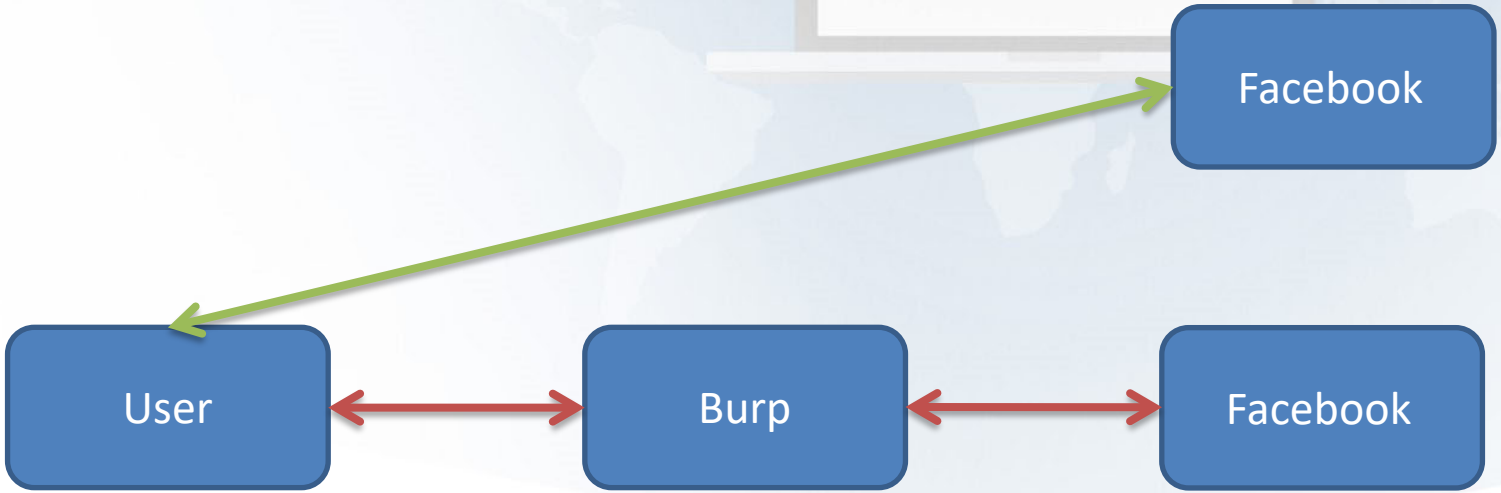
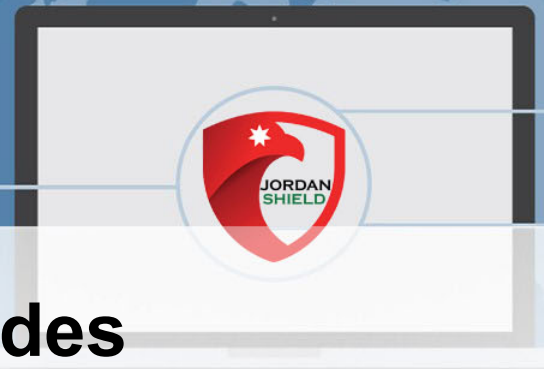
CSC – Jordan Shield Special Edition

Powered By : Mohammed Kher Al-Khawaldeh.

