AVS Scanner: A Black Box Vulnerability Scanner with Minimum False Positives

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Information Security

Submitted By
Sparsh Sharma
(Roll No. 801233023)

Under the supervision of:

Dr. Maninder Singh
Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

July 2014

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*AVS Scanner: A Black Box Vulnerability Scanner with Minimum False Positives*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Information Security* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Maninder Singh* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Sparsh Sharma)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Maninder Singh)
Associate Professor, CSED
Thapar University, Patiala

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

The successful completion of any task would be incomplete without accomplishing the people who made it possible and whose constant guidance and encouragement secured the success.

First of all I wish to acknowledge the benevolence of omnipotent God who gave me strength and courage to overcome all obstacles and showed me the silver lining in the dark clouds.

With the profound sense of gratitude and heartiest regard, I express my sincere feelings of indebtedness to my guide Dr. Maninder Singh**,** Associate Professor, Computer Science and Engineering Department, Thapar University for their positive and excellent guidance, constant encouragement, keen interest, invaluable co-operation, generous attitude and above all his blessings have been persistent source of inspiration for me.

I am grateful to Dr. Deepak Garg, Head of Department, Computer Science and Engineering Department, Thapar University for the motivation and inspiration that triggered me for this thesis.

Last but not the least I would like to express my heartfelt thanks to my parents and my friends who with their thought provoking views, veracity and whole hearted co-operation helped me in doing this thesis.

Sparsh Sharma
M.E (IS)
801233023

Abstract

With the increase in dependence on the web and web applications, the security over web is also becoming an important concern. Various Surveys and News Reports show the significant increase in the number of cyber crimes which are possible only due to the presence of security vulnerabilities in the web applications. Most of these web vulnerabilities exist due to lack of awareness regarding security among the web developers and designers, as a result large number of websites are still lacking security features and are vulnerable. These vulnerabilities if not patched can lead to various adverse affects like database stealing, shell hijacking and arbitrary command execution and much more.

This thesis demonstrates how simple it is for the attackers to automatically find and exploit security vulnerabilities in web applications. Understanding these security vulnerabilities can be somehow a complex task for the web developers and for the individuals who designs their websites from third-party sources. So for easing their life and with an aim of making web world more secure AVS Scanner, a simple to use but effective vulnerability scanner that automatically scans the websites with the aim of finding critical security issues in web applications, is presented. This thesis also explains about importance of False Positives and False Negatives in a Vulnerability Scanners and discuss about the various logics used in the AVS Scanner that has helped to reduce the False Positives to minimum. Performance comparison between various existing vulnerability scanners and AVS Scanner is performed in term of important parameters like False Negatives, False Positives, and Resource Consumption. For examining the accuracy and efficiency of AVS Scanner in comparison to other scanners available in the market, various popular and high-profile websites, which includes some well-known social networking websites like facebook.com, flickr.com and few educational websites like thapar.edu, nitkkr.ac.in, nitc.ac.in, were analyzed and results were overwhelming.

TABLE OF CONTENTS

Certificate i

Acknowledgement ii

Abstract iii

Table of Contents iv

List of Figures & Tables vii

CHAPTER 1: Introduction 1

1.1 Web Applications

1.1.1 Benefits of Web Applications 2

1.1.2 A Web Application Toolkit 2

1.1.3 Intercepting Web Proxy Tools 3

1.2 Thesis Outline 3

CHAPTER 2: LITERATURE REVIEW 5

2.1 HTTP Protocol 5

2.1.1 HTTP Methods 5

2.1.2 HTTP Headers 5

2.2 Approaches for Web Application Testing 6

2.2.1 White Box Testing 7

2.2.2 Black Box Testing 8

2.3 Web Application Firewall 9

2.3.1 Various WAF available in market 10

2.4 False Positives and False Negatives 10

2.5 Clickjacking 11

2.5.1 Clickjacking Adverse Effects 12

2.5.2 Frame Busting 13

2.5.3 Image based Authentication bypass by Clickjacking 13

2.5.4 Clickjacking Protection in Mobile Websites 13
2.5.5 Effective Methods for Dealing with Frame Attacks 14
2.6 Open Web Application Security Top Attacks 14
2.7 Cross Site Scripting Protection Techniques 19
2.7.1 Server Side Protection 20
2.7.2 Client Side Protection 20
2.8 Adverse Effect of XSS on the Internet 21
2.9 XSS Detection Approaches 21

CHAPTER 3: Problem Statement 23
3.1 Gaps in study 23
3.2 Problem Statement 23
3.3 Objectives 24

CHAPTER 4: Implementation 25
4.1 Automated Vulnerability Scanner (AVS): 25
4.1.1 Modes of AVS Scanner 25
4.1.2 Attacks Supported and Logics used 26
4.2 Hall of Shame 29
4.3 Banner Grabbing and Information Gathering 30
4.4 Python 30
4.4.1 Python Modules Used 31

CHAPTER 5: Experimental Results 33
5.1 Comparison of AVS Scanner with other Scanners 33
5.2 XSS Exploitation Payloads 35
5.2.1 Web Defacement 35
5.2.2 URL Redirection 36
5.2.3 Injecting Trojon Functionality 36
5.2.4 Cookie Stealing 38
5.2.5 XSS Port Scanning 39

CHAPTER 6: Conclusion and Future Scope **41**

6.1 Conclusion 41

6.2 Future Scope 41

References **43**

List of Publications **46**

List of Figures & Tables

List of Figures

	Page
Fig 2.1: Black Box Testing and White Box Testing	7
Fig 2.2: Web Application Firewall in Action	9
Fig 2.3: WAF Architecture	10
Fig 2.4: Clickjacking Attack to hack webcam of Victim	12
Fig 2.5: Web Attacks Distribution in Web Applications	15
Fig 2.6: Reflected XSS	17
Fig 2.7: Stored XSS	17
Fig 2.8: Cross Site Request Forgery attack Flow	19
Fig 4.1: AVS Scanner	26
Fig 4.2: AVS Scanner for reducing False Positives	27
Fig 4.3: XSS Payload Count in different Vulnerability Scanners	27
Fig 4.4: XSS Second Logic	28
Fig 4.5: Clickjacking Logic	28
Fig 4.6: AVS Scanner Hall of Fame Section	29
Fig 4.7: AVS Banner Grabbing of Website	30
Fig 5.1: False Positives, False Negatives, Resource Utilization Comparison	34
Fig 5.2: XSS Website Defacement	35
Fig 5.3: XSS URL Redirection	36
Fig 5.4: XSS Trojon Functionality	37
Fig 5.5: XSS Fake Form on a Bank Website	37
Fig 5.7: XSS Cookie Stealing on avanthagroup	39
Fig 5.8: XSS Port Scanning	40

List of Tables

Table 5.1: AVS Scanner Performance in Real World	34

Chapter 1
Introduction

Internet has become an important part of our lives. Every day users come across various web applications that have been implemented using a variety of technologies. Every Web Application is normally composed of a backend, a server side component (can be any language like PHP, ASP, even interpreted languages like Python) running on some Provider`s Server and a client part running in the user`s web browser. Due to this complex nature in the way the web applications are implemented there exist various web vulnerabilities which can lead to security breach, data leakage and theft and much more.

Since with the increasing dependence of critical applications such as e-banking, e-commerce, on the internet, there is a great need to provide security check on these websites and web applications. Many web application security vulnerability arise from the developer`s habit of coding insecurely. Major vulnerabilities include SQL injection, Cross-Site Scripting (XSS), Brute-force, Clickjacking etc.

The occurrence of these vulnerabilities could be reduce by educating and motivating the web developers to code securely and making the secure coding their habit. But these approaches require huge effort and cost to train the developers. So these constraints make the developers to stick to an insecure but fast model of coding.

1.1 Web Applications

Web Applications are designed to allow users to perform almost every task that one can perform online. Following are the tasks which one can perform using Web Applications:

- Online Shopping
- Social Networking (Twitter, Facebook)
- Bill Payment
- Search (Google, Bing)
- Auctions (Ebay, Olx)
- Entertainment (YouTube)

1.1.1 Benefits of Web Applications

HTTP Protocol used for accessing the internet is lightweight and connectionless. So this means the server does not need to maintain connection with every user trying to connect to it. Also HTTP allows tunneling over the other protocols for secure communication. Web Browser is found in every user`s computer and mobile and web applications has a dynamic interface that makes it possible for the user to view the web page without the need of having some pre-defined client software for a particular web application. The technologies and the languages used for building the web applications are very simple to learn and use. Also various web developments tools and editors are available in the market that makes it possible even for a beginner to build a web application.

1.1.2 Web Application Toolkit

For most of the web attacks, the only thing that is required is a web browser. Some attacks can be performed using the web browser like Mozilla only and there are attacks that require some extra software to be installed either in the web browser as an add-on and some require installing an external software working side-by-side with the web browser.

Some of the famous add-ons that need to be installed in pentester`s web browser for a particular attack to take place are [34, 35]:

- Hackbar- a Mozilla add-on used for web attacks like Cross-site Scripting, SQL Injection etc.
- Live Headers- another Mozilla add-on used for capturing HTTP Header values of a web page.
- XSS ME- checks for reflected XSS Vulnerability.
- No-Script- This add-on is used for disabling the use of JavaScript in a web-page thus prevents JavaScript from executing in user`s browser.
- Clickjacking Reveal- This add-on detects the presence of Clickjacking vulnerability in a webpage.

1.1.3 Intercepting Web Proxy Tools

These tools are mostly used for intercepting the POST requests before sending to the web server. So this means all the requests to and from some web server can be

intercepted and modified before actually reaching its final destination. So in this way pentester can view and modify the hidden form fields and also the post request parameters.

These Tools are very useful in finding vulnerabilities like Cross Site Request Forgery (CSRF), Cross Site Scripting etc [35].
Various Web Proxy Tools available are:

- Burp Suite
- WebScarab
- Paros
- Fiddler

Out of all the Web proxy tools available, Burp Suite is the mostly used and has all the features that are required for intercepting the requests.

Following are the features of Burp Suite:

- Proxy: Burp Suite act as a proxy for all the HTTP/HTTPS request to and from the web server. So attacker can act as a man in the middle using the Burp Suite and thus can intercept the requests that are coming from the server.
- Scanner: Burp Suite has the important functionality for scanning website itself for the presence of security vulnerabilities.
- Repeater: This tool is for generating and sending the HTTP requests to the web server and checking for the response code.
- Decoder: This allows decoding all the encoded data and also converting the raw data to some encoded form.

1.2 Organization of Thesis

This thesis is organized as follows:

Chapter 2 is literature survey that includes explanation of HTTP Protocol, approaches for Web Application Penetration Testing, Web Application Firewalls, OWASP Top 10 Web attacks and their available mitigation techniques.

Chapter 3 explains about the problem statement including the gap in existing system and contribution work to the area.

Chapter 4 is Implementation that includes detailed explanation of the AVS Scanner which is a black box vulnerability scanner and its features and functionalities.

Chapter 5 includes results in which Comparison of various available vulnerability Scanners with the AVS Scanner is performed, Proof of Concept of the various attacks that can be performed after detection of XSS is also shown.

Chapter 6 includes conclusion and suggestions for future enhancements of the AVS Scanner.

Chapter 2
Literature Review

2.1 HTTP Protocol

HTTP stands for Hypertext Transfer Protocol that is primarily responsible for accessing the internet. HTTP Protocol is a stateless protocol. Also HTTP is a connectionless protocol which means no connection is maintained during communication and every request is a new request [14].

2.1.1 HTTP Methods

GET and POST are the most commonly used HTTP methods and one need to understand the differences between the two before attacking the web-applications.

a) GET Method: In this method, all the parameters and input data are sent in the URL. So it's possible to bookmark the URL for the later use. So as the data is sent along with the URL, so this method is not used for sending sensitive information [12].

b) POST Method: In this method, the request data is sent in the body of the HTTP Header. So this method can be used to send data in both URL and body of the HTTP Header. This method is used in login forms where there is a need to submit sensitive information like passwords, credit card numbers etc [12].

