MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

SUMY STATE UNIVERSITY

Faculty of Electronics and Information Technology

Division IH 85Aan

"INFORMATION PROTECTION TECHNOLOGY"

DETECTION AND PREVENTION OF VULNERABILITY XSS

Applicant Jackrich Lords TS

Scientific adviser, Ph.D., Assoc. Olena Protsenko

Head of the Department, Professor Anatoliy Mykolayovych Dovbysh Doctor of Medical Sciences

TABLE OF CONTENTS

ABSTRACT

Note: 41 pages, 18 figures, 1 appendix, 22 reference sources .

Object of study – Informational web system for online shopping

Purpose - To design and implement an online shopping store using modern client technologies

Results - a web based online shopping store have been designed and implemented using modern client technologies to meet the requirement of the project.

JAVASCRIPT, HTML, CSS, PHP

INTRODUCTION

Nowadays, it’s a ways greater accurate to think about websites as online programs that execute a number of features, instead of the static pages of old. A good deal of this robust capability is due to good sized use of the JavaScript programming language. While JavaScript does allow websites to do some pretty cool stuff, it additionally offers new and specific vulnerabilities with Cross site scripting (XSS) being one of the maximum extensive threats.

It's far one of the maximum popular and susceptible assaults which is known through each superior tester. It's miles taken into consideration one of the riskiest assaults for net programs and can carry harmful consequences too.

XSS is regularly compared with comparable client-side assaults, as client-aspect languages are broadly speaking getting used throughout this attack. However, an XSS assault is considered riskier, due to its potential to damage even less susceptible technologies.

1 INFORMATION REVIEW

Cross-site scripting, typically known as XSS, takes place while hackers execute malicious JavaScript within a victim's browser. The malicious script can be saved at the webserver and performed each time whilst the person calls the suitable capability (the awful actor attaches their malicious code on pinnacle of a legitimate website, essentially tricking browsers into executing their malware on every occasion the website is loaded). It could also be completed with the opposite strategies with none saved script in the webserver. Unlike Remote Code Execution (RCE) attacks, the code is administered inside a person's browser.

The principle purpose of this assault is to steal the opposite consumer's identification statistics – cookies, session tokens, and other statistics. In maximum cases, this assault is getting used to scouse borrow the other man or woman‘s cookies. As we realize, cookies help us to log in routinely. Consequently with stolen cookies, we can log in with the other identities. And that is one of the reasons, why this attack is considered one of the riskiest attacks. As opposed to concentrated on the application's host itself, XSS attacks generally target the utility's customers directly. Companies and companies strolling net applications can leave the door open for XSS assaults in the event that they display content material from customers or untrusted resources without right escaping or validation.

An XSS attack is being accomplished on the customer-facet. It is able to be done with unique customer-aspect programming languages. However, most often this assault is finished with Javascript and HTML. So by writing secure code, checking out for vulnerabilities, and operating with protection equipment like Veracode Dynamic evaluation, builders can prevent, come across, and restore capacity vulnerabilities making an allowance for XSS exploitation.

1.1 Cross-site scripting attack

When attackers inject their personal code into an internet web page, generally executed by exploiting a vulnerability on the website's software, they can then inject their personal script, that's performed with the aid of the sufferer's browser.

Since the JavaScript runs on the sufferer's browser page, sensitive details about the authenticated person may be stolen from the session, basically allowing a terrible actor to target web site directors and absolutely compromise an internet site.

Some other famous use of pass-web site scripting attacks are whilst the vulnerability is to be had on maximum publicly available pages of an internet site. In this situation, attackers can inject their code to goal the site visitors of the internet site by using including their personal commercials, phishing prompts, or different malicious content. If the hunt discipline is vulnerable, while the consumer enters any script, then it'll be accomplished.

Let's look at an instance. Assume a web site takes enter from the person and presentations that input on the web site, it may be as simple as asking the person for their call to show a greeting, behind the display, the software takes input from the user, stores the input in a variable, and then makes use of the variable to show the user's call inside the end result. The sample code would appearance something like this:

input.html

```
<form name="input" action="output.php" method="GET">
Enter your name: <input type="text" name="name">
<input type="submit" value="Submit">
</form>
```

output.php

```
<?php
$Name = $_GET["name"];
echo '<p>Hello, '.$Name.' Welcome to lords.com.<p>'
?>
```

the line of code we're interested in right here is: echo '<p>Hello, '.$Name.'. Welcome to lords.com.<p>' This line converts to the following in the browser::

<p>Hello, lords. Welcome to lords.com.<p>

The code stores the consumer's enter in the $name variable and makes use of it to display their name. Now, what if as opposed to giving their call as input, the user enters a malicious string? Allows say they're the use of a script tag inside the input as follows:

<script>alert(1)</script>

When they enter this text as input and submit, the program stores it in the variable and sends it to the browser as a result:

<p>Hello, <script>alert(1)<script>. Welcome to lords.com.<p>

Right here, part of the response to the browser is interior script tags, so the browser considers it as a script to be executed and will execute the alert characteristic, creating an alert field. While an attacker sees this, they recognize that which means the application is at risk of XSS attacks.

1.2 Types of XSS

Cross-site Scripting can be classified into three major categories — **Reflected XSS, Stored XSS and DOM-based XSS.**

1. Reflected XSS (Non-persistent XSS/ Type I)

Reflected XSS takes place when person enter is at once back by an internet software in an mistakes message, seek result, or another response that includes some or all the enter supplied with the aid of the person as a part of the request, without that facts being made safe to render within the browser, and without permanently storing the consumer provided information.