Burp Suite



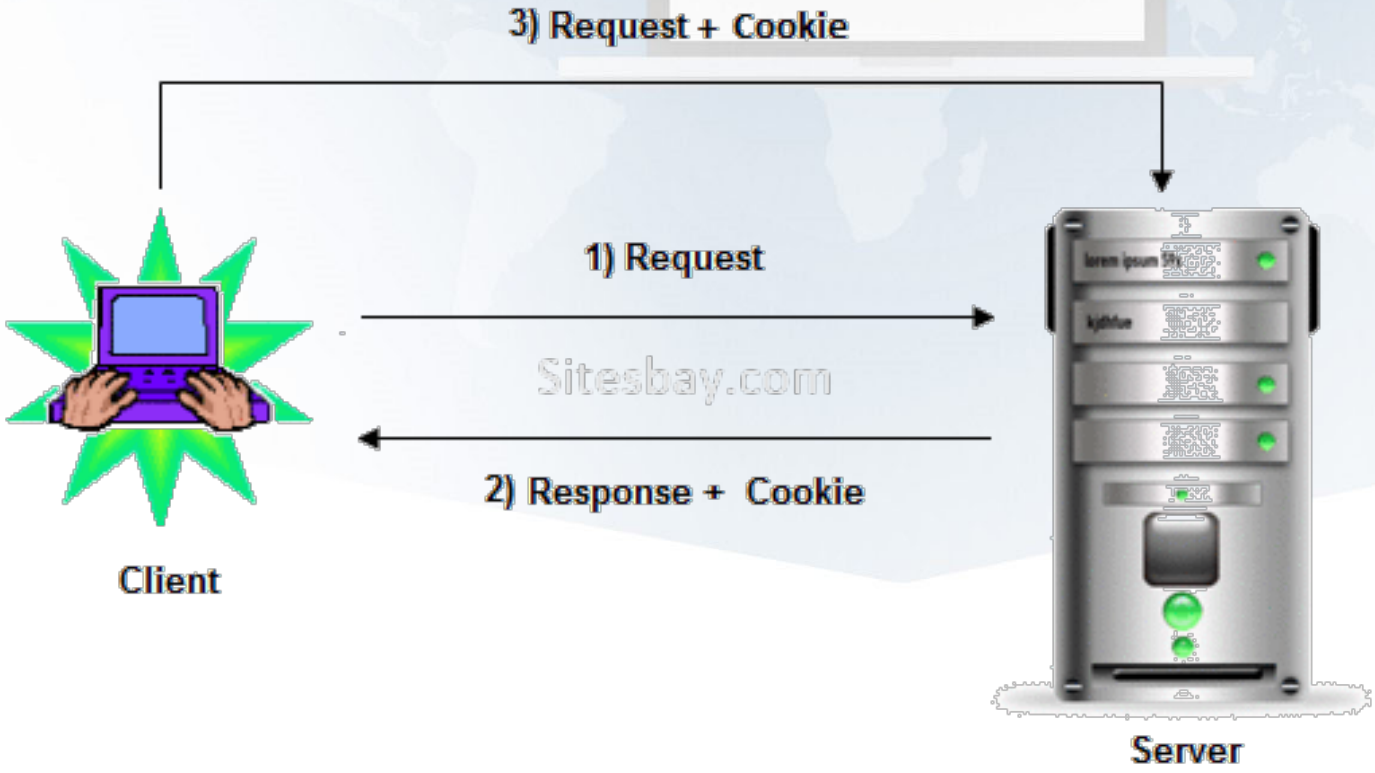


HTTP response status codes

1. Informational responses (100–199)
2. Successful responses (200–299)
3. Redirects (300–399)
4. Client errors (400–499)
5. Server errors (500–599)