2.1.2 HTTP Headers

HTTP uses various headers which have different functions and features. Some are used for both request and response and some are used for just a particular message type. Various HTTP Headers that are worth to know for web application penetration testing are as follows [11]:

a) General Headers

- Connection: This header tells the other end whether it should disconnect the TCP connection or wait for other messages.
- Content-Encoding: This header tells which type of encoding scheme has been used for encoding the data in the body. Encoding is usually done for fast data transmission [11].

- Content-Type: This tells the type of the content in the message body.
- Content-Length: This header specifies the length of the content of the message body.

b) Request Headers

- Host: This header specifies the full host name of the resource being requested.
- Accept: This header tells what type of data the client is expecting like image, text etc.
- User-Agent: This header gives information regarding the browser and other software being used by the client requesting the resources.
- Accept-Encoding: This HTTP request header tells the web server that the browser is ready and prepared to accept the compressed data [13].
- Referrer: This header tells about the host from which the request has been initiated.

c) Response Headers

- X-Frame Options: This header is the one responsible for mitigating the Clickjacking attack. This header decides whether the I-Frames can be used in a website or not.
- Content-Encoding: This header is set if the Accept-Encoding request Header was set. So this header tells the web browser in client side that whether the content arrived is compressed or not [13].
- Server: This header gives information regarding the server being used like server version, type etc.
- Expires: This header tells browser about the validity of a message.
- Set-Cookie: This header sets the cookie in the client`s browser and these cookies are returned in the future communication with the server.

2.2 Approaches for Web Application Penetration Testing

Penetration Testing is a process of evaluating an existing system for the presence of any loophole or security vulnerability. Penetration Testing is done with the permission of the owner and thus considered legal. Penetration Testing is a critical process and is done by list of professionals who uses different approaches to find flaws in a system

and they tries to think like a black-hat hacker and provide solutions for their mitigation [15].

2.2.1 WHITE BOX TESTING

White Box Testing gives the perspective of an internal attacker. In White Box Testing, penetration tester is given all the important information regarding the system he has to inspect. This includes even the source code of the website [26]. So the pentester has not much task to do as in Black Box Testing. In this approach source code of a web application is closely examined for the presence of any insecure code used that can lead to the compromise. This may appear too many that this approach has benefit over Black-Box approach but in actual sometimes this approach leads to various False Negatives. So White Box Testing should be followed by Black-Box Testing approach so that the results that are skipped by the first approach are caught by the later.

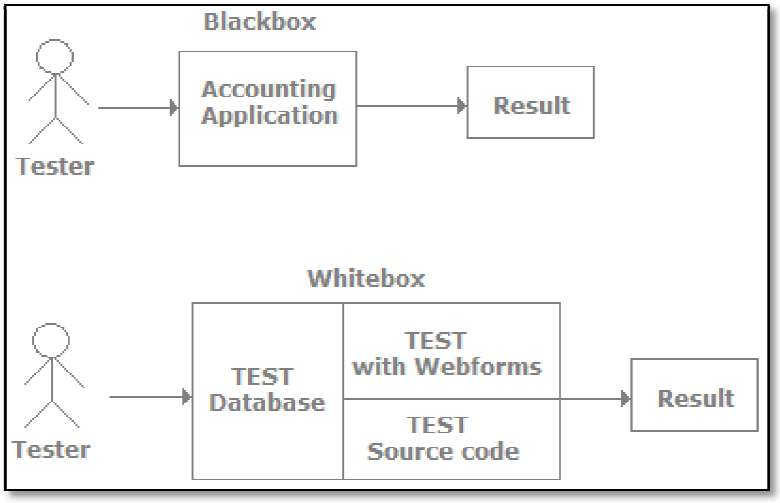

Figure 2.1 Black Box Testing and White Box Testing

Advantages of White Box Testing Approach:

- **Rigorous Testing:** This approach is ideal for the web-applications where each line of code is need to be closely inspected and thus all the internal working is examined properly. This approach offers more reliability in term of security checking.
- **Introspection:** The ability of the tester to look inside the application and recognizing objects and members programmatically. This is ideal in cases when the Graphical User Interfaces are changing frequently.

- **Less False Positive:** Since this approach works by analyzing the source code and not by any hit or trial method so chances of getting false results are very low.

Disadvantages of White Box Testing Approach:

- **Complexity:** This approach requires the tester to inspect all the details and source code available closely and thus require huge efforts from tester end.
- **Programming Language:** Since this approach requires analyzing the source code and thus the tester is require to be highly skillful in the programming language used by the web-application. So for web-applications written in different programming language require different testers.
- **Testing Duration**: This approach takes more time in completing its task of security checking.
- **Expensive**: This approach consumes more time and requires manual testing so this approach is quite expensive compare to the other approach.

2.2.2 Black Box Testing

Black Box Testing is a challenging approach as in this approach only information a pentester is given is just the target name .So this approach truly requires hacking skills. In this approach the pentester has to think like an attacker. In this approach, specially crafted payloads as inputs are sent to server and then their response is examined. Vulnerable applications respond differently and uniquely when some special input is sent and thus the presence of some specific response indicates that certain vulnerability exists [26]. Here pentester start with information gathering, then scanning the open ports and vulnerable services running from where he can enter into the system and can gain access [26, 19].

Advantages of Black-Box Testing Approach

- **Language Independent:** In this approach tester need not know the programming language used in web.
- **Ease of Use:** No need to worry about the internal working of the web application

- **Less Expensive:** This approach is cost-effective as it just involves the cost of purchasing the tool and same tester can performs all the tasks independent of the platform in which website is developed.
- **Require no knowledge:** This approach works by sending a specially crafted inputs and observing the outputs. So this approach doesn't require much knowledge regarding the internals of the language and logic**.**
- **Faster Testing:** This approach is faster compared to the previous approach as all the test cases are defined and there is just a need to apply them.

Disadvantages of Black-Box Testing Approach:

- **More False Positives:** This approach is prone to false results.
- **Tool Updation:** The tool need to be kept updated with newer attacks as well as need to make them capable to bypass WAF (web application firewall).

2.3 Web Application Firewall

A Web Application Firewall is a security add-on used in web-applications server to protect the web-applications from various web attacks like Cross-Site Scripting, SQL-Injection, Local File Inclusion (LFI), Remote File Inclusion (RFI) etc by observing the HTTP and HTTPS request made to the Web Server. Web Application Firewall is given add-on status because it works at application layer and keeps an eye on malicious application request which cannot be detected by Intrusion Detection System which works at Network Layer. A Web Application Firewall works by creating rules for the already known attacks and thus applying that rules to the HTTP/HTTPS request to detect and prevent various known web-attacks [17].

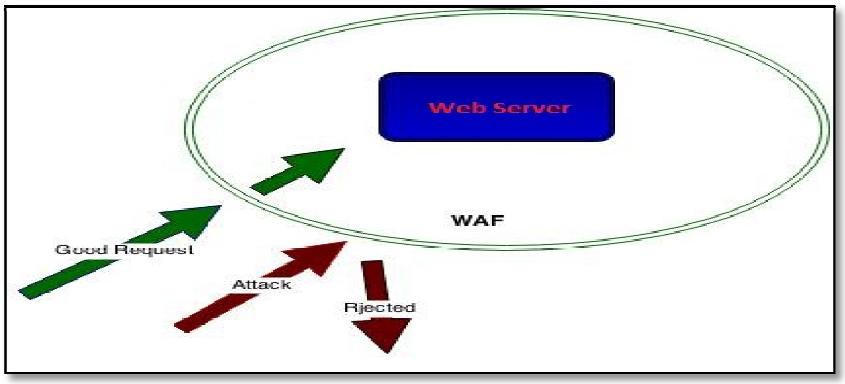

Figure 2.2 Web Application Firewall in action

2.3.1 Various Web Application Firewall available in the Market

Mod Security WAF: This is the most common and widely used application layer firewall to mitigate the various common web-attacks. Although these WAF works well only with known attacks whose rule are present in their database.

AQTRONIX WebKnight: Another open source WAF used for web servers and IIS servers. It has added features to deal with Buffer Overflow attacks, SQL Injection attacks etc.

ESAPI WAF: This is web application firewall presented by OWASP and is a Java Base WAF. Easy Installation and code based is some of its features.

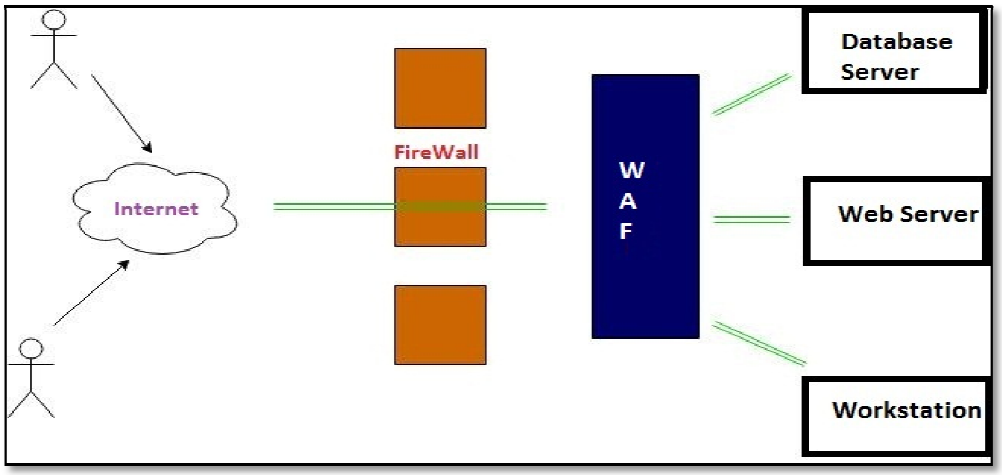

Figure 2.3 WAF Architecture.

2.4 False Positives and False Negatives

The main requirement of a vulnerability assessment tool is accuracy. So no matter how many attacks one tool supports if the accuracy is not there then that tool is of no use. So accuracy of a vulnerability assessment tool is measured in two terms: False Positive and False Negative.

2.4.1 False Positive:

False Positive simply means that the result found by a vulnerability scanner is not confirmed and is just a false alarm. So reducing False Positive is a great need in a vulnerability scanner else lot of time of the user will be spent in cross-

checking/confirming the results shown by the scanner. In layman's words False Positive means reporting something that does not exist [20, 21, 22].

Consider a scenario where a vulnerability assessment tool has reported 5 XSS vulnerabilities and during close examination it comes to notice that that 4 of them do not exist. So eventually one will ignore the 5th report in this belief that it would also be a false alarm.

Example and common reasons of False Negatives:

- Wrong Logic used
- Another way has been established to avoid the vulnerability

2.4.2 False Negative:

False Negative is the inability of the Scanner in detecting an existence of a flaw in a system under consideration. False Negative can lead to system vulnerabilities hidden and make system still vulnerable. In layman`s terms, False Negative means ignoring a vulnerability that in actual exists [20].

Example and common reasons of False Negatives:

- Malicious file is packed (in case of antivirus), and antivirus scanner is not able to read the header.
- Scanner is updated but requires system restart for installation to complete.
- Newer Attack
- Signature obfuscation

2.5 Clickjacking

Clickjacking is a newly introduced web-based attack that has gained popularity in very short span of time. In this attack, a custom malicious web-page is designed to trick the victims to click on the desired elements of a page that is not visible to the victim. By tricking the victim to click anywhere an attacker wants, an attacker could force the user to perform any malicious action that is beneficial for the attacker (e.g., deleting account, changing-primary email id etc) [32].

2.5.1 Clickjacking Adverse Effects

Clickjacking is a newly introduced vulnerability in web applications which tricks users to click on hyperlinks that are not visible. In this way attacker can trick to perform any action he wants to. The victim can be tricked to purchase an item or to transfer funds.

