2.2 Cross-Site Scripting Attack:

In the Cross-Site Scripting attack the Scripts (JavaScript, VBScript, ActiveX, HTML) are embedded in web pages that run in the browser. These scripts can access cookies, create requests for getting private information. Web application often takes user input and use this input as a part of a web page. The use of XSS might compromise private information, manipulate, and create requests that can be mistaken for a valid user, or execute malicious code on the end-user systems. The data is usually layout as a hyperlink containing malicious data and which is distributed over any web page on the internet. In a typical XSS attack the hacker contaminates a legitimate web page with the malicious client-side script. When a legitimate user visits this web page the script is downloaded to the hacker browser and executed.

Example of Cross-Site Scripting:

An user to visit the specially crafted link by the attacker. When the user visit the link, the crafted code will get executed by the user's browser.

```
<?php

$name = $_GET['name'];

echo "Welcome $name<br>";
```

The attacker will craft an URL as follows and send it to the victim:

guestbook.php? name=guest<script>alert('Hacking attempt

detected')</script>

2.3 Brute Force Attack:

In a brute force attack, an automated software is used to create a large number of successive expectations as to the value of the desired data. Brute force attacks may be used by criminals to crash encrypted data, or by security experts to test an organization's network security. In this type of attack it could be very time consuming to try all possible combinations. A brute force attack is a trial method used to obtain information such as a user password or personal identification number (PIN). Brute force (also known as brute force cracking) is a trial and error method used by the application programs to crack the encrypted data such as passwords through exhaustive effort (using brute force) rather than employing logical strategies.

3.Proposed Approach:

SWART (Secure Web Application Response Tool) consists of four modules. Assumed that the application contains login page which uses SQL query to be connected with underlying database. .Fig(2) shows Application framework for Detection and Prevention of Sqlinjection attacks,Cross-site scripting Attacks and Brute force attacks are the important part of the proposed approach.

- Brute Force Attack Module(Login sensor Module)
- SQLInjection Module
- Cross-site Scripting Module

These three modules implemented by the following process

- Classification Analyzing Engine
- Intrusion Detection Engine
- Threshold Comparator Module
- Response Redirect Module

The proposed mechanism used to detect and prevent input validation attacks but it also detect and prevent other types of attacks such as Invalidated redirects and forwards.

3.1 Session Manager:

Every application will need to be registered to the database. Session Manager provided a unique Authentication-token key When an application communicates with the server, the authentication token and the domain from the request headers are validated and only if the tokens are authenticated, a short-lived session will be set on the client side for near future access and the server will continue processing requests from the same.

ISSN 2348 – 7968

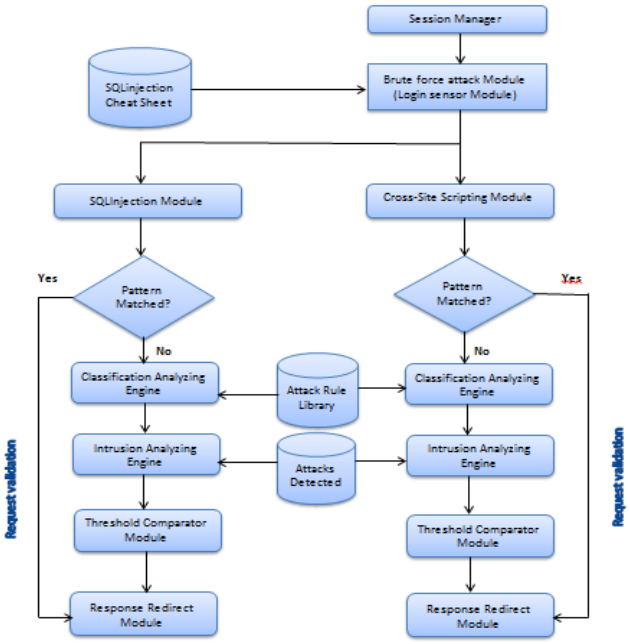
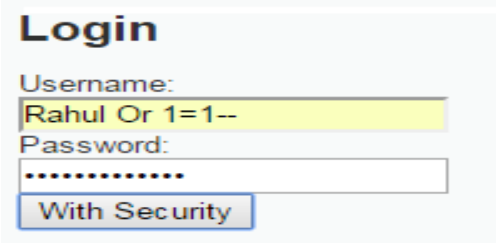
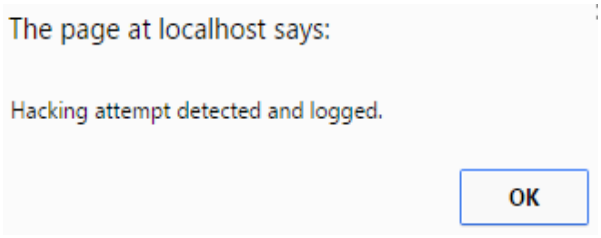


Fig 2 : Proposed Framework

3.2Brute Force Attack Module(Login Sensor Module):

The first module is Login Sensor Module. As the intrusion occurs from login module & Brute force attack is the most prominent attack that occurs at the login form. The proposed approach develop login form having username & password connected to underlying attack rule library. When the authorized user will enter the details the multistep verification form or the application form will be generated but if intruder tries to attack the login form with SQL query the attack pattern will be matched & intruder will be taken to alarm page.