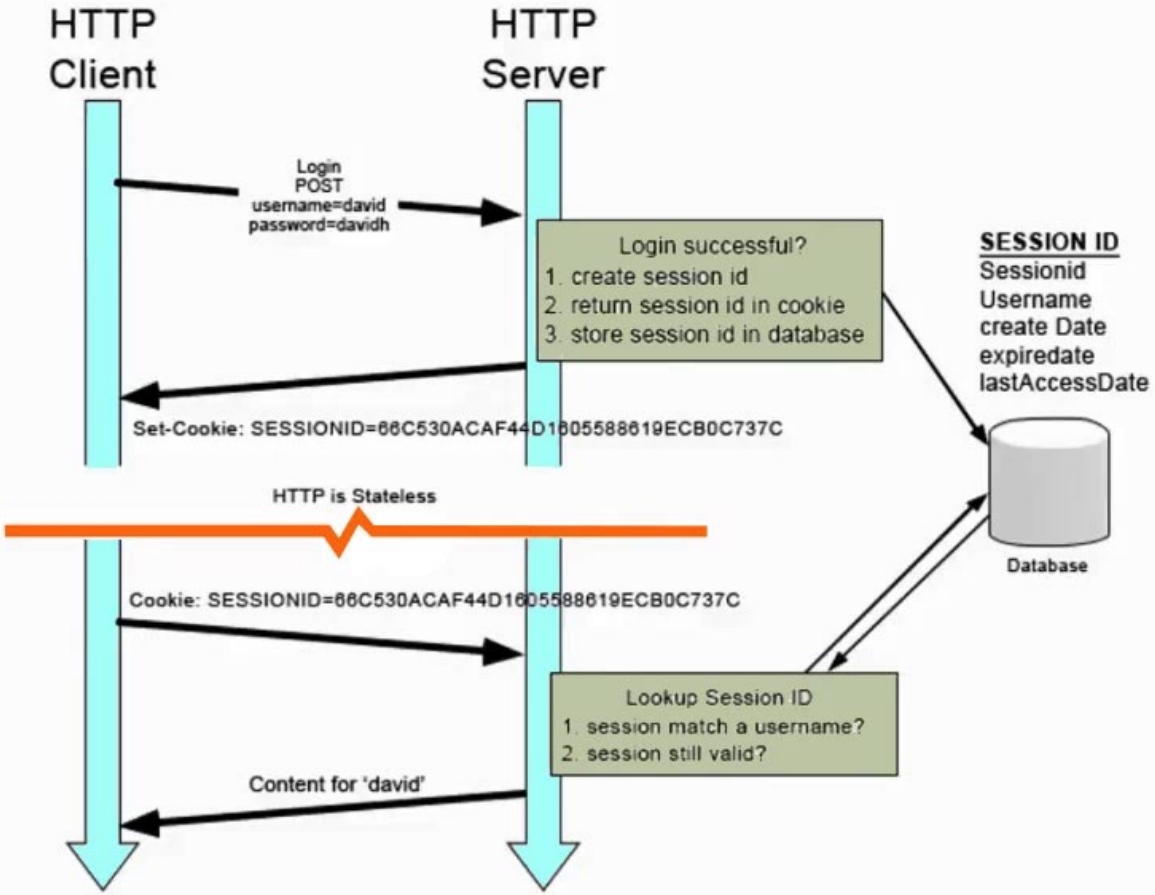
https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