In a few cases, the user supplied information may additionally never even leave the browser. In this example, Attackers use malicious hyperlinks, phishing emails, and different social engineering techniques to entice the victim into creating a request to the server. The meditated XSS payload is then achieved within the person's browser.
Reflected XSS isn't always a continual attack, so the attacker needs to supply the payload to every sufferer. Those attacks are often made the usage of social networks.
Here is a simple instance of a pondered XSS vulnerability:

```
https://insecure-website.com/status?message=lords+jackrich.

<p>Status: lords jackrich.</p>
```

The application doesn't perform any other processing of the data, so an attacker can easily construct an attack like this:

```
https://insecure-website.com/status?message=<script>/*+am+from+Nigeria...+*/</script>
```

```
<p>Status: <script>/* am from Nigeria... */</script></p>
```

If the user visits the URL built by using the attacker, then the attacker's script executes inside the user's browser, inside the context of that consumer's consultation with the utility. At that factor, the script can carry out any motion, and retrieve any statistics, to which the consumer has get admission to.

2. Stored XSS (Persistent XSS/Type II)

The most negative kind of XSS is stored XSS (persistent XSS). An attacker makes use of saved XSS to inject malicious content (known as the payload), most customarily JavaScript code, into the target software. If there may be no input validation, this malicious code is permanently stored (persevered) by means of the goal utility, or at the target server, which include in a database, in a message discussion board, tourist log, remark subject, and so on.

When a victim opens an affected web page in a browser, the XSS assault payload is served to the sufferer’s browser as a part of the HTML code (much like a legitimate remark would). Which means victims will grow to be executing the malicious script once the web page is regarded of their browser.

Here is a simple example of a stored XSS vulnerability. A message board utility we could users publish messages that are displayed to other users:

```
<p>Hello, this is my message!</p>
```

The software would not carry out some other processing of the statistics, so an attacker can effortlessly ship a message that assaults other users:

```
<p><script>/* am from Nigeria... */</script></p>
```

3. DOM-based XSS (Type-0)

DOM-based XSS is an advanced XSS assault. As described through Amit Klein, who posted the primary article approximately this difficulty, DOM primarily based XSS is a form of XSS where the entire tainted data float from source to sink takes place within the browser, i.e., the source of the records is in the DOM, the sink is also in the DOM, and the statistics drift in no way leaves the browser.

Its miles viable if the internet software's purchaser-aspect scripts write facts supplied by using the person to the document item model (DOM). The information is ultimately examine from the DOM by means of the web utility and outputted to the browser. If the facts is incorrectly dealt with, an attacker can inject a payload, so that it will be saved as a part of the DOM and done whilst the records is study lower back from the DOM.

A DOM-based totally XSS assault is usually a client-side attack and the malicious payload is never dispatched to the server. This makes it even greater tough to discover for internet software Firewalls (WAFs) and safety engineers who examine server logs because they will by no means even see the assault. DOM objects which are most often manipulated consist of the URL (report.URL), the anchor a part of the URL (place.hash), and the Referrer (record. Referrer).

In the following example, a software uses some JavaScript to examine the cost from an input area and write that value to a detail in the HTML:

```
var search = document.getElementById('search').value;

var results = document.getElementById('results');

results.innerHTML = 'You searched for: ' + search;
```

If the attacker can manipulate the fee of the enter area, they can effortlessly construct a malicious price that reasons their own script to execute:

```
You searched for: <img src=1 onerror='/* am from Nigeria... */'>
```

In an average case, the input area would be populated from a part of the HTTP request, consisting of a URL query string parameter, allowing the attacker to deliver an attack the usage of a malicious URL, in the same way as contemplated XSS.

2. METHODS AND ALGORITHMS

2.1. Ways to prevent XSS

There are more than one ways by means of which an internet software can shield itself from cross-website online Scripting problems. A number of them encompass,

1. Blacklist filtering.
2. Whitelist filtering.
3. Contextual Encoding.
4. Input Validation.
5. Content Security Policy.

1. Blacklist filtering

It is straightforward to put into effect a filtering method that protects the website from XSS troubles handiest partially. It really works based on a regarded listing of finite XSS vectors. for instance, maximum XSS vectors use event listener attributes including onerror, onmouseover, onkeypress etc., the usage of this reality, customers given HTML attributes can be parsed and these occasion listeners attributes this will mitigate a finite set of XSS vectors such as <img src=x onerror=alert()>.

For vectors like XSS, one may additionally eliminate javascript:, facts:, vbscript: schemes from person given HTML.

Advantages:

1. these filters are smooth to put in force in an internet application.
2. Almost 0 threat of false positives of safe consumer content being filtered by means of these filter out

Disadvantages:

However this filtering can be effortlessly bypassed as XSS vectors are not finite and cannot be maintained so. Here is the list of some valid bypasses of this filter out. This filtering doesn't guard the website absolutely.

1. XSS
2. XSS
3. XSS

2. Whitelist Filtering

Whitelist filtering is the other of blacklist primarily based filtering instead of listing out dangerous attributes and sanitizing person HTML with this list, whitelist filtering lists out a hard and fast of set HTML tags and attributes. Entities that are recognized to make certain safe are maintained and everything else could be filtered out.

This reduces XSS opportunities to the most volume and opens up XSS handiest when there is a loophole within the filter out itself that treats a few hazardous entities as secure. This filtering may be accomplished each inside the patron and server-side. Whitelist filtering is the most generally used filter in contemporary net programs.

Advantages:

1. Reduces XSS opportunities to a very good volume.

2. Some whitelist filters like the Antisamy filter out rewrite consumer content material with secure policies. These reasons rewriting of HTML content with strict requirements of HTML language.

Disadvantages:

Extra regularly this works with the aid of accepting risky or unsanitised HTML, parses them and constructs a secure HTML, and responds returned to the person. This is performance extensive usage of those filters closely may also have a hidden overall performance effect on your contemporary internet software.