Clickjacking Attack on Facebook

Clickjacking vulnerability was first seen on Facebook in 2010. The vulnerability was called LikeJacking as in that attack users were tricked to like a attacker`s chosen page. So in this way attackers were tricking Facebook users to grant permission to post content on their profile page. So whosever clicked on that like-jacked page gets infected and contents like porn video etc get posted on their page. Similar attack was also seen on twitter in which twitter users were tricked to give access to post on their behalf [23].

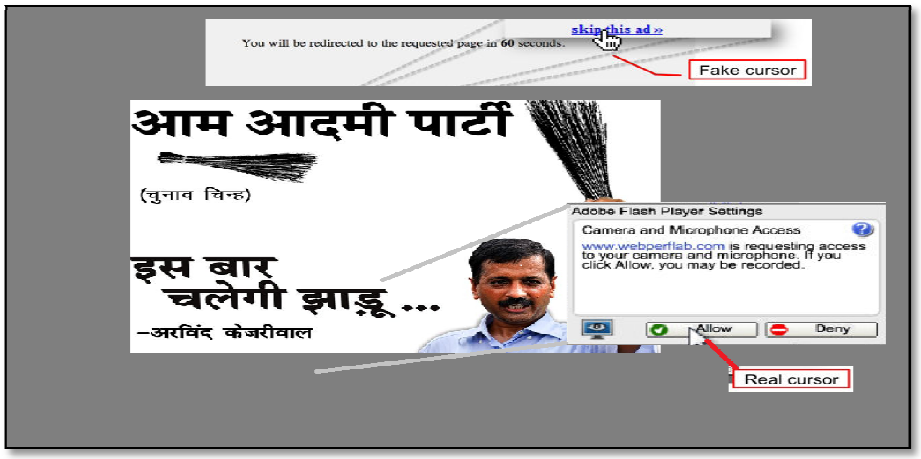

Figure 2.4 Clickjacking attack to hack webcam of victim.

Things Attacker can perform with this attack:

- Post objectionable contents on victim`s behalf
- Change passwords
- Change username
- Change Profile Pictures
- Perform Transactions
- Delete User account
- Webcam Hijacking

2.5.2 Frame Busting and Approaches used for bypassing traditional Clickjacking fixes:

Frame Busting is a most common approach used for dealing with Clickjacking. This technique works by using some way to make a particular website un-functional when used in I-frames. However this technique can easily be bypassed using various ways.
Frame busting is a use of some code and technique that can be a JavaScript code that uses some logic to prevent the site to be used in frames. This technique was supposed to be the most robust and recommended one for dealing with the frames attack.
Frame busting normally uses conditions for preventing attacks and these conditions can easily be bypassed [24].

Clickjacking attack can easily be mitigated by few lines of code in any language like python. This technique works by comparing the source of the content with the resources in the browser and if there is a mismatch, then page is not displayed [10].

2.5.3 Image based Authentication Security bypass by Clickjacking

If no counter measures are used for these framing attacks, then the Image based authentication which are used by many websites to ensure the authenticity of a website can easily be bypassed. Image based authentication is used by sites like yahoomail.com which use a Sign-In Seal that tells the website is authentic. But with the Clickjacking attack, one can use open the real yahoo login webpage in sub-frame and the Sign-In Seal is made visible to the user, so in this way user will easily fall in trap [15].

2.5.4 Clickjacking Protection in Mobile Websites

Most big websites have mobile websites that allows their website to be accessed on mobiles. These websites provide almost full access or a subsequent subset of their functionalities. Most of these websites don't have anything for dealing with the frames attack and thus Clickjacking attack can easily be performed on websites with sub domains like m.mobile.com [15].

2.5.5 Effective methods for Dealing with Frame Attacks

a) X-Frame Header Options: This protection was introduced by Microsoft and provides an effective way for dealing with the Clickjacking Attacks. X-Frame Headers are the special type of headers present in the response page of a website. If this header is present then the website is secure from the Clickjacking attack.

X-Frame Options can have the values DENY,ALLOW-FROM origin or SAMEORIGIN, which will prevent I-Frames to be placed and thus prevent framing from external websites.

- Deny: prevent any framing.
- SAMEORIGIN: prevent framing by external sites.
- ALLOWORIGIN: allow framing by specific sites

Limitation of X-Frame Header Options

- **Per-Page Enforcement:** This mitigation step can be imposed on one page at a time. This means for big websites with lot of pages these Security Step can be little cumbersome.
- **Web Proxies nullify X-Frame Header Options.** Web proxies are used for anonymously browsing internet but these Web proxies.

b) Using JavaScript: X-Frame Header Option will take time to get universally deployed. So till then the other effective way of dealing with the Clickjacking is the use of JavaScript. So if the page is framed then the javascript will block the page from loading and user will get the blank page. So it happens sometimes javascript are disabled in user`s web browser say the user is using no-script add-on, then the page user will see is a blank page.

Limitation of using JavaScript as Frame attacks mitigation technique:

- Javascript are the client side programming language and can easily be attacked so use of this technique is not recommended.

2.6 Open Web Application Security (OWASP) Top Attacks

OWASP stands for Open Web Application Security Project. It is a non profitable organization that is making continuous efforts for improving the web applications security. Every year OWASP comes with its OWASP TOP 10 Attacks Project for

creating awareness among internet users and web developers about the various web attacks, their severity and their mitigation steps. OWASP lists and ranks various Web Attacks on the basis of their criticality and popularity.

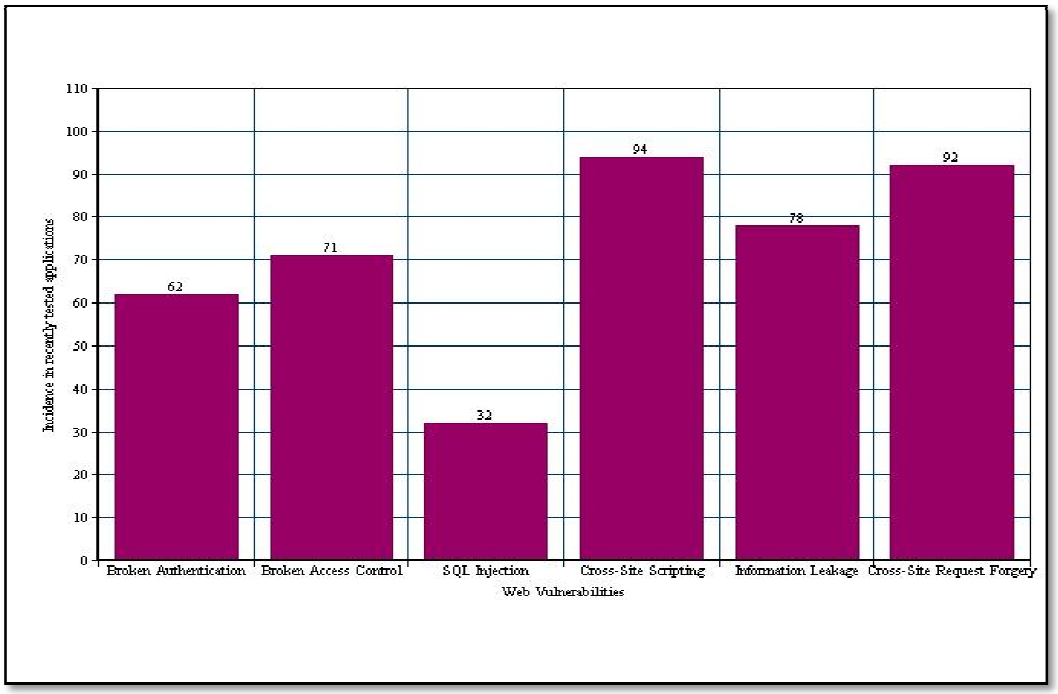

Figure 2.5 Web Attacks Distribution in Web Applications

Following categories of attacks has been created by OWASP ordered according to their severity:

1) Injection Attack

As a name suggests, injection attacks allows attackers to injection malicious code or string in to the query or URL or injecting shellcode which can lead to remote commands to be executed that lets the attacker to read databases ,execute system commands and change system data.

There are lots of well known injection attacks that can lead to full system compromise.

a) SQL Injection- SQL Injection is the most common and dangerous attack found in majority of websites. Website vulnerable with SQL Injection can let the attacker to read and steal the whole website database. It can also lead to website defamation and admin account takeover [25].

b) OS Command Injection – In this type of injection attacker can execute arbitrary operating system commands through web applications and can upload shell or can steal sensitive data and can steal passwords.
c) HTML Injection – Attacker injects HTML queries into the URL and thus can change the look of the website.
d) Buffer Overflow Attack- Attacker changes the flow of the web-application by modifying parts of the memory.

2) XSS (Cross Site Scripting)

XSS aka Cross Site Scripting is another common attacks found in most of the websites. In a survey conducted by OWASP, it was found that most of the websites are vulnerable to XSS. This security bug takes place due to improper input validation, allowing an attacker to inject malicious Java Script into a web application [4, 5, 16].Whenever a victim views the web page with the malicious script, this malicious JavaScript appears as a part of website only and is thus trusted and therefore, this malicious script also gets loaded and executed along with the webpage. As a result, the script can access and steal cookies , session IDs , record key logs, deface web pages ,install worms and virus and much more.XSS attacks are generally very simple to execute , but are difficult to prevent and can cause significant damage.

In XSS attacker uses Javascript as payload and using javascript he can performs malicious activities like

- session hijacking
- URL redirection
- Phishing
- Pawn Shell
- Website Defamation

XSS are of three main types:

a) Reflected XSS:

Reflected XSS also called as Non-Persistent XSS is the most common type of XSS found in majority of the web-applications. Reflected XSS are very easy to find and its target is the particular user. This type is called reflected because the malicious code

comes on the victim`s browser after being reflected by the server. This type of XSS can lead to attacks like session hijacking, phishing.

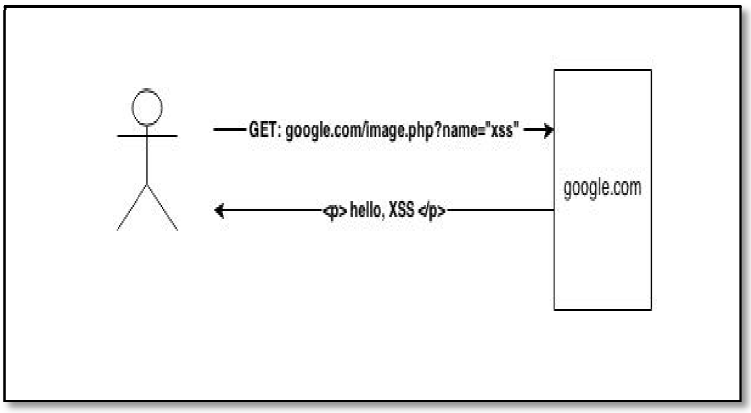

Figure 2.6 Reflected XSS.

b) Stored XSS:

This is another popular type of XSS, but more powerful and less common compared to the Reflected XSS. The power of this particular type of XSS can be imagined by knowing the fact that here the target is not a particular one or two users but the whole community who views that malicious webpage. Unlike Reflected XSS, malicious script is not reflected on the same time by the server but is stored as comment or something like that for later execution. This type of XSS is mainly found in forums, message boards that do not perform proper input validation. An attacker can post the message containing malicious script .This message gets stored and whosever views it gets infected.