Burp Suite


3) Request + Cookie
1) Request
Sitesbay.com
2) Response + Cookie
Client
Server

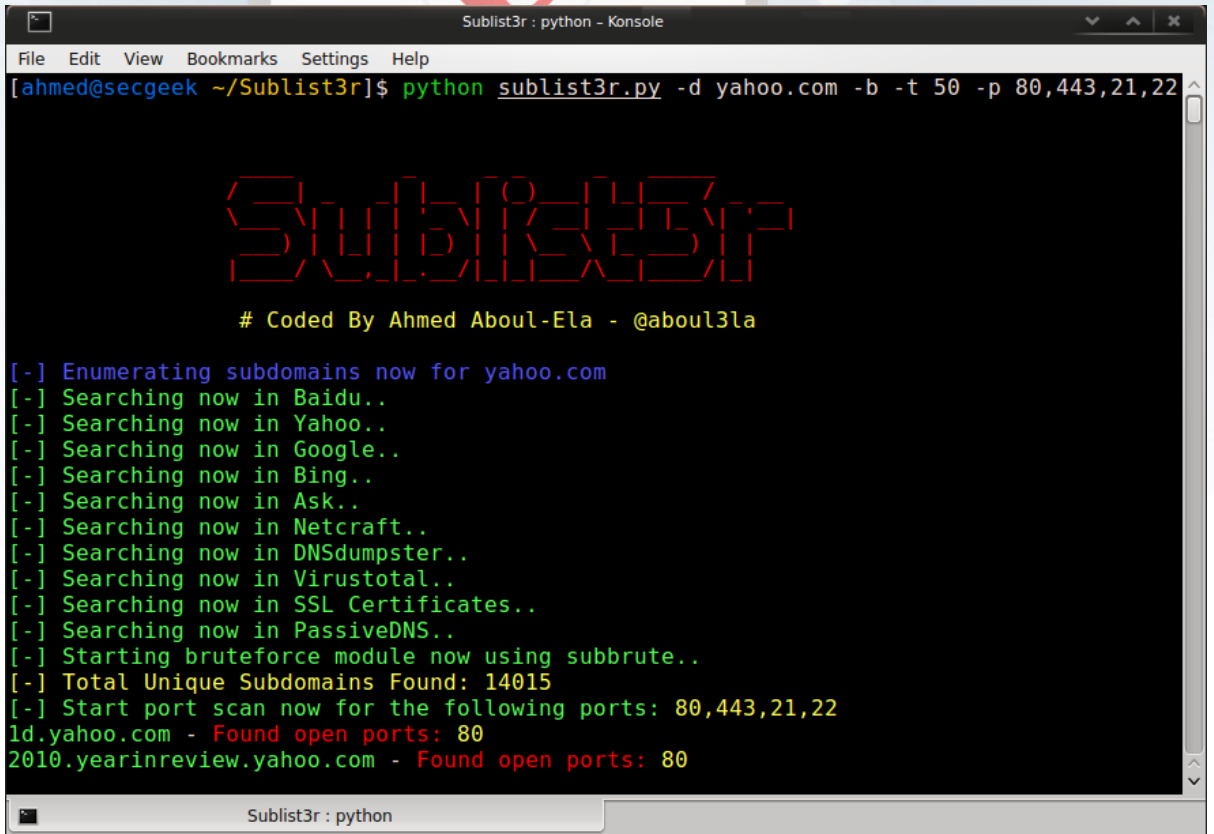
Burp Sui

Cookies and Sessions

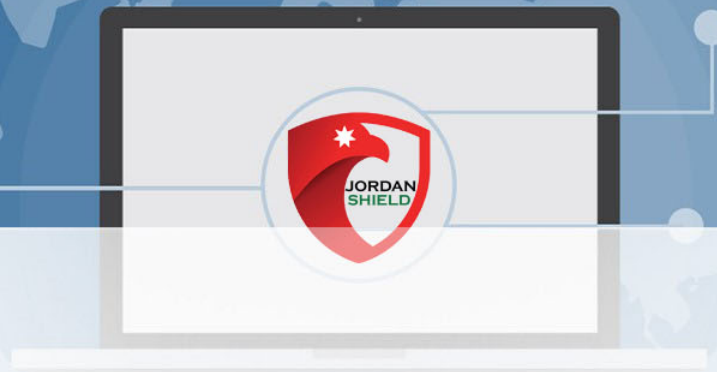


SubList3r

Sublist3r is a python tool designed to enumerate subdomains of websites using OSINT