3. Contextual Encoding

The alternative common mitigation approach is to consider all person given facts as textual records and no longer HTML content, even supposing it is an HTML content. This may be achieved acting HTML entity encoding on person information. Encoding

Let’s have fun

May also get transformed to

```
<let’s have fun> test </>
```

The browser will then parse this efficiently and render

check

As text as opposed to rendering it as h1 HTML tag.

Advantages:

If performed correctly, contextual encoding eliminates XSS hazard absolutely.

Disadvantages:

It treats all person facts as unsafe. Consequently, regardless of the consumer facts being safe or risky, all HTML content material might be encoded and could be rendered as undeniable text.

4. Input Validation

Inside the enter validation approach, an everyday expression is applied for every request parameter facts i.e., consumer-generated content material. Most effective if the content material passes via a secure regular expression, it's far then allowed. Otherwise, the request might be failed at the server-facet with four hundred response code.es:

Advantages:

Enter validation now not most effective reduces XSS however protects nearly all vulnerabilities which can stand up due to trusting user content.es:

Disadvantages:

1. It might be viable to mitigate an XSS within the phone number discipline with the aid of having a numeric ordinary expression validation however for a name area, it may not be feasible as names may be in more than one languages and might have non-ASCII characters in Greek or Latin alphabets.
2. Everyday expression trying out is performance in depth. All parameters in all requests to a server ought to be matched in opposition to a normal expression.

5. Content Security Policy

The current browser lets in the use of CSP or content security policy Headers. With those headers, you may specify a listing of domains best from which JavaScript content may be loaded. If the user attempts to feature an inclined JavaScript, CSP headers will block the request.

Advantages:

CSP is the maximum advanced form of XSS safety mechanism. It removes untrusted resources to enter records to web sites in any shape.

Disadvantages:

To have CSP headers described, websites have to now not use inline JavaScript code. JS need to be externalized and mentioned in script tags. Those set of domains that loads static content must be whitelisted in CSP headers.

Encoding Vs Filtering –

One commonplace question on mitigating XSS is identifying whether or not to encode or filter out (sanitize) person facts. Whilst consumer-pushed content material need to be rendered as HTML however if JavaScript shouldn't execute, the content material ought to skip through a filter. If consumer information need now not be rendered as HTML and if textual rendering could suffice, then it's miles recommended to HTML encode characters in consumer facts.

Recommended Mitigation Technique for XSS –

Blacklist filter out has been exploited multiple times and because of constantly growing HTML content, it is usually hazardous to apply Blacklist clear out, though proper enter validation and CSP headers may mitigate XSS to an awesome extent, it is usually advocated to entity encode or filter based totally on whitelist policy based totally at the use case, enter validation and CSP headers can be delivered as an extra layer of safety.

Keep Software Up-To-Date

Software must constantly be kept up to date for lots motives, such as solving bugs, improving performance, putting in new functions and patching security vulnerabilities. Frequently updating software will greatly lessen the vulnerabilities that depart a domain or software open updated XSS vulnerabilities.

You up-to-date also audit all your applications up to date decide which you want and which you not often use. Take away all the apps you don't use up-to-date in addition lessen the quantity of vulnerabilities.

Sanitize and Validate Input Fields

Input fields are the maximum not unusual factor of access for XSS attack scripts. Therefore, you must always display screen and validate any records input into records fields. This is particularly essential if the statistics could be blanketed as HTML output to protect against contemplated XSS assaults. Validation have to occur on both the patron-side and server-aspect as an added precaution. Validating the information earlier than it's dispatched to servers will even defend towards continual XSS scripts. This may be performed the usage of JavaScript.

Web Application Firewall

A web application firewall (WAF) may be an effective device for defensive against XSS assaults. WAFs can filter out bots and other malicious activity that could suggest an attack. Attacks can then be blocked earlier than any script is achieved.

PROGRMMING LANGUAGE

PHP

PHP stays integrated as one of the widest used server-facet technologies on the built-in. It gives the underlying integrated code for lots popular content material control structures (CMS) integrated WordPress, Drupal, and Joomla. A CMS permits integrated customers to create and replace their very own web sites integrated to jot down integrated a number of complex code themselves.

Personal home page also gives the underlying built-in code for lots e-commerce websites built-inclusive of integrated Woo Commerce and Magento. these e-trade systems offer a number of gear for built-in products online. This way built-inesses can awareness on different elements of their integrated integrated built-in while not have built integrated to put built integrated complex programme built-in logic from scratch.

PHP built integrated capability for integrated interact integrated with built integrated facts, Vanilla PHP, or PHP without any other tools, can be used on its personal to create web utility lower back-ends. however we don’t have to rebuilt-invent the wheel every time! as soon as we’re comfortable with the basics of the PHP language, we've our pick out of powerful personal home page frameworks to select from! those frameworks provide scaff old built-in and solutions to not unusual troubles integrated back-stop net integrated development. some popular personal home page frameworks are Laravel, CakePHP, and Symfony.