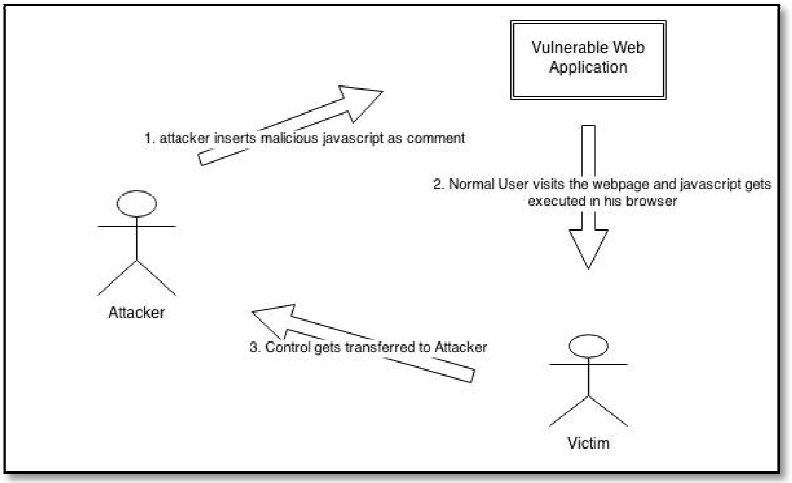

Figure 2.7 Stored XSS

c) DOM Based XSS:

Dom Based XSS is the rarest one and usually is not considered as a type of XSS. The reason for this is because of the fact that in this attack the malicious JavaScript code does not need to be sent to the server and this type of XSS is not due to any validation issues at the server side. Dom Based XSS is due to Document Object Model of the web site. It takes place when the JavaScript of the webpage is directly taking the input from the URL without any sanitization and is writing the input to the web-page [9].DOM based XSS takes place at the client side and is due to the validation issues on user's browser side. The resident JavaScript in the website do not validate the input provided by the attacker and thus this attack takes place [8].

3) Insecure Object Reference: This type of attack takes place due to improper and insufficient authentication and authorization check performed and thus can lead to low privilege account to access high privilege data. Example of this attack is User1 manage to get access to USER2 personal data.

4) Cross Site Request Forgery (CSRF or XSRF): This attacks works by stealing and changing account information of already authenticated user. In CSRF , the attacker sends a specially crafted webpage and through some social engineering he tricks the user to open that webpage and thus a request, which can be anything as per choice, is sent to server on victim`s behalf. It appears to server that original user is requesting for an information change but in actual attacker is taking advantage of this already authenticated session [27, 28]. This account can lead to:

- Password Change
- Username Change
- Primary Email Change
- Deleting Accounts

There exists various mitigation techniques for CSRF but most of them has limitations associated and can easily be bypassed and thus not recommended.

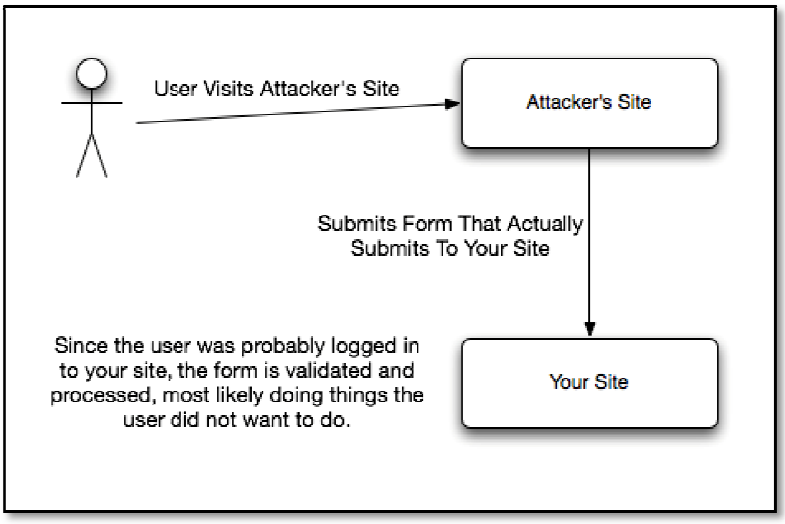

Figure 2.8 Cross-Site Request Forgery attack Flow.

Use of CSRF Anti-Tokens is the CSRF mitigation technique which provides complete protection against CSRF attacks.

5) Open URL Redirection: Open URL Redirection flaw in a web application can be used by attackers for phishing and scamming purposes. In this attack a website having this flaw can be user to redirect innocent victims to some malicious websites.
This type of flaw has huge impact on businesses and websites as it can lead to information stealing and also can lead to installation of malicious worm on victim`s system [29].

6) Broken Authentication & Session Management: As name suggests, due to weak authentication mechanism attacker is able to bypass the authentication system and get hold of the user accounts.

It also includes trying and guessing the username with different passwords using brute forcing. Special mechanism like Captcha has been used by website designers to mitigate these attacks but still there are ways with which these mitigations techniques can be bypassed [29].

2.7 Cross Site Scripting Protection Technique

Cross-Site Scripting Protection mechanism used can be broadly categorized in two ways [30]:

- Location of deployment of Cross-Site Check Mechanism(Client Side or Server Side)
- Type of analysis (Dynamic or Static)

2.7.1 Server Side Protection:

In Server Side Protection mechanism, the security check for Cross-Site Scripting is done at the server side and all the input validation check of all the un-trusted data is carried out on Server Side.

An anomaly based Intrusion Detection System is shown in [1] which is capable of detecting and preventing Cross Site Scripting attacks. It works by creating a log of secure Web Server which is Secure from all types of Cross-Site Scripting, both reflected and Persistent XSS. This log information contains useful data about the request made by different users to the secure Web Server and also contains information about the parameters requested along with the length of the parameters. So this Intrusion Detection System makes a profile according to the data provided and distinguishes between the legitimate request and the Cross-Site Scripting attack.

Another Example of dynamic server side XSS protection mechanism is the use of Perl Taint model described in [2]. This model checks for all the un-trusted input values through its Perl Interpreter. The attempts to use the un-trusted values directly or indirectly in the flow, which can lead to modification, deletion of directories, are considered as threat and are discarded. Similarly, PHP Taint Model is described in [3].

Apart from the dynamic Server Side techniques, there exist various Static Server Side Techniques for dealing with Cross Site Scripting.

2.7.2 Client Side Protection

In this approach, the check for XSS is performed at the client side i.e. on the User Browser itself. There exist various mechanisms in browser like Google Chrome which has its inbuilt XSS filter which prevent XSS from taking place.

In paper [4], it is described that the Cross Site Scripting occurs mainly due to improper input validation check done on the Server Side and therefore XSS can be mitigated by performing proper input validation check and also by using XSS filters. But this mitigation can take some time depending upon the developers, leading its

users exposed to this critical attack for some time until this vulnerability is not patched at the server end. A Client Side XSS protection mechanism is presented which ensures that the JavaScript can transfer information to the website from which it was loaded.

2.8 Adverse Effects of XSS Attack on the Internet

XSS vulnerability can be very dangerous. To many people, XSS appears to be just the presence of a Pop-Up box but in actual XSS vulnerability is very much more than this.

XSS is very serious and critical web vulnerability because of following reasons:

- This attack is very easy to conduct and don't need any extra knowledge.
- Majority of the websites are vulnerable to this vulnerability. According to the survey conducted in [5], more than 80% of websites in Japan was vulnerable to XSS.
- XSS vulnerability also gives scope for another critical Vulnerability like Cross-Site Request Forgery (CSRF). XSS helps an attacker to bypass the CSRF protection tokens.
- It becomes very difficult for the developer to locate the sinks and source of the XSS vulnerability.
- XSS filters can easily be bypassed using the various encoding schemes.

2.9 XSS Detection Approaches

Di Lucca, Giuseppe A. presents an approach of detecting XSS vulnerability in a website [6]. This approach performs both static testing as well as dynamic testing to check for the presence of XSS vulnerability. Static approach checks whether the web-page at server is vulnerable to XSS or not. Dynamic approach cross checks whether the web-page identified as vulnerable is in actual vulnerable to XSS or not.

Klein. has proposed a Server Side XSS detection System[7]. The approach used is based on monitoring the Http traffic passively. This system detects the presence of Reflected XSS as well as Stored XSS.

Summary: In this chapter, HTTP Protocol, its various methods and headers were discussed. White Box Testing and Black Box Testing are the two main approaches for Web Application Penetration Testing. Web Application Firewall (WAF) is an extra protection layer working at the application layer to protect web applications from various web attacks. Various WAF are available in market that offers different features. False Positives and False Negatives are the two crucial parameters that need to be kept minimum while designing web application vulnerability scanners. OWASP TOP 10 lists the various web attacks based upon their criticality and severity. OWASP also provides various mitigation steps for these web attacks. Clickjacking and Cross-Site Scripting (XSS) are the two critical web attacks that are very common in web applications and need to be addressed. Various techniques are available to mitigate them.

Chapter 3

Problem Statement

3.1 Gaps in Study

Web Applications now are everywhere. Web applications nowadays are used for business, entertainment, bill payment, shopping and much more. But because of the time constraints and high competition among the websites, very less stress is laid on the security aspect while developing the websites. So there is a huge need to encourage and educate the web developers and web administrators to consider the security aspect and the web attacks associated with the web applications. The security flaws will not only affect the business and reputation of the company but it can also lead to loss in terms of money as for example these web vulnerabilities can allow an attacker to transfer funds from one account to another. So there are very few Automated Vulnerability Assessment tools that are available in market that aims to analyze the websites for the presence of security vulnerabilities in the websites but the problem associated with these scanners are the accuracy and efficiency. These scanners has very high false positive and false negative ratio which eventually raises the burden on the shoulders of the web developer and the administrator only to confirm each and every hits that are detected by the Vulnerability Scanner.

3.2 Problem Statement

After studying the problems in the existing system, this conclusion has been derived that the already available vulnerability Scanners in the market are not able to detect all the vulnerabilities in a website. Also the False Positive rate, which are the fake alarms that is given by the scanner, are also very high which adds the additional burden on the administrator to re-check and confirm each and every hit. So the very first problem is to develop a tool that reduces the False Positive rate to minimum. Second problem is that due to lack of documentations and research on attacks, developers take some attacks like Cross-site Scripting very lightly but in reality XSS is one of the powerful web attacks. So this problem is solved by demonstrating all the possible attacks that can be carried out with the XSS.

3.3 Objectives

1. To develop a scanner capable of detecting web vulnerabilities with minimum false positives.
2. To compare the accuracy and performance of the developed system with the existing tools in market in terms of parameters like False Positives, False Negatives and Resource Consumption.
3. To demonstrate the various attack vectors that can be used for performing various possible attacks on a website once the XSS is detected

Chapter 4

Implementation

4.1 Automated Vulnerability Scanner (AVS):

Automated Vulnerability Scanner (AVS) is an open-source web vulnerability scanner that uses a black-box approach to crawl and scan web sites for the presence of exploitable and serious security vulnerabilities like SQL Injection, XSS, Clickjacking etc.

AVS has a flexible architecture that consists of multithreaded crawling, attack and analysis components. With the help of a graphical user interface the user can configure the attack vector, payloads, toggle between single or combined crawling and attack runs.

Currently AVS Scanner has support for following four attack components:

- XSS
- SQL Injection
- Clickjacking
- Banner Grabbing

4.1.1 Modes of AVS Scanner

To make the things easier for the novice web developers who are not much aware about how these vulnerabilities, two modes are included in this scanner:

Crawler Mode: Using this mode user just need to put the website name he/she want to analyze and the crawler which uses stack will scan all the WebPages in a given website and checks these stored WebPages one by one for the presence of vulnerabilities. In Crawler mode, scanning time depends upon the size of the website and it may vary from few minutes to few hours. The speed of this mode depends on factors like internet speed, Size of the website (no. of WebPages), System processing speed etc.

Direct Mode: In this mode user can analyze a specific webpage for the presence of security vulnerability. This mode, however, require more efforts and skill from the user as it need the user to put the URL in some specific format.