DNS Record Types



A : IPv4

AAAA : IPv6

PTR : IP pointer multi domains

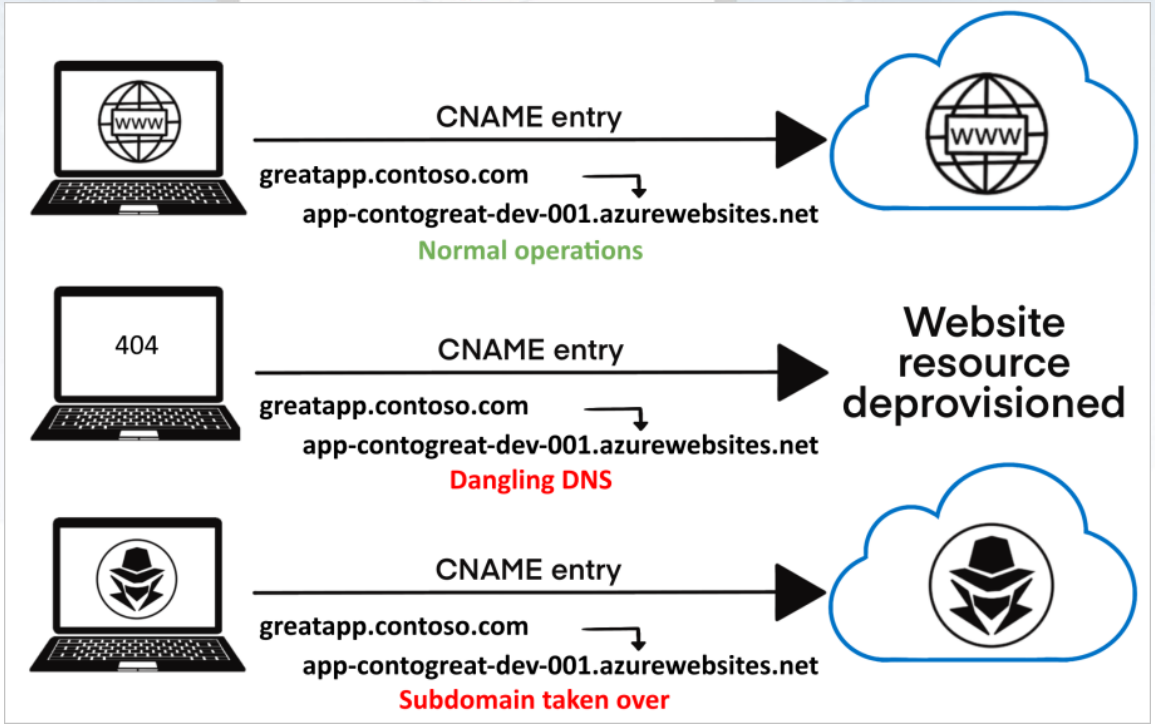
CNAME : alias

MX : mail server

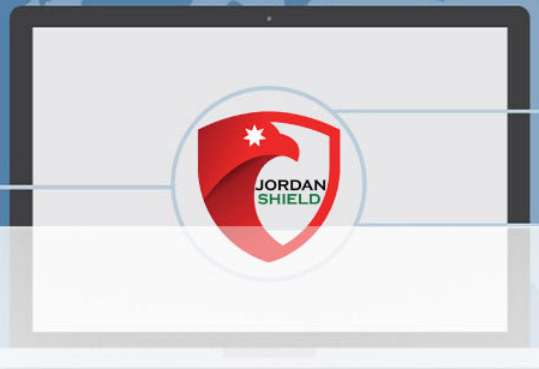
Subdomain Take Over

Subdomain Take Over

Tool to use : Its Over



Vulnerability

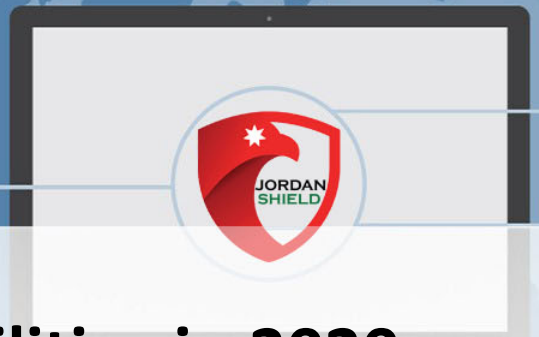


What is a vulnerability?

A vulnerability is a weakness or error in a system or device's code that, when exploited, can compromise the confidentiality, availability, and integrity of data stored in them through unauthorized access, elevation of privileges, or denial of service. A code or tool used to take advantage of a vulnerability is called an exploit.

Rec: TrendMicro

Vulnerability



The Top 10 OWASP vulnerabilities in 2020 are:

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access control
- Security misconfigurations
- Cross Site Scripting (XSS)
- Insecure Deserialization
- Using Components with known vulnerabilities
- Insufficient logging and monitoring

Vulnerability

HTTP [Header , Methods , Status code.]

HTTP Methods: [GET , POST , OPTIONS , DELETE , PUT]

HTTP Request

HTTP Response

Vulnerability

HTTP request methods

HTTP defines a set of request methods to indicate the desired action to be performed for a given resource.
Although they can also be nouns, these request methods are sometimes referred to as HTTP verbs.
Each of them implements a different semantic, but some common features are shared
by a group of them: e.g. a request method can be safe, idempotent, or cacheable.