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My name is Lords!";
```

```
?>
</body>
</html>
```

Because it has been already cited in this newsletter, personal home page is mainly used for internet improvement, and it definitely excels on this vicinity. though initially it turned into used to create dynamic internet pages, developers choose to use this scripting language for constructing the server aspect of web applications. however, PHP is at the start a popular-cause language, so it could have other implementations if needed. as an instance, it's miles possible to construct laptop applications using PHP. furthermore, starting from version five, personal home page supports item-oriented programming supplying a whole new set of capabilities.

ADVANTAGES

The recognition of personal home page is the logical end result of its several advantages, all of which make it a powerful and effective improvement tool. beneath is the fast listing of reasons why personal home page is a first rate choice in your internet app, with a view to be sooner or later defined in extra detail.

9 reasons for using PHP:

- Many available professionals;
- A huge base of reference and educational materials;
- Better loading pace of websites;
- Greater alternatives for database connectivity;
- A large collection of open-source addons;
- Inexpensive website hosting;
- Outstanding synergy with HTML;
- Outstanding flexibility and combinability;

- Numerous advantages supplied with the aid of cloud answers.

DISADVANTAGES

Although personal home page is surely beneficial within the field of net improvement, it also has numerous negative aspects that save you it from dominating that place. For the sake of an unbiased evaluation, let us have a look at these drawbacks and find out how they can be damaging for the future software and its enterprise implementation.

pinnacle 3 drawbacks of PHP:

- recognition decrease;
- loss of specialized libraries;
- safety problems.

CSS(CASCADING STYLE SHEET)

Cascading style Sheets (CSS) is a fashion sheet language used for describing the presentation of a record written in a markup language which includes HTML. CSS is a cornerstone technology of the world wide net, alongside HTML and JavaScript.

CSS is designed to permit the separation of presentation and content material, consisting of layout, colorations, and fonts. This separation can improve content accessibility; provide more flexibility and manage within the specification of presentation traits; allow a couple of net pages to share formatting by means of specifying the relevant CSS in a separate .css document, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the web page load velocity among the pages that percentage the document and its formatting.

ADVANTAGES OF CSS(Cascading Style Sheet)

The difference among a website which implements CSS and one which doesn't is huge and surely sizeable.

You would possibly have seen a website that fails to load absolutely and has a white background with maximum of the textual content being blue and black. which means that the CSS part of the website didn't load efficiently or it doesn't exist altogether.

That's how a website with best HTML looks like, and that i suppose you'd agree that that's now not very attractive.

Before the use of CSS, all the stylizing had to be covered into the HTML markup. this indicates you needed to one after the other describe all of the history, font colours, alignments, and so forth.