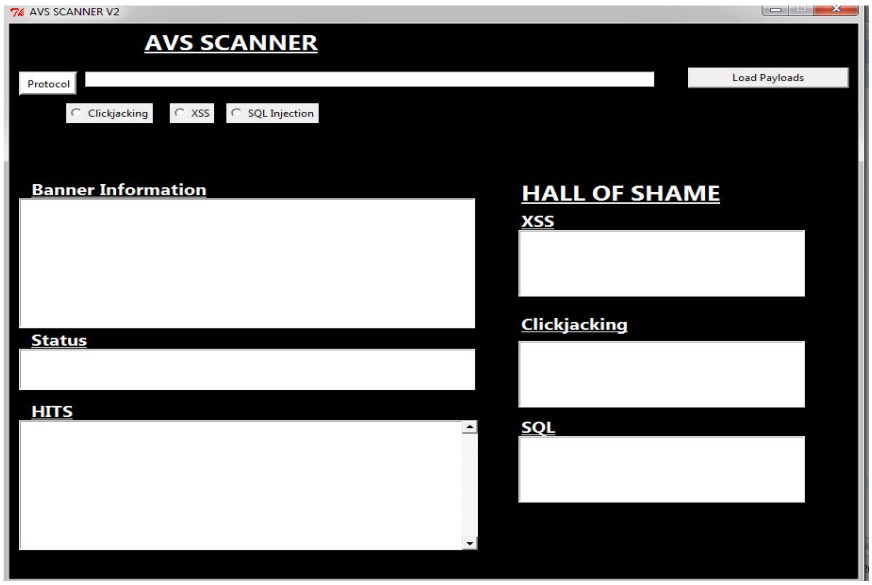

Figure 4.1 AVS Scanner

4.1.2 Attacks Supported and Logic Used

AVS SCANNER is capable of detecting following security vulnerabilities and following are the logic used by AVS SCANNER for improving the performance and accuracy.

4.1.2.1 XSS (Cross-Site Scripting):

The AVS Scanner scans for the reflected XSS vulnerability in websites. The logic used for detecting XSS vulnerability makes this tool powerful. Normally pentester get to know that a webpage is vulnerable to XSS if a pop-up box appears in the web-browser on providing the XSS payload.

So the above logic only is used for the detection of XSS. A web-browser is designed in python and is embedded in the AVS scanner application. So all the crawled web-pages are checked for the XSS on this browser itself. So if Pop Box appears then the website is vulnerable as shown in Figure 4.2. So this means that there is no chance of any False-Positive to occur. Thus all the results detected by this scanner are confirmed.

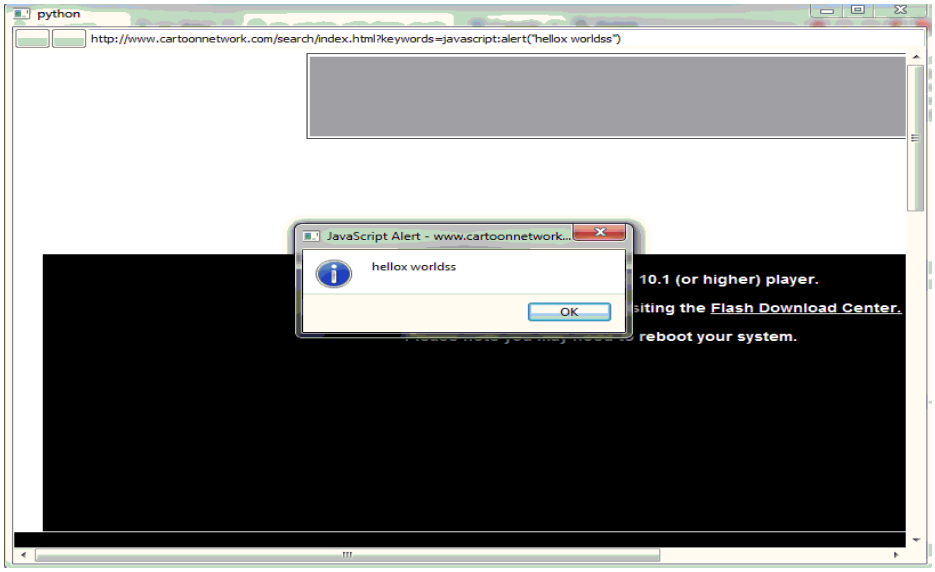

Figure 4.2 AVS Browser for reducing FALSE POSITIVIES.

AVS Scanner is even capable of bypassing Web Application Firewall (WAF) using its own built-in XSS payload. More the payloads more are the chances to evade the XSS filters. AVS SCANNER has a list of more than 1500+ specially crafted XSS payloads. These payloads are enough to bypass almost any weak XSS bypass-filters applied by the developer. This number is more than any existing Vulnerability Scanner in the market. Figure 4.3 show that AVS SCANNER has highest no. of XSS payloads when compared with the existing scanners available in the market.

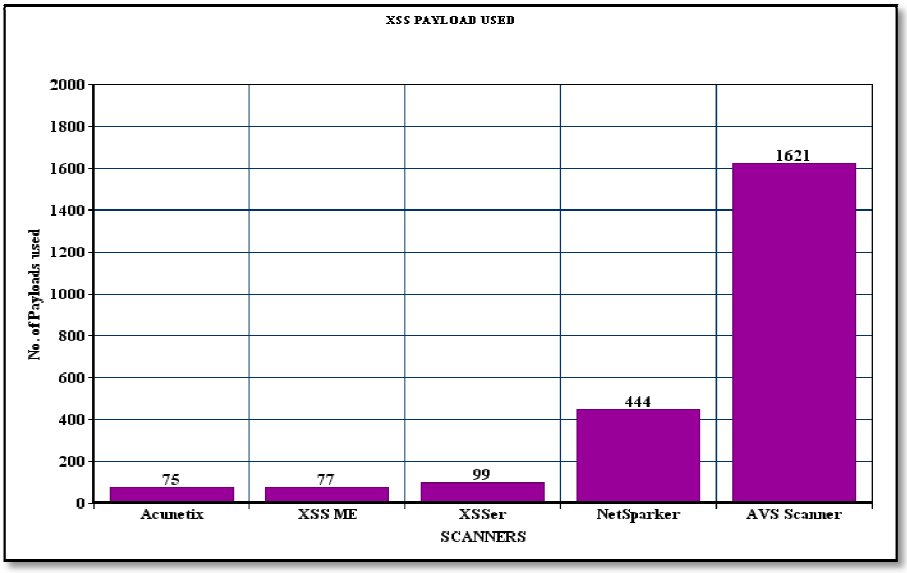

Figure 4.3 XSS Payload count in different Vulnerability Scanners.

To improve the accuracy, AVS double checks the URL for the presence of XSS vulnerability with another simple logic. This logic is that whenever a XSS payload is sent as a request if that payload comes back as it is without any alteration in response then that webpage is vulnerable to XSS. So this makes this XSS scanner part very effective. More the payloads more are the chances to evade the XSS filters.

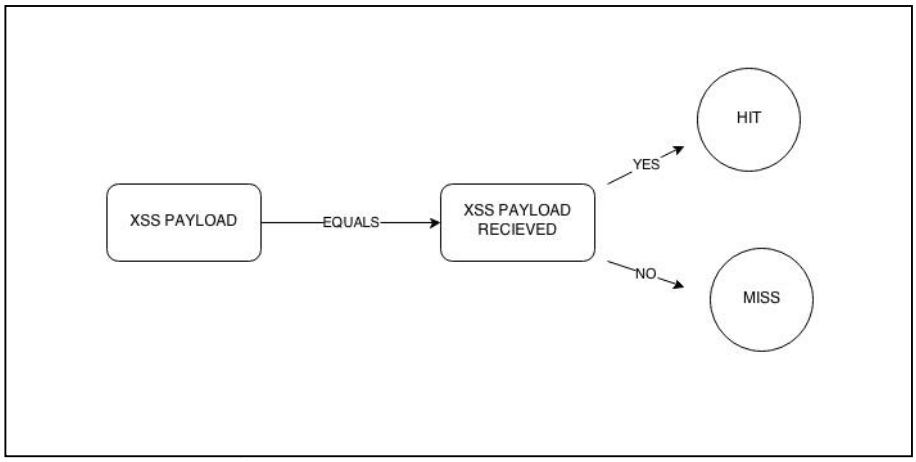

Figure 4.4 XSS Second Logic. If XSS payload sent equals XSS payload received then the site is vulnerable to XSS.

4.1.2.2 CLICKJACKING:

For checking the Clickjacking Vulnerability in a website, AVS SCANNER tool analyse the response HTTP headers and check for the presence of X-Frame-Options in it and if this option is there, then the website is secure against this attack as demonstrated in Figure 4.5.

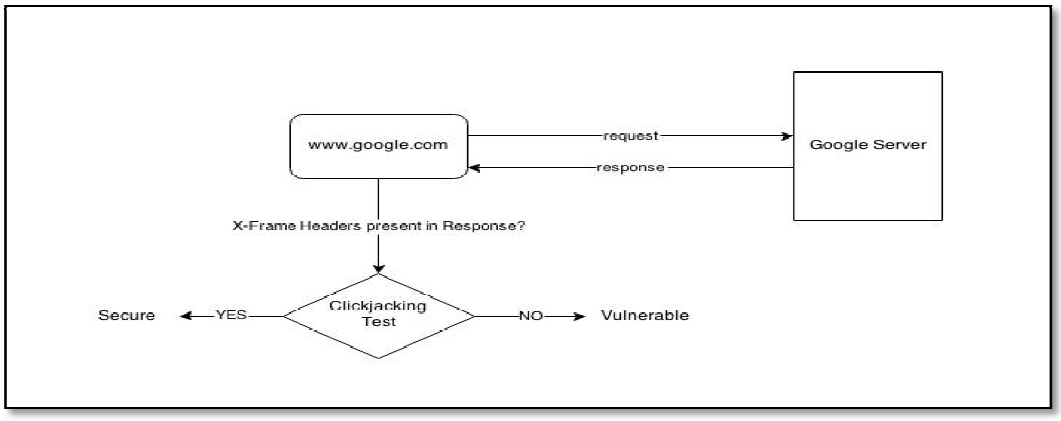

Figure 4.5 Clickjacking Logic: This figure shows if X-Frame Headers are missing in response, then site is vulnerable to clickjacking.

X-Frame Options can have the values DENY,ALLOW-FROM origin or SAMEORIGIN, which will prevent I-Frames to be placed and thus prevent framing from external websites.

- Deny: prevent any framing.
- SAMEORIGIN: prevent framing by external sites.
- ALLOWORIGIN: allow framing by specific sites.

4.1.2.3 SQL INJECTION*:*

The Logic used for this SQL injection is very simple and our AVS tool right now checks for the Integer based SQL-Injection only.AVS Scanner simulates all the steps of the Integer based SQL-Injection for detecting the SQL Injection Vulnerability in a website.

4.2 Hall of Shame:

This is a special section added in this scanner. This section shows which among the entire list of websites that has been scanned by the AVS Scanner, is most vulnerable. All the websites that have been scanned once in this scanner, their result gets stored in the hard disk in a text format file and those results are used to rank and categorize the website on the basis of the total vulnerabilities found.

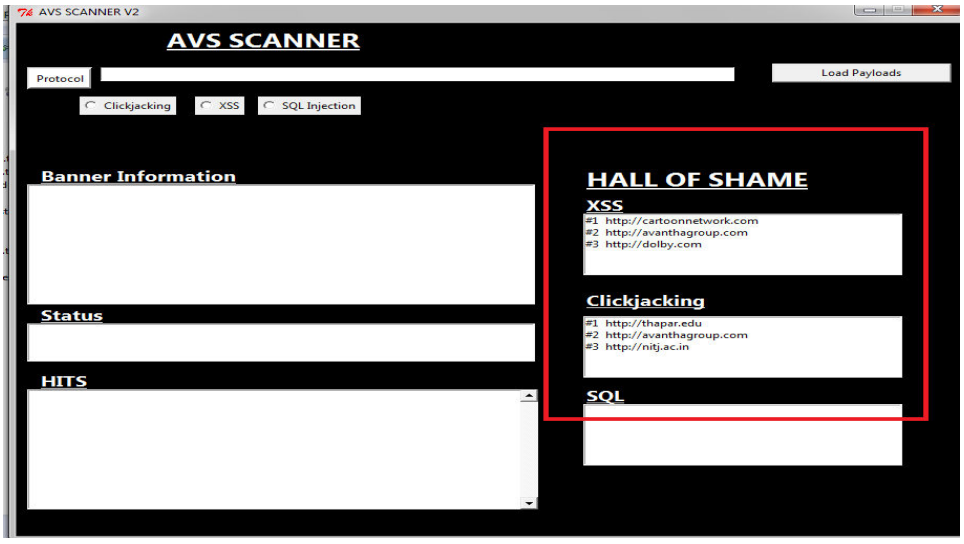

Figure 4.6 AVS Scanner Hall of Shame Section

4.3 Banner Grabbing and Information Gathering:

The AVS Scanner provides support for extracting the banner of the web-server on which the website is hosted. AVS Scanner tries to extract various important information about the website like:

- Server Details- which include the type of server being used, which Operating System Web Server is using and Version of the Web Server Software.
- Version Disclosure- this tells about which technology is being used for the website.
- Website Ranking- this part tells about the popularity of the website. The Global Ranking and the Local Ranking of the website is first extracted from Alexa.com and then is displayed.