GET

[The GET method requests a representation of the specified resource.
Requests using GET should only retrieve data.]

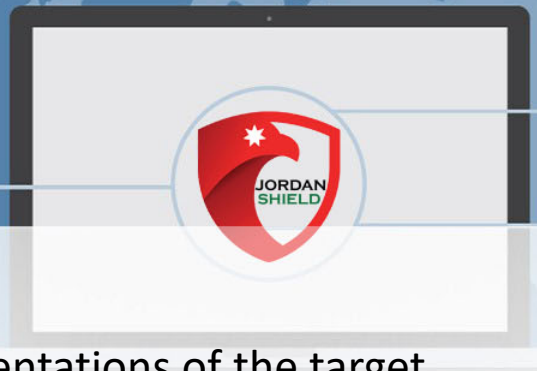
HEAD

[The HEAD method asks for a response identical
to that of a GET request, but without the response body.]

POST

[The POST method is used to submit an entity to the specified resource,
often causing a change in state or side effects on the server.]

Vulnerability



PUT

[The PUT method replaces all current representations of the target resource with the request payload.]

DELETE

[The DELETE method deletes the specified resource.]

CONNECT

[The CONNECT method establishes a tunnel to the server identified by the target resource.]

OPTIONS

[The OPTIONS method is used to describe the communication options for the target resource.]

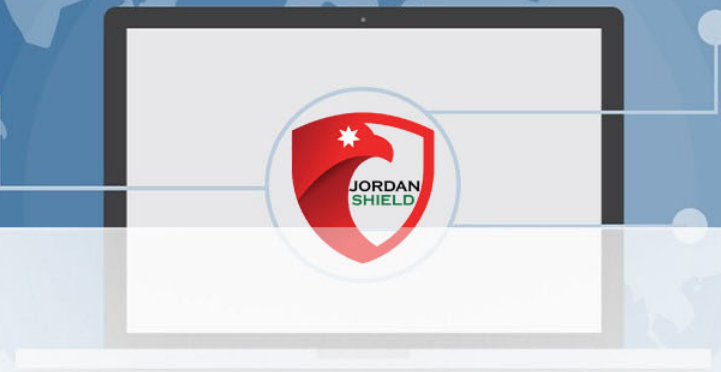
TRACE

[The TRACE method performs a message loop-back test along the path to the target resource.]

PATCH

[The PATCH method is used to apply partial modifications to a resource.]

Vulnerability



Introduction To HTML

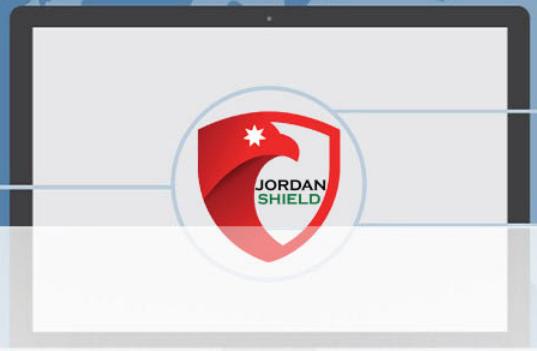
```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Ref:W3School

Vulnerability



The <!DOCTYPE> Declaration

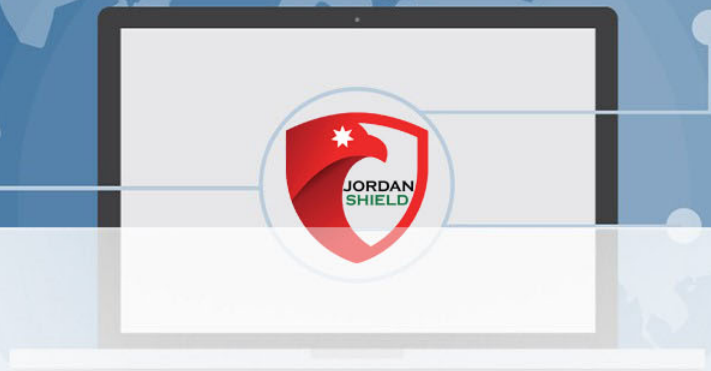
The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