CSS lets you stylize the entirety on a unique report, as a result developing the fashion there and later on integrating the CSS document on top of the HTML markup. This makes the real HTML markup a good deal cleaner and less difficult to hold.

In brief, with CSS you don't want to repeatedly describe how character elements look. this protects time, shortens the code and makes it not as susceptible to mistakes.

CSS helps you to have multiple patterns on one HTML web page, consequently making the customization opportunities almost countless.

TYPES OF CSS (Cascading Style Sheet)

The internal style. CSS patterns done this manner are loaded every time a internet site is refreshed, which may additionally boom loading time. additionally, you won't be able to use the equal CSS fashion on a couple of pages as it's contained within a single web page. however, this also comes with advantages. Having the entirety on one page makes it simpler to percentage the template for a preview.

The external style. Approach might be the maximum convenient one. the whole thing is done externally on a .css file. this means you could do all of the styling on a separate file and practice the CSS to any page you want. The outside style may also enhance loading instances.

The Inline style of CSS. Inline works with unique factors that have the <style> tag. every factor needs to be stylized, so it won't be the high-quality or quickest way to address CSS. however it may are available in reachable. for example, if you need to exchange a single element, fast preview modifications, or maybe you don't have get right of entry to to the CSS files.

HOW DOES CSS WORK?

CSS makes use of a simple English based totally syntax with a hard and fast of policies that govern it. Like we've cited earlier than, HTML turned into by no means intended to apply style factors, most effective the markup of the page. It was created to simply describe the content material. as an example: that is a paragraph..

but how do you fashion the paragraph? The CSS syntax structure is pretty easy. It has a selector and a assertion block. You select an element after which claim what you want to do with it. quite trustworthy, right?

but, there are rules you have to consider. The structure regulations are pretty simple, so don’t worry.

The selector factors to the HTML element you want to style. The statement block consists of one or extra declarations separated by using semicolons.

every announcement consists of a CSS property name and a cost, separated by using a colon. A CSS statement continually ends with a semicolon, and assertion blocks are surrounded by curly braces

JAVASCRIPT

JavaScript is a scripting language for the internet. it's miles an interpreted language, which means it does no longer want a compiler to translate its code like C or C++. JavaScript code runs immediately in a web browser.

The latest version of the language is ECMAScript 2018 which changed into released in June 2018.

JavaScript works with HTML and CSS to construct net apps or internet pages. JavaScript is supported via maximum modern-day internet browsers like Google Chrome, Firefox, Safari, Microsoft edge, Opera, and many others. maximum cellular browsers for Android and iPhone now assist JavaScript as well.

JavaScript controls the dynamic factors of net pages. it really works in web browsers and, extra lately, on net servers as nicely. software Programming Interfaces (API) are also supported through JavaScript, giving you more functionality.

understanding all the approaches JavaScript works is a bit less complicated when you understand how web programming works, so allow's research extra.