3.3 SQLinjection Module:

SQLinjection module checks the attack pattern is matched from the attack rule library. If the attack pattern is matched, Intrusion D e t e c t i on Engine detect the attack as intrusion & add attack score whenever attack is detected. Threshold Comparator Module compares the intrusion detection engine score with the adjusted threshold value. Response Redirect Module redirects the user according to the threshold value.

3.4 Cross-Site Scripting Module:

This module also checks the attack pattern is matched from the a t t a c k rule library. If the attack pattern is matched, Intrusion D e t e c t i on Engine detect the attack as intrusion & add attack score whenever attack is detected. Threshold Comparator Module compares the intrusion detection engine score with the adjusted threshold value. Response Redirect Module redirects the user according to the threshold value.

3.5 Classification Analyzing Engine:

Classification Analyzing Engine which classifies the input from the user to the attack into different categories. The attack pattern is matched from the attack rule library & pattern is matched. Blacklisting is used for identifying how the user is using the application. For extending the tool to another level of attacks such as authentication or business logic we can develop policy rules for identifying the malicious activity.

Classification Of Attacks	Risk Status
Authentication Testing	High
Authorization Testing	High
Business Logic Testing	Medium
Data Validation Testing	High
Denial of Service Testing	High

Table 1: Classification of Attacks

3.6 Intrusion Detection Engine:

D e t e c t i on Engine which d e t e c t the attack as intrusion & add attack score whenever attack is detected. Whenever attack is detected the attack is logged in the intrusion table having field as Username, Attack Category, and Attack score for anal yzing attack rates over the application. If no intrusion is detected then connection to database is established after sanitizing the input. This provides double security to the database.

Intrusions occurs	Risk score
Input contain SQL Query	4
Input Contain Scripts, JavaScript , HTML	3
Session Invalidation	2
Privilege access	4
Invalidated redirects	2
Unauthorized directory Access	2

Table 2 Intrusion Risk Associated

ISSN 2348 – 7968

3.7 Threshold Comparator Module:

Threshold Comparator Module which compares the intrusion detection engine score with the adjusted threshold value. The threshold value can be adjusted anytime according to the risk identified.

Risk	Score	Response Redirect
1-5	Low	Login Failed
5- 10	Medium	Account Logout
15-above	High	Alarm Page

Table 3 Threshold Score

3.8 Response Redirect Module:

Response Redirect Module which redirects the user according to the threshold value. If the attack occurs at the login time direct alarm page is generated. Analyzed the vulnerable input points of the web application are URL, Form Input, and Cookies. At runtime the Validation response of the application are checked for analyzing intrusions

3.9 Validation Response Algorithm I

Input : Login Input
Output : Validation Response
Set I → Login Input
If I Matches AttackPattern
Then set Attack → True
Goto Response Redirect Alarm Page
Else Login Successful
If (2 < login attempts < 5)
Then Goto Response
Warning Message

3.10 Validation Response Algorithm II

Input: User Input
Output: Validation Response
Do While(SessionID != null)
{
S →User Input
{
Check for Match Attack
Classification
{
If Match→ True
{
Check for Number Of Intrusions
{
Set Attack → True;
Set Attack → Attack +1;
Set Attack Score →Attack Points + Attack Points
}
Goto Response Redirect.
}
}
Else Connect to Database & Sanitize Input
}}

5. chi square test

The Chi square Hypothesis testing to prove the importance of response tool for the web application. Hypothesis testing is a test for accepting and rejecting the assumption about the population. Population here refers to the kind of data over which test is applied. Chi Square test (X2) is a non-parametrical test to find the association or dependency between the classified variables. In hypothesis testing to test whether number of users and number of forms can affect the number of attacks over the web application. Chi square test (X2) is divided in to three categories for testing.

1) Chi square test for Goodness of Fit: - It is applied when we have one categorical from the single population and want to test how close the observed values are from the expected values.

2) Chi Square test for Homogeneity: - It is applied when we have one categorical data from two different populations and we want to test the frequency distribution across different population.

3) Chi Square test for Independence: - It is applied when we have two categorical data from single population and we want to test the dependency between the variables. Chi square test for independence for finding dependency between the number of users and number of attacks and also for the requirement of response tool. Chi square test for independence for finding dependency between the number of users and number of attacks and also for the requirement of response tool. For performing test required the following:

Test1: Testing Dependency for Number of users and Number of Inputs for Number of Attacks.