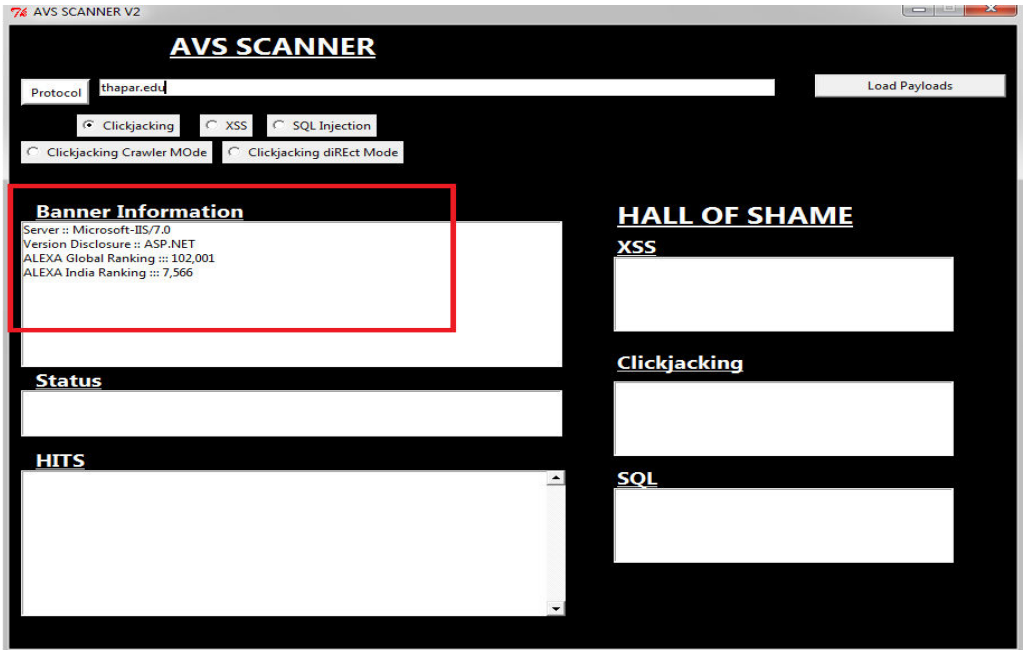

Figure 4.7 AVS Banner Grabbing of website.

4.4 PYTHON: PYTHON is the language behind this Proposed System. Python is as fast and most appropriate language for making pentesting tools. It is used in variety of application domains. Some of the distinguishing features of Python are [31]:

- It involves very clear and readable syntax.
- Python is an open source language.

- Like other object-oriented languages, Python also have object oriented features.
- It has mechanism defined to deal with run-time errors with its exception handling feature.
- It includes very high level dynamic data types.
- Python Code is platform independent and can be run on any O.S without modifying the code.
- It has extensive set of standard libraries and third party modules for almost every task.
- It can be embedded within other programming languages like php and applications as a scripting interface.

4.4.1 Python Modules Used:

Following modules are used in the implementation of the project:

- Tkinter: This module is used for providing Graphical User Interface to our Scanner.
- OS: This module as the name suggest is the Operating System module and is used for executing operating system commands within python environment. Some of its common functions are: chdir()-for changing the directory, getcwd()-for getting the complete path of current working directory etc.
- subprocess: This module is used for executing system commands. This module also generates status code of the command it executes so it help in finding whether command has been executed successfully or not.
- random: This module is used for generating random values within a given defined range.
- TkFont: this modules deal with the font of the labels used in the GUI.
- TkFileDialog: This module is for creating a Graphical window for browsing some file in system`s directory.
- Mechanize: This module is used for simulating a web browser in python. It creates a virtual browser that works within a python interpreter. So everything that one can perform in any standard browser like firefox,chrome can be performed within python using this module.

- Urllib: This is another module that helps our AVS Scanner to interact with the internet.
- BeautifulSoup: This module is a HTML web parser that helps in web-scrapping. This module helps user to extract important information like title of the webpage etc that is required by the user from the webpage.
- Urlparse: This module is used for breaking the URL in different components and then re-assembling them to generate some absolute URL from the base URL.
- Sys: Sys module is the System module used in python for tasks like getting command line arguments etc.
- PyQt4: This is another module for adding Graphical User Interface in the project.
- Thread: Thread module is the very important module responsible for adding concurrency and parallelism to our Scanner.

Chapter 5

Experimental Results

5.1 Comparison of AVS SCANNER with other Scanners: AVS SCANNER in Real World

In order to check the accuracy of AVS SCANNER and also to show how this scanner performs and what advantages this Scanner has to offer over other existing Vulnerability Scanners available in the market, a comparison test on few websites has been performed and checked which tool is better on the basis of following parameters:

a) False Positive: False Positive simply means that the result found by a vulnerability scanner is not confirmed and is just a false alarm. So reducing False Positive is a great need in a vulnerability scanner else lot of time of the user will be spent in cross-checking/confirming the results shown by the scanner.

b) False Negative: False Negative is the inability of the Scanner in detecting an existence of a flaw in a system under consideration. False Negative can lead to system vulnerabilities hidden and make system still vulnerable.

c) Resource Utilization: Resource utilization parameter means how harsh the Scanner is on the system on which it is running and how much resources it is consuming. With resource consumption actually means CPU Consumption and RAM Consumption. So Scanners should not use all the resources of the system and thus avoid throttling other applications.

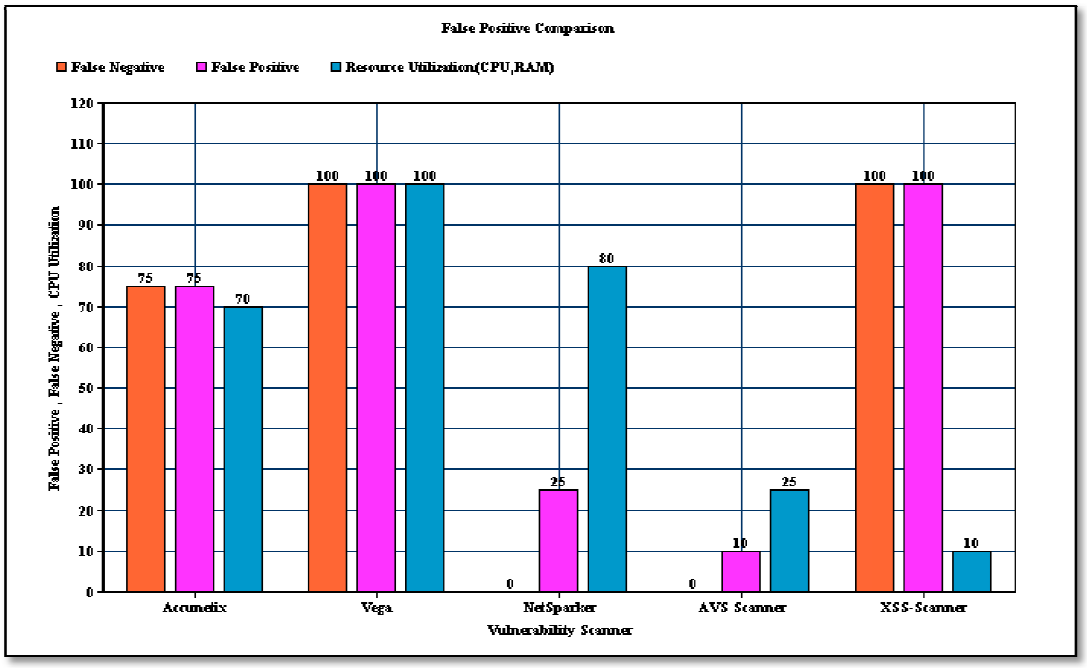

Figure 5.1 False Positives, False Negatives, Resource Utilization Comparison

The comparison Chart in Figure 5.1 clearly shows that AVS Scanner outperforms all other existing Vulnerability Scanner in term of False Negative, False Positive and Resource Consumption. AVS Scanner is very gentle on system in term of Resource Consumption and also it gives minimum False Positive which made it better in performance compared to other existing tools.

All the websites that were scanned in the comparison test for the presence of security vulnerabilities includes some High-Profile websites like Yahoo, HelpScout.com, EngineYard.com, Thapar University, NIT Calicut, NIT Kurukshetra, BSNL and many more. Following are the few websites which were analyzed using AVS Scanner and in which vulnerabilities were found and reported.

Table 5.1 AVS Scanner Performance in Real world

Website	Vulnerability	Status
www.flickr.com	Clickjacking, XSS	Patched
www.tumblr.com	Clickjacking, XSS	Patched
www.nitkkr.ac.in	XSS, Clickjacking, SQL-Injection	Patched

www.nitc.ac.in	Clickjacking, XSS, SQL-Injection	Patched
www.paymill.com	Clickjacking, XSS	Patched
www.magix.com	Clickjacking, XSS	Patched
www.bsnl.co.in	XSS, SQL-Injection, Clickjacking	Un-patched
www.statuspage.io	Clickjacking	Patched
www.engineyard.com	Clickjacking, XSS	Patched
www.shaukk.com	Clickjacking, XSS	Patched

All these websites were the part of bug-bounty program and all their vulnerabilities were responsibly reported to their respective owner.

5.2 XSS EXPLOITATION PAYLOADS

Most Web Pentester knows that XSS vulnerability exists in a web application when a pop-up box appears but majority of them don't know how to exploit this vulnerability. So to show what an attacker can do with this vulnerability, Proof of Concept has been created.

So following are the malicious activities that an attacker can perform with the XSS:

5.2.1 Web Defacement

XSS can be used for defacing the look of the website. This can be achieved in both the Reflected as well as Stored XSS. However the defacement is not permanent in Case of the Reflected XSS and the defacement is for a specific user only.

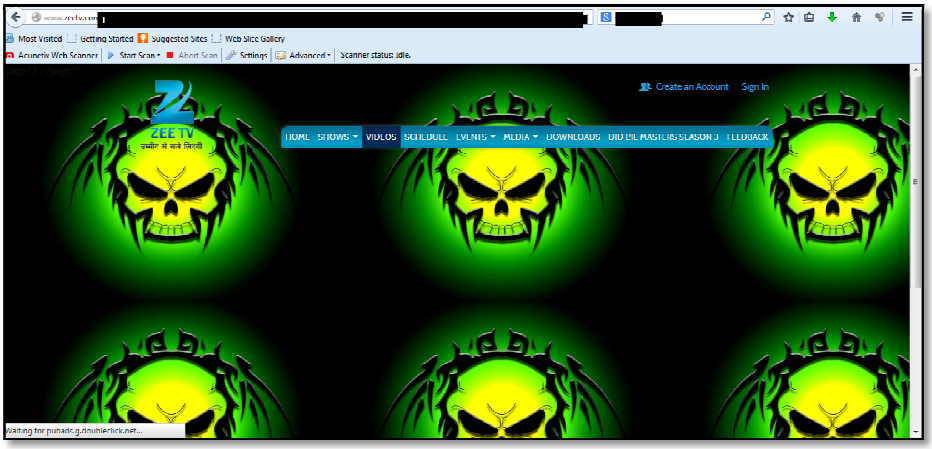

Figure 5.2 XSS Website Defacement

Attacker can take advantage of this by modifying the look of the webpage for conducting the phishing attack for stealing the credentials.