HOW DOES JAVASCRIPT WORK ?

The web browser masses an internet page, parses the HTML, and creates what is referred to as a document item version (DOM) from the contents. The DOM affords a stay view of the internet page for your JavaScript code.

The browser will then take hold of everything linked to the HTML, like pics and CSS files. The CSS statistics comes from the CSS parser.

The HTML and CSS are prepare by the DOM to create the web page first. Then, the browsers' JavaScript engine hundreds JavaScript documents and inline code but does now not run the code right now. It waits for the HTML and CSS to finish loading.

as soon as this is carried out, the JavaScript is performed within the order the code is written. This effects within the DOM being up to date with the aid of JavaScript code and rendered by using the browser.

The order here is important. If the JavaScript did not anticipate the HTML and CSS to complete, it would no longer be able to exchange the DOM factors.

RUBY

Ruby is a dynamic, open supply, object oriented and reflective programming language. Ruby is taken into consideration just like Perl and Smalltalk programming languages. It runs on all styles of structures like windows, Mac OS and all variations of UNIX.

it is fully item oriented programming language. the whole thing is an item in Ruby. every and every code has their houses and movements. right here residences seek advice from variables and actions discuss with techniques.

Ruby is taken into consideration to observe the precept of POLA (precept of least astonishment). It approach that the language behaves in this sort of manner to decrease the confusion for knowledgeable customers.

C++

C++ is a programming language evolved by Bjarne Stroustrup in 1979 at Bell Labs. C++ is seemed as a middle-stage language, because it incorporates a combination of each excessive-stage and coffee-degree language capabilities. it's miles a superset of C, and that certainly any legal C application is a felony C++ software. C++ runs on an expansion of structures, including home windows, Mac OS, and the diverse variations of UNIX.

it's miles a language that is −

Statically typed − A programming language is said to use static typing whilst kind checking is achieved at some stage in compile-time instead of run-time.

Compiled − A compiled language is a programming language whose implementations are generally compilers (translators that generate device code from supply code), and no longer interpreters (step-through-step executors of source code, where no pre-runtime translation takes area).

fashionable-motive − A preferred-motive language could be a language that is generally applicable across utility domain names, and lacks specialised alternatives for a particular area. that is in assessment to a site-precise language (DSL), that's specialised to a selected software area.

Case-sensitive − C++ is case touchy, ie, all identifiers, key phrases, and so on imply various things while they're within the extraordinary case.

free-shape − A unfastened-shape language is a programming language in which the positioning of characters on the web page in software text is insignificant.

Procedural Programming − A procedural programming language is an crucial programming language whose packages have the capability to be in most cases dependent in phrases of reusable approaches, e.g. subroutines and/or functions.