Step1: State of Hypothesis

H0 = the number of users and number of inputs have no affect on Number of attacks over a web application.

H1 = the number of users and number of inputs affect number of attacks over a web application.

H0 is assumed as null hypothesis. Number of users, Number of Inputs and Number of attacks are variables.

Step2: Significance Level

The significance level we have chosen is 0.001 which states that if H0 is accepted than it has 0.001 probability likely to dependent. If H0 is rejected than H1 has 99.99% likely to be dependent on variables**.** We have used contingency table of 3x2. Where degree of freedom df is (r-1)(c-1) which is resulted as 2**.**

$X^2 = \Sigma\, n\,(O - E)^2$……(1)

Test 2: Testing effect of warning response tool

Step1: State of Hypothesis

H0 = Warning response tool have no effect on web application attacks. H1=Warning response tool affect in lower web application attacks **Step2:** Significance Level

The significance level chosen 0.001 which states that if H0 is accepted than it has 0.001 probability likely to dependent. If H0 is rejected than H1 has 99.99% likely to be dependent on variables and also used possibility table of 2x 2. Where degree of freedom is (r-1)(c-1) which is resulted as 5.

ISSN 2348 – 7968

	Observed Value(O)	Expected Value(E)	(O-E)	$(O-E)^2$	X^2
No Of Users	**15**	**30**	**15**	**225**	**75**
No Of Inputs	**25**	**50**	**25**	**625**	**12.5**
No Of Attacks	**100**	**50**	**50**	**100**	**2**
Chi Square Value					**22**

Table 4: Chi Square Test Table

6. Conclusion

Web applications are often deployed with minimum security as the developer mainly focuses on the implementation of the application. Moreover, the development team of the application lacks the security expertise because of budget constraint due to which application frauds & threats are increasing at an alarming rate. The proposed mechanism is a tool for detecting & preventing web application attack .The proposed approach with PHP web application and also perform Chi Square test to validate the assumptions of Input Validation Attacks. The proposed system aims at creating an open source cross platform application side intrusion detection and response framework to detect and respond to web application based intrusion attacks.

References

[1] Mudzingwa, D.; Agrawal, R. A study of methodologies used in intrusion detection and prevention systems (IDPS), Orlando, FL *Southeastcon, 2012 Proceedings of IEEE Year: 2012*

[2] Sampda Gadgil, Sanoop Pillai,Sushant Poojary." SqlInjection Attacks And Prevention Techniques", *International Journal on Recent and Innovation Trends in Computing and Communication Volume: 1 Issue: 4 293 –296, APR 2013*

[3] Lashkaripour, Z.; Bafghi, A.G. "A security analysis tool for web application reinforcement against SQL injection attacks (SQLIAs)", *Information Security and Cryptology (ISCISC), 2013 10th International ISC Conference on Year: 2013.*

[4]https://www.owasp.org/index.php/Blocking_Brute_Force_ Attacks accessed on dt. 24/02 /2013

[5]http://en.wikipedia.org/wiki/Brute-force_attack accessed on 2/3/2013

[6]http://www.codeproject.com/Articles/17111/Preventing-a-Brute-Force-or-Dictionary-Attack-How accessed on 5/3/2013

[7] Konark Truptiben Dave," Brute-force Attack "Seeking but Distressing", *International Journal of Innovations in Engineering and Technology (IJIET), Vol. 2 Issue 3 June 2013 ISSN: 2319-1058*

[8] S.Shalini1 , S.Usha, "Prevention Of Cross-Site Scripting Attacks (XSS) On Web Applications In The Client Side", . *IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011.*

[9] Grossman, J., Hansen, R., Petkov, P., Rager, A., and Fogie, S. *Cross site scripting attacks: XSS Exploits and defense.. Syngress, Elsevier, 2007.*

[10] Hallaraker, O. and Vigna, G. Detecting Malicious JavaScript Code in Mozilla. *10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05), pp.85– 94, 2005.*

[11] Attacks Jyoti Snehi1 , Dr. Renu Dhir, "Web Client and Web Server approaches to Prevent XSS Attacks", International Journal of Computers & Technology www.cirworld.com Volume 4 No. 2, March-April, 2013

[12] Dr R.P Mahapatra, Ruchika Saini, Neha Saini," A Pattern Based Approach to Secure Web Applications from XSS Attacks". *International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 3, June 2012.*

[13] Punam Thopate, Purva Bamm, Apeksha Kamble, Snehal Kunjir, Prof S.M.Chawre,"Cross Site Scripting Attack Detection & Prevention System", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 11, November 2014.*

[14] R. Ezumalai , G. Aghila , "Combinatorial Approach For Preventing SQL Injection Attacks" , IEEE International Advance Computing Conference , 2009 , PP.1212-1217