5.2.2 URL Redirection

XSS can be used in URL Redirection. So with this an attacker can easily redirect the innocent victim to some malicious website and can perform further attacks.

To demonstrate this, helpscout.com is taken as an example. The website www.helpscout.com was vulnerable to Persistent XSS. So a XSS Payload that redirects the users to google.com was chosen and posted as comment. So this payload can redirect innocent users to some malicious website.

So on visiting the following link **https://www.helpscout.net/blog/insincere-support/** the user gets redirect to www.google.com.

Payload Used:

<script>alert("URL REDIRECTION VIA XSS");document.location=
http://google.com;</script>

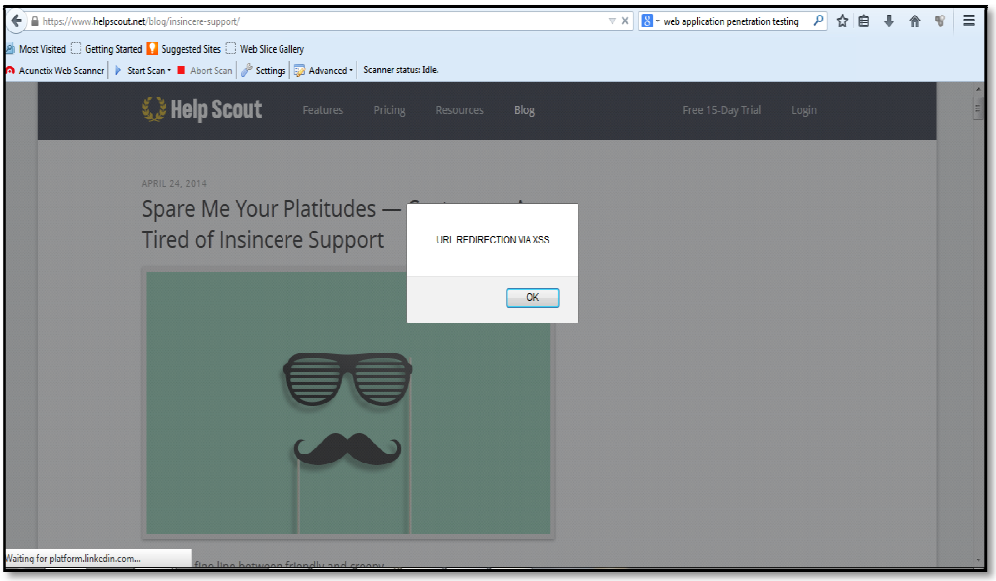

Figure 5.3 XSS URL Redirection.

5.2.3 Injecting Trojan Functionality

The website defacement through XSS is temporary but using XSS an attacker can do more malicious tasks like stealing Credentials, Passwords, Credit Card Information and much more. Using XSS vulnerability an attacker can inject actual malicious

functionalities in the web application. Attacker can insert some fake login form of any website like facebook.com and can easily deceive a user to steal its credentials.

The above attack is explained with the real life example on couple of websites like www.israpost.com and a bank website whose identity is kept secret for security reasons.

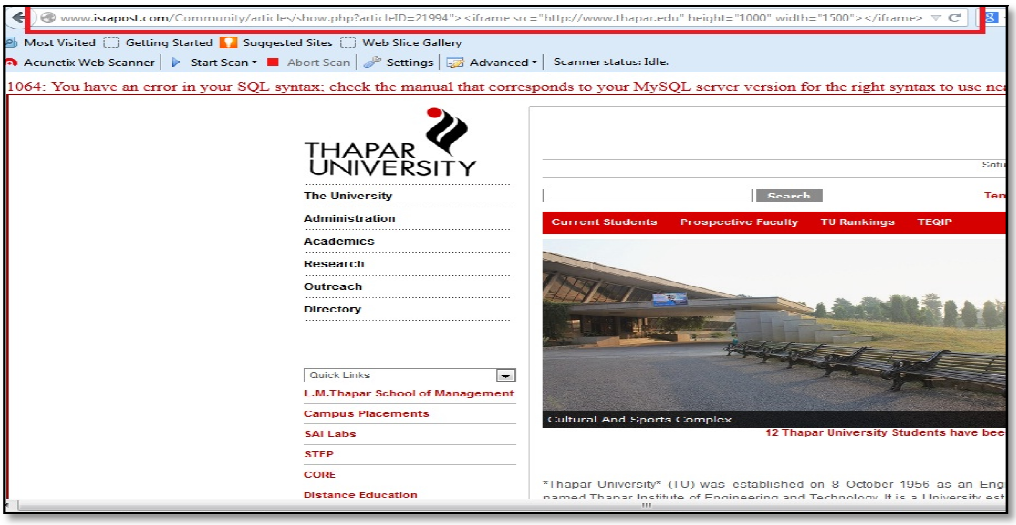

Figure 5.4 XSS Trojan Functionality.

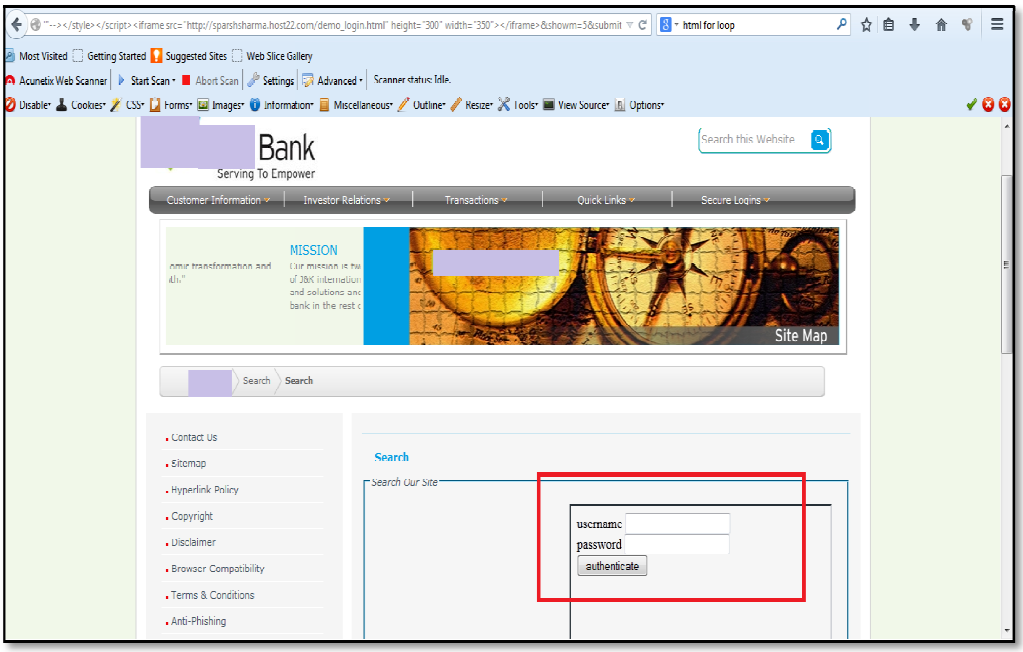

Figure 5.5 XSS Fake Form on a Bank Website

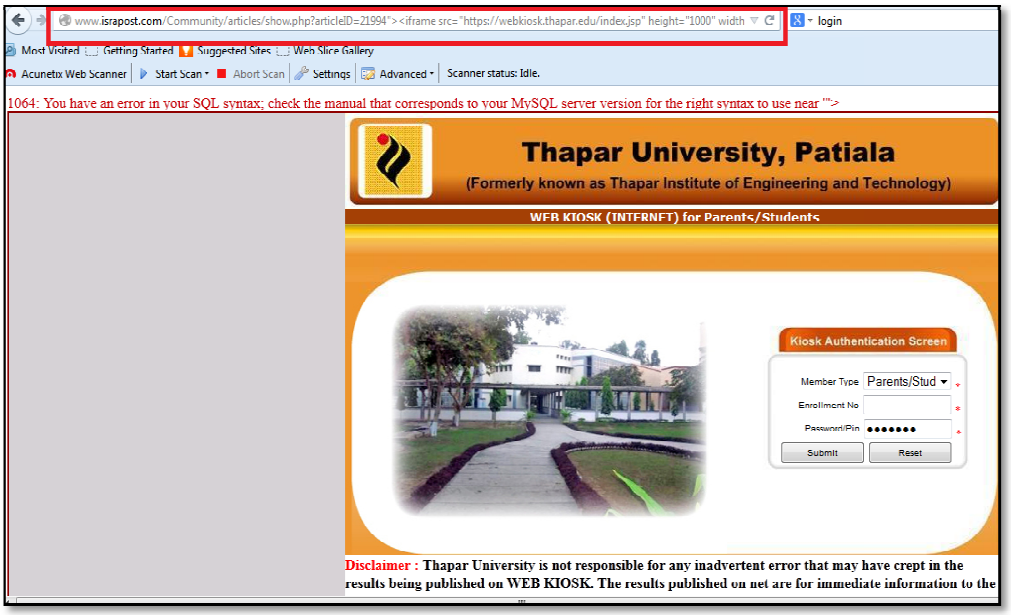

Figure 5.6 XSS Trojan Functionality in Thapar.

As shown in above Figures 5.4 and 5.6, the website israpost.com was vulnerable to XSS and using iframes tags Thapar`s main website was embedded in israpost.com`s website and also the student login webpage of Thapar was inserted into the webpage of www.isropost.com. So in this way an attacker can trick the innocent user because to the user it appears that thapar website is opening inside the www.isropost.com as the URL is of www.isropost.com and thus can easily trust and will fall into the trap.

One can imagine how worse this attack can be if some Bank website is vulnerable to XSS. Then this attack vector can be used to create a fake login page on that vulnerable bank`s website and eventually stealing the user`s credentials as demonstrated in Figure 5.5.

5.2.4 Cookie Stealing

Cookie Stealing is the attack one can perform on the website vulnerable to XSS. An attacker can hijack the user`s session by stealing the user cookie who is already authenticated and thus can access the user account without the need of any password.

Cookie Stealing attack via XSS can be carried out using the following Payload:

<script>alert(document.cookie)</script>

Also a more advance Payload can be inserted that can send the user`s cookie to some attacker defined server. The Cookie Stealing attack was performed on www.avanthagroup.com which was also vulnerable to XSS.

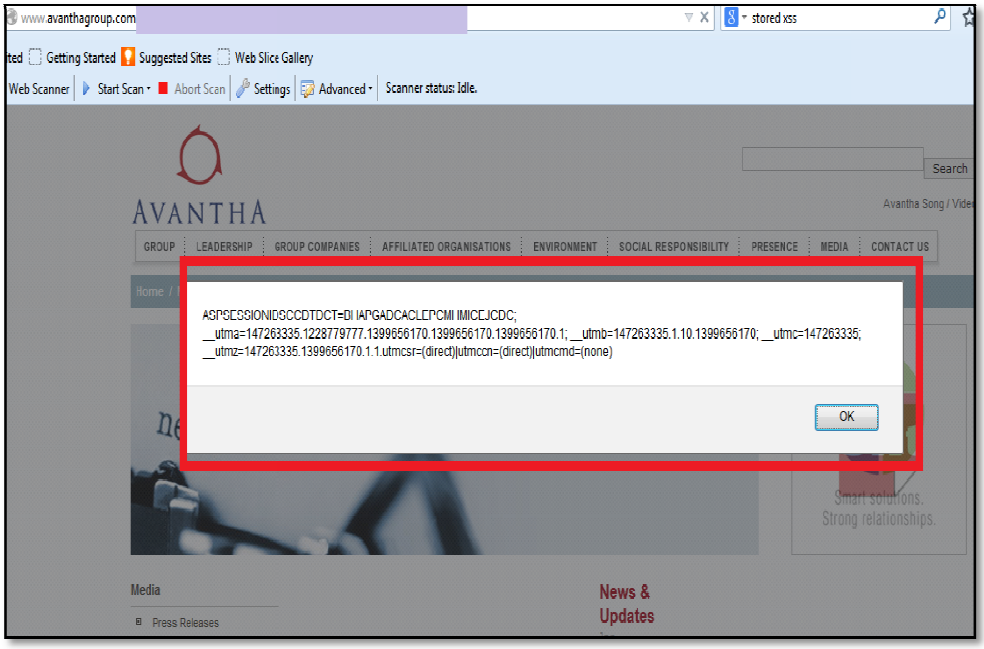

Figure 5.7 XSS Cookie Stealing on avanthagroup.com.