```
<!DOCTYPE html>
```

Ref:W3School

Vulnerability




```
<html>

</html>
```

```
<head>
<title>Page Title</title>
</head>
```

```
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
```

Ref:W3School

Vulnerability

Link:

```
<a href="https://www.w3schools.com">This is a link</a>
```

Image:

```
<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">
```

Ref:W3School

Vulnerability

Absolute URLs

```
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>
```

Relative URLs

```
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

Ref:W3School

Vulnerability

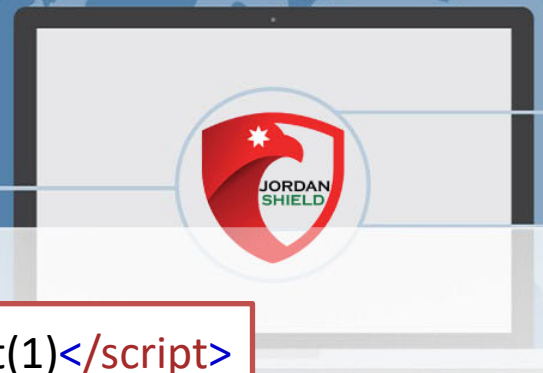
```
<a href="default.asp"><img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;"></a>
```

```
<button onclick="document.location='default.asp'">HTML Tutorial</button>
```

```
<iframe src="demo_iframe.htm" height="200" width="300" title="Iframe Example"></iframe>
```

Ref:W3School

Vulnerability



```
<script>alert(1)</script>
```

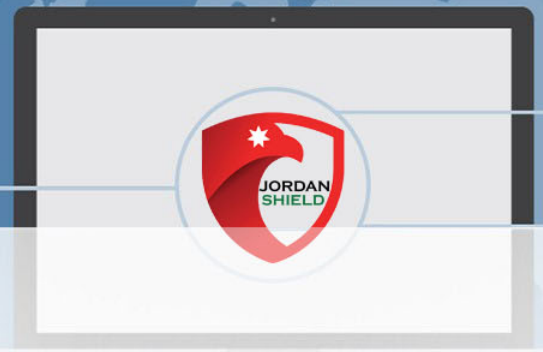
```
<script>confirm(1)</script>
```

```
<script>prompt(1)</script>
```

```
<script>confirm(1)</script>
```