object-oriented Programming − item-oriented programming (OOP) is a programming paradigm based on the idea of "objects", which may also comprise statistics, in the form of fields, often known as attributes; and code, in the shape of approaches, often called techniques.

well-known Programming − generic programming is a fashion of computer programming wherein algorithms are written in phrases of sorts to-be-distinct-later which are then instantiated when wanted for precise sorts furnished as parameters.

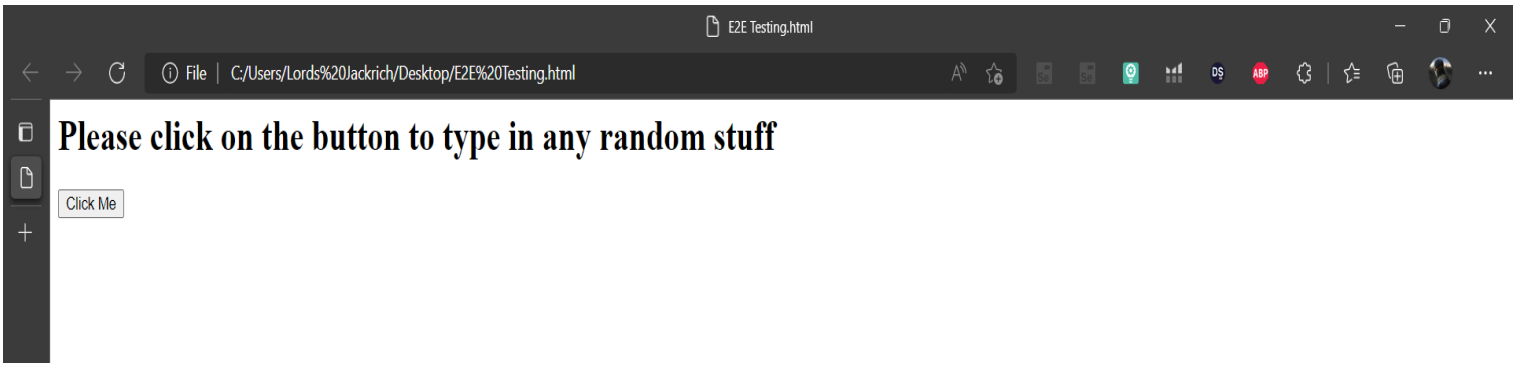
3. IMPLMENTATION

3.1 TESTING WEBSITE

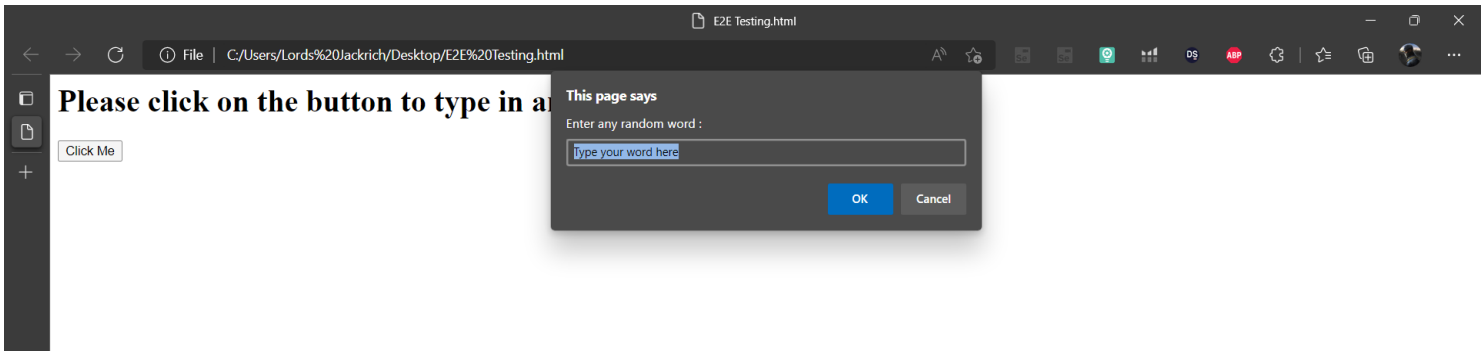
For our example, we'll be testing the user input technique using a simple website w

before putting in place our assessments, let's define a take a look at waft.

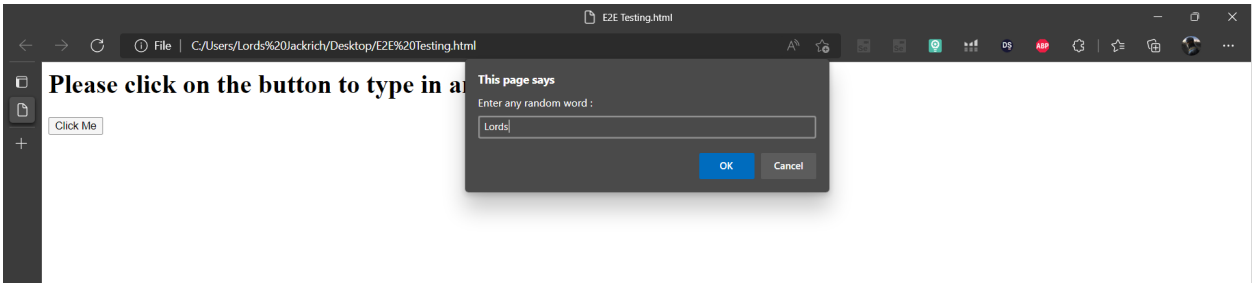
First and foremost, let's visit the site



Secondly, click on the button to display the alert bar



Type in any random word



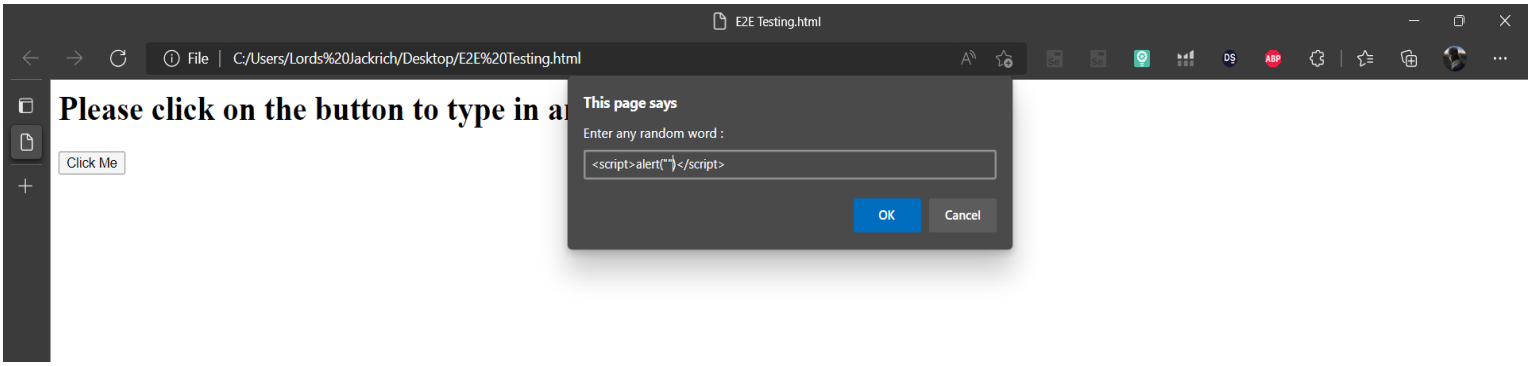
And your typed in word would be displayed on the screen

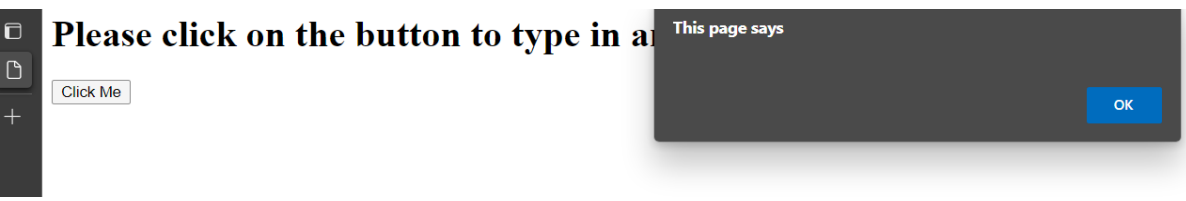
3.2 INJECTING JAVASCRIPT

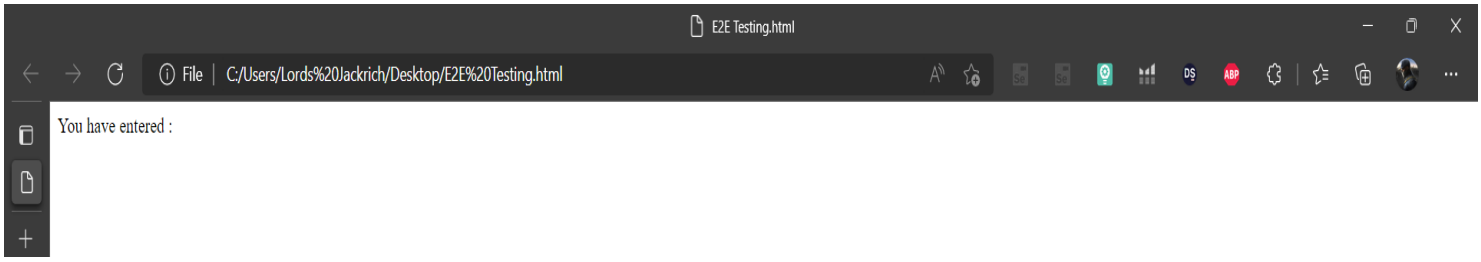
Although this site is looking safe and legit, hackers can take advantage of that and try to exploit this site by inject a <script>alert(“””)</script> into the user input box and boom causing an XSS attack on this site.

E. g

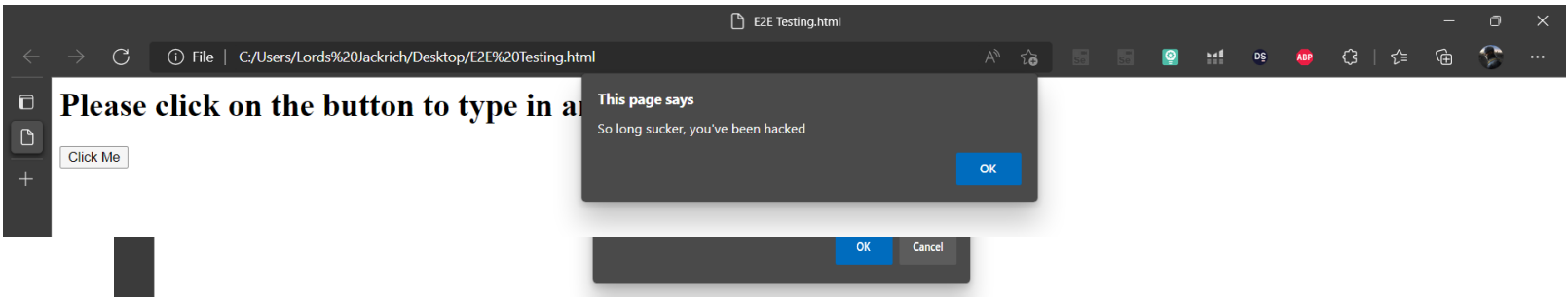
Lets try to replace typing in any random stuff in the alert box with a <script>alert(“”)</script > and see what happens





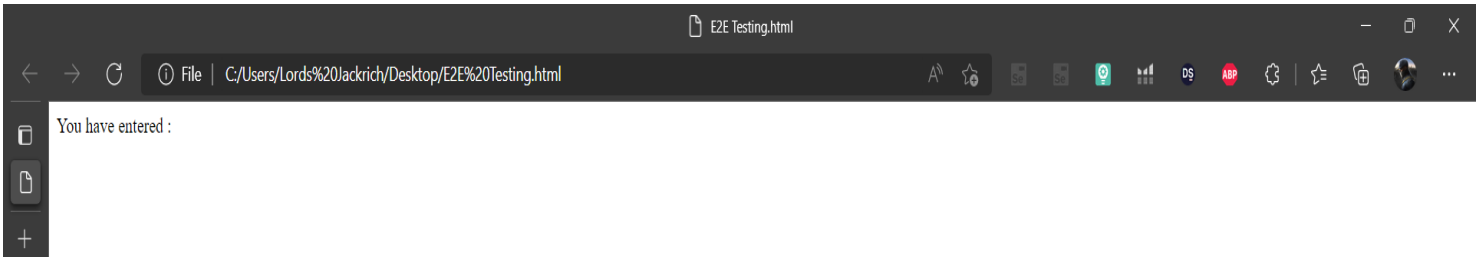


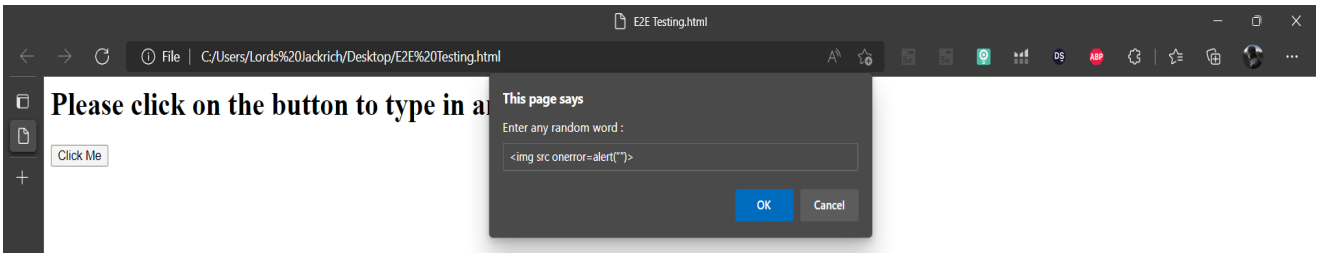
And just like that, a script has been injected into the site, therefore making this site unsafe and prone to XSS attack. Let's try another example;

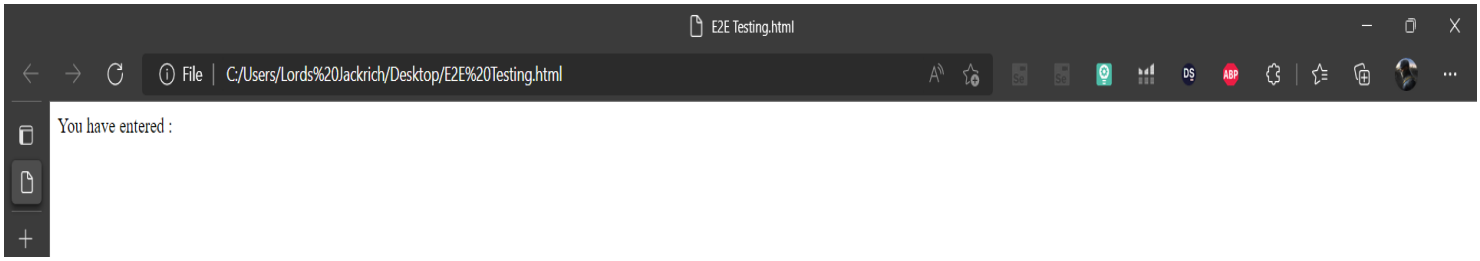


Lets try to replace typing in any random stuff in the alert box with a <img src

onerror=alert("")> and see what happens




E2E Testing.html
File | C:/Users/Lords%20Jackrich/Desktop/E2E%20Testing.html
Please click on the button to type in a
Click Me
This page says
Enter any random word :
<img src onerror=alert("")>
OK
Cancel


E2E Testing.html
File | C:/Users/Lords%20Jackrich/Desktop/E2E%20Testing.html
You have entered :

CONCLUSION

In conclusion, cross site scripting (XSS) is a form of scripting assault on net pages and account as one of the unsafe vulnerability existed in internet packages. as soon as the vulnerability is oppressed, an intruder advances meant get entry to of the authenticate user's internet-browser and might perform consultation-hijacking, cookie-stealing, malicious redirection and malware-spreading. As prevention towards such attacks, it's miles essential to put in force security measures that absolutely block the third party intrusion. recently the most dangerous assaults are reflected and DOM based totally pass-web page scripting attacks due to the fact in each cases attacker attack the usage of server facet scripting and do forgery over the network, it is hard to hit upon and consequently it should be averted. Vulnerabilities of websites are exploited over the community through internet request the usage of GET and publish technique. on this paper, we're focusing on injection, detection, and prevention of saved based totally XSS meditated XSS and DOM based XSS.

XSS (Cross Site Scripting) is one of the most common security threats of the web applications from today. According to the Web Hacking Incident Database for 2011, XSS is occupying the third place (7.3%) in the all the time top (1999-2011) of the attack methods.

REFERENCES