5.2.5 XSS PORT Scanning

Once XSS vulnerability is found then it is possible to do port scanning also. Port Scanning is the very first step of information gathering an attacker follows. Port Scanning checks for the open ports on target machine which can be used as an entry point. But while doing port scanning attacker don't want to get caught so he need to hide himself behind wall via techniques like VPN, Proxies etc. But attacker can use a XSS vulnerable website to do the port scanning of the desired target and the best part of this attack is that to the victim it will appear that the XSS vulnerable website is trying to do port scanning.

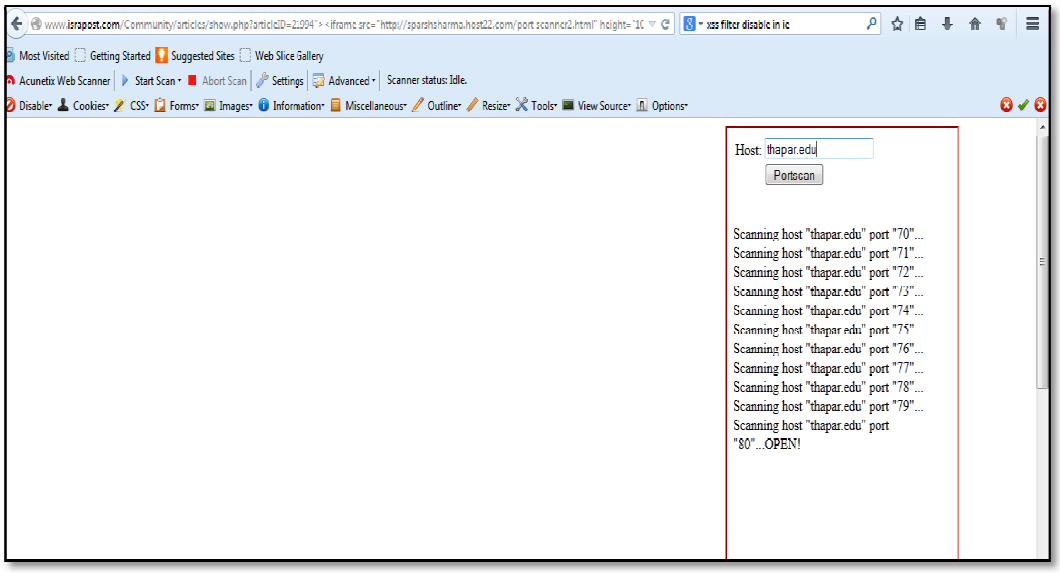

Figure 5.8 XSS Port Scanning

Vulnerable URL :

http://www.israpost.com/Community/articles/show.php?articleID=21994”><iframe src=http://sparshsharma.host22.com/port_scanner2.html height=”1000” width=”1500”></iframe>

PAYLOAD USED:

<iframe src=http://sparshsharma.host22.com/port_scanner2.html height=”1000” width=”1500”></iframe>

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Web Application now have become lifelines for all of us and thus Security issues associated with these web applications are also emerging along with it. So there is a huge need of dealing with these security issues in order to have a secure World Wide Web. If not patched these vulnerabilities can affect the three pillars of the Security, i.e. Confidentiality, Integrity and Availability. It is very easy for the attackers to find and exploit these security vulnerabilities present in the web-applications and majority of websites are still lacking security patches and thus are vulnerable. Dealing with these security issues require to educate the users about all the safe practice they need to follow when working online and also to web developers regarding how to code securely and creating awareness among them regarding different possible web attacks and also there mitigation techniques.

So a web administrator is required to follow all the security practices and keep their web server, database server software's up-to-date. Vulnerability Scanner can be an ideal and simplest option for the web-developers to analyze their websites for the presence of Security vulnerabilities quickly and efficiently. These scanners can also be used for malicious purposes (just like Hping, Nmap[4]) by hackers, so we hope this scanners are used by web developers and website owners to audit the security of their website.

6.2 Future Scope

For making this tool more powerful, support for more serious web attacks like Cross Site Request Forgery (CSRF), Cross Site Port Attack (XSPA) etc will be included in the future version of AVS Scanner.

Also web application is a crucial field and normally deals with the critical data, so attackers keep on trying to find some newer ways to steal and own the Web applications. So this tool need to be kept updated to deal with all the Zero Days attacks that are emerging.

Also the current version of AVS Scanner shows only the vulnerability in a particular website. So besides just detecting the vulnerabilities in a given website, features will be added in next version of AVS Scanner that will allow AVS to demonstrate how a found vulnerability in a website can be exploited and how it can be used by hacker if not patched.

REFERENCES

[1] C. Kruegel and G. Vigna, “Anomaly Detection of Web-based Attacks”, 10th ACM Conference on Computer and Communication Security (CCS-03) Washington, DC, USA, October 27-31, pp no. 251 – 261, 2003.

[2] J. Allen. Perl Version 5.8.8 Documentation – Perlsec [Online]. Available: http://perldoc.perl.org/perlsec.pdf,2006

[3] A. Nguyen-Tuong, S. Guarnieri, D. Greene, J. Shirley,and D. Evans., “Automatically Hardening Web Applications Using Precise Tainting”, 20th IFIP International Information Security Conference, Makuhari-Messe, Chiba, Japan, 2005.

[4] Spett, Kevin, "Cross-site scripting." *SPI Labs* (2005): 1-20.

[5] Kazuhito Ohmaki,“Open Source Software Research Activities in AIST towards Secure Open Systems”, Proceedings of the 7th IEEE International Symposium on High Assurance Systems Engineering, p.37, October 23-25, 2002.

[6] Di Lucca, Giuseppe A., et al., "Identifying cross site scripting vulnerabilities in web applications.", 26th INTELEC International Telecommunications Energy Conference ,IEEE, 2004.

[7] Klein., “Cross site scripting explained”, White Paper, Sanctum Security Group, http://crypto.stanford.edu/cs155/CSS.pdf, June 2002

[8] Grossman, Jeremiah. “XSS Attacks: Cross-site scripting exploits and defense”, Syngress, 2007.

[9] Kirda, Engin, et al. "Client-side cross-site scripting protection.", Computers & Security 28.7 (2009): 592-604.

[10] Balduzzi, Marco, et al. "A solution for the automated detection of clickjacking attacks.", Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. ACM, 2010.

[11] Mogul, Jeffrey C. "Clarifying the fundamentals of HTTP.", Software: Practice and Experience 34.2 (2004): 103-134.

[12] Tomar, Nishi, and Manoj Singh Gaur. "Information theft through covert channel by exploiting HTTP post method.", Tenth International Conference on Wireless and Optical Communications Networks (WOCN), IEEE, 2013.

[13] Chandran, Rajalaxmi, and Sathiamoorthy Manoharan, "Performance analysis of New Zealand websites using HTTP header values." Communications, Computers and Signal Processing (PacRim), 2011 IEEE Pacific Rim Conference on. IEEE, 2011.

[14] Berners-Lee, Tim, Robert Cailliau, and Ari Luotonen. "Henrik Frystyk Nielsen, and Arthur Secret." *The world-wide web. Communications of the ACM* 37.8 (1994): 76-82.

[15] K. Scarfone, M. Souppaya, A. Cody and A.Orebaugh, "Technical Guide to Information Security Testing and Assessment", National Institute of Standards and Technology, pp. 36-39, Sep. 2008.

[16] Johns, Martin, Björn Engelmann, and Joachim Posegga. "Xssds: Server-side detection of cross-site scripting attacks.", Computer Security Applications Conference (ACSAC), 2008. Annual. IEEE, 2008.

[17] Razzaq, Abdul, et al. "Critical analysis on web application firewall solutions." Autonomous Decentralized Systems (ISADS), 2013 IEEE Eleventh International Symposium on. IEEE, 2013.

[19] Bau, Jason, et al. "State of the art: Automated black-box web application vulnerability testing.", *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010.

[20] Vulnerability Assessment Accuracy [Online]. Available:http://www.beyondsecurity.com/va_accuracy_false_positive_negative.html

[21] P. G. Spathoulas, and S. K. Katsikas. "Reducing false positives in intrusion detection systems" Computers & Security, Vol. 29, no.1, pp no. 35-44, 2010.

[22] Víctor, Ganta Jacob, Sreenivasa Rao Meda, and V. C. H. Venkaiah. "False Positives in Intrusion Detection Systems."

[23] Bradbury, Danny. "Spreading fear on Facebook.", Network Security 2012.10 (2012): 15-17.

[24] Rydstedt, Gustav, et al. "Busting frame busting: a study of clickjacking vulnerabilities at popular sites.", IEEE Oakland Web 2 (2010): 6.

[25] Sadeghian, Amirmohammad, Mazdak Zamani, and Suhaimi Ibrahim. "SQL Injection Is Still Alive: A Study on SQL Injection Signature Evasion Techniques.", Informatics and Creative Multimedia (ICICM), 2013 International Conference on. IEEE, 2013.

[26] Zanero, Stefano, Luca Carettoni, and Manuel Zanchetta. "Automatic Detection of Web Application Security Flaws." Black Hat Briefings (2005).

[27] Alexenko, Tatiana, et al. "Cross-site request forgery: attack and defense." 7th Consumer Communications and Networking Conference (CCNC), IEEE, 2010.

[28] De Ryck, Philippe, et al. "Automatic and precise client-side protection against CSRF attacks." Computer Security–ESORICS 2011. Springer Berlin Heidelberg, 2011. 100-116.

[29] Stuttard, Dafydd, and Marcus Pinto., "The web application hacker's handbook: discovering and exploiting security flaws", John Wiley & Sons, 2008.

[30] Vogt, Philipp, Florian Nentwich, Nenad Jovanovic, Engin Kirda, Christopher Kruegel, and Giovanni Vigna. "Cross Site Scripting Prevention with Dynamic Data Tainting and Static Analysis.", NDSS. 2007.

[31] Seitz, Justin., "Gray Hat Python: Python programming for hackers and reverse engineers", No Starch Press, 2009.

[32] Shahriar, Hossain, Vamshee Krishna Devendran, and Hisham Haddad. "ProClick: a framework for testing clickjacking attacks in web applications." Proceedings of the 6th International Conference on Security of Information and Networks. ACM, 2013.

[33] Benefits of Web Based Applications [Online]. Available:http://www.dbnetsolutions.co.uk/Articles/BenefitsOfWebBasedApplications.aspx

[34] Peltsverger, Svetlana, and G. Zheng., "A Virtual Environment for Teaching Technical Aspects of Privacy.", Proceedings of the 2013 on InfoSecCD'13: Information Security Curriculum Development Conference. ACM, 2013.

[35] McRee, Russ. "OWASP Top Ten Tools and Tactics." (2012).

List of Publications

- S. Sharma and M. Singh, "AVS SCANNER: a Black Box Vulnerability Scanner with Zero False Positive", Peer reviewed and accepted for publication in ELSEVIER International Conference on Emerging Research in Computing, Information, Communication and Applications, ERCICA 2014,Aug 01-02, 2014, Bangalore, India.