Ref:W3School

Vulnerability



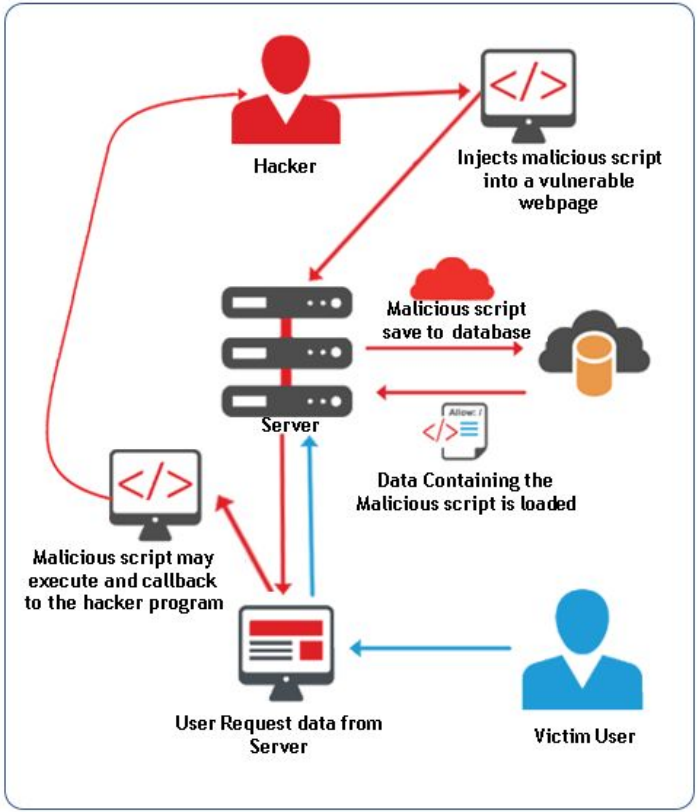

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Ref:W3School

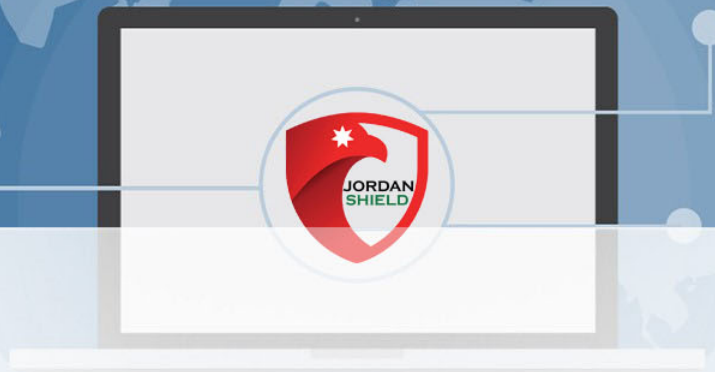
Vulnerability

XSS : Cross Site Script

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites



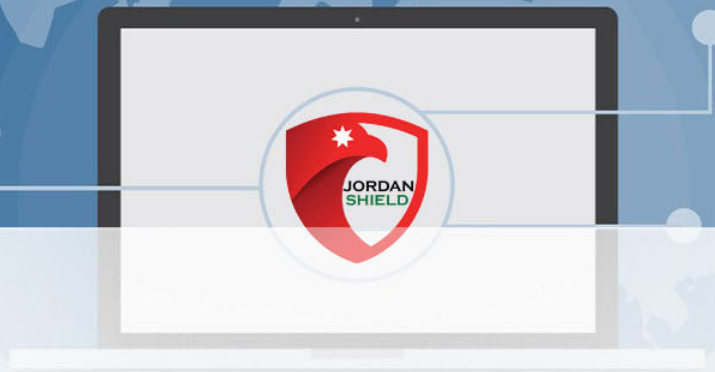
Vulnerability



Types of XSS:

1- XSS (REFLECTED)
2- XSS (STORED)
3- XSS (DOM BASED)

Vulnerability



1- XSS (REFLECTED)

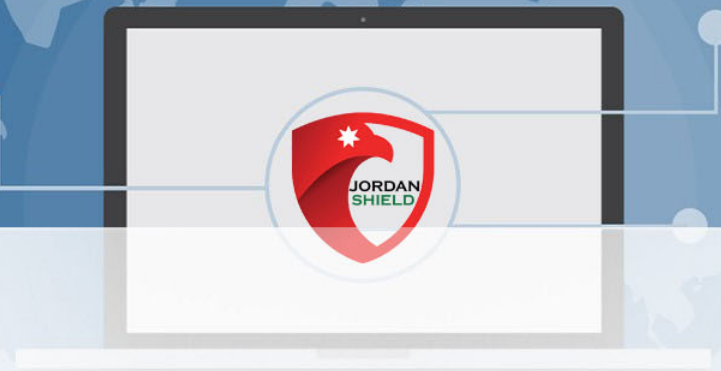
Nothing saved in database.

Try This Payload: <script>alert(1)</script>

<img src="null" onerror="alert(1)">

<sCriPt>confirm(1)</sCriPt>
<script>prompt(1)</script>

Vulnerability



2- XSS (STORED)

Payloads are saved in database.

Try This Payload: <script>alert(1)</script>

<img src="null" onerror="alert(1)">
<sCriPt>confirm(1)</sCriPt>
<script>prompt(1)</script>

Vulnerability

Payloads to test:

<script>alert(1)</script>	Moh
<sCriPt>confirm(1)</sCriPt>	Moh
<sCriPt>alert(1)</sCriPt>	<button onclick="document.location='default.asp'">HTML Tutorial</button>
<ScRiPt>prompt(1)</ScRiPt>	
moh	

Vulnerability