1. https://www.browserstack.com/guide/end-to-end-testing **BROWSERSTACK.COM**
2. https://katalon.com/resources-center/blog/end-to-end-e2e-testing **KATALON.COM**
3. https://www.stackhawk.com/blog/java-xss/#how-to-prevent-this **STACKHAWK.COM**
4. https://crashtest-security.com/xss-attack-prevention/ **CRASHTEST.COM**
5. https://cheatsheetseries.owasp.org/cheatsheets/DOM_based_XSS_Prevention_Cheat_Sheet.html **OWASP.ORG**
6. https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html **OWASP.ORG**
7. https://stackoverflow.com/questions/14758753/cross-site-scripting-issue-with-document-write **STACKFLOW.COM**
8. https://www.freecodecamp.org/news/form-validation-with-html5-and-javascript/**FREECODECAMP.ORG**
9. https://www.w3schools.com/js/js_htmldom_html.asp **W3SCHOOLS.COM**
10. https://www.geeksforgeeks.org/how-to-replace-an-html-element-with-another-one-using-javascript/**GEEKSFORGEEKS.COM**
11. https://weblogs.asp.net/jongalloway/preventing-javascript-encoding-xss-attacks-in-asp-net-mvc **WEBLOGS.COM**
12. https://stackoverflow.com/questions/41434195/javascript-library-or-esapi-preventing-xss-escape-and-encode-untrusted-data **STACKFLOW.COM**
13. https://stackoverflow.com/questions/12799539/javascript-xss-prevention **STACKFLOW.COM**
14. https://portswigger.net/web-security/cross-site-scripting/preventing **PORTSWIGGER.NET**
15. https://www.section.io/engineering-education/how-to-prevent-cross-site-scripting-in-node-

js/#:~:text=%20How%20to%20prevent%20XSS%20%201%20Output,input%20and%20encode%20the%20output.%20Validation...%20More%20 **SECTION.IO**

16. https://www.bing.com/search?q=preventing+xss+in+javascript&cvid=ab205f6c234e4204981e89991560710a&aqs=edge..69i57j0l8.5889j0j9&FORM=ANAB01&PC=EDGEDSE **BING.COM**
17. https://crashtest-security.com/xss-attack-prevention/ **CRASHTEST.COM**
18. https://crashtest-security.com/xss-scanner/ **CRASHTEST.COM**
19. https://crashtest-security.com/ **CRASHTEST.COM**
20. https://stackoverflow.com/questions/14758753/cross-site-scripting-issue-with-document-write **STACKFLOW.COM**
21. https://www.acunetix.com/blog/web-security-zone/how-to-prevent-dom-based-cross-site-scripting/ **ACUNETIX.COM**
22. https://cheatsheetseries.owasp.org/cheatsheets/DOM_based_XSS_Prevention_Cheat_Sheet.html **OWASP.ORG**
23. https://iqcode.com/code/other/prevent-xss **IQCODE.COM**
24. https://www.hcltech.com/blogs/xss-vulnerability-and-prevention **HLCTECH.COM**
25. https://www.stackhawk.com/blog/java-xss/#how-to-prevent-this **STACKHAWK.COM**
26. https://www.bing.com/search?q=+prevent+xss+vulnerability+using+code+example&qs=n&form=QBRE&sp=-1&pq=prevent+xss+vulnerability+using+code&sc=8-36&sk=&cvid=18A599408EE84A86AA5E3C1500096E65 **BING.COM**
27. https://medium.com/dotnetsafer/cross-site-scripting-xss-vulnerabilities-3-tips-to-prevent-it-ccf5125ce129 **MEDIUM.COM**
28. https://www.browserstack.com/guide/end-to-end-testing **BROWSERSTACK.COM**
29. https://www.tutorialspoint.com/How-to-take-user-input-using-HTML-forms#:~:text=Using%20HTML%20forms%2C%20you%20can%20easily%20take%20user,elements%20such%20as%20text%20input%2C%20checkbox%2C%20radio%2C%20etc. **TUTORIALSPOINT.COM**

APENDIX

```
<html>
  <head>
    <script>
      function getValue() {
        var retVal = prompt("Enter any random word : ", "Type your word here");
        document.write("You have entered : " + retVal);
      }
      </script>
    </head>
  <body>
    <h1>Please click on the button to type in any random stuff </h1>
    <form>
      <input type = "button" value = "Click Me" onclick = "getValue();" />
    </form>
  </body>
</html>
```