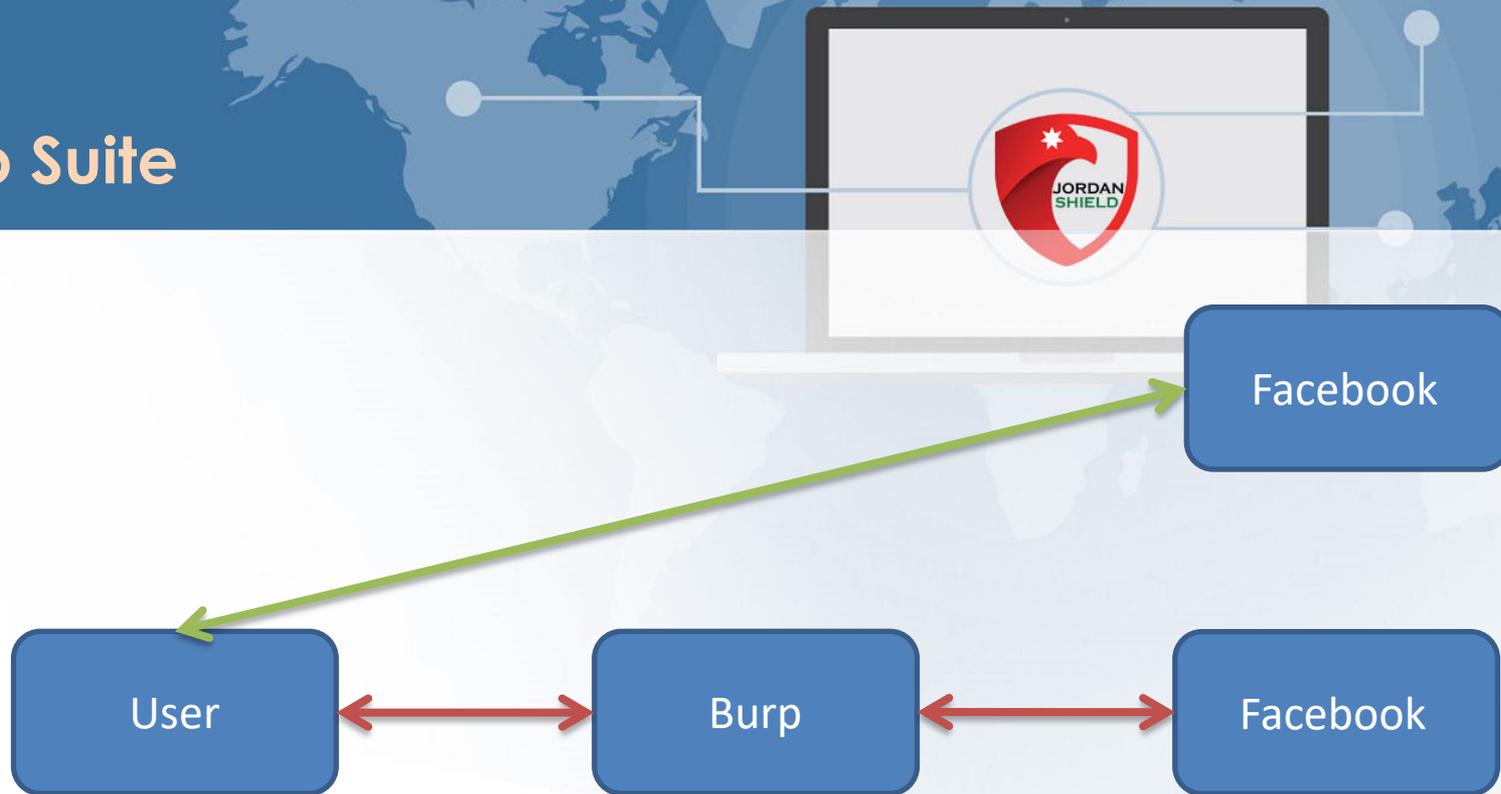




CSC – Jordan Shield Special Edition  
Powered By : Mohammed Kher Al-Khawaldeh.

# Burp Suite





## HTTP response status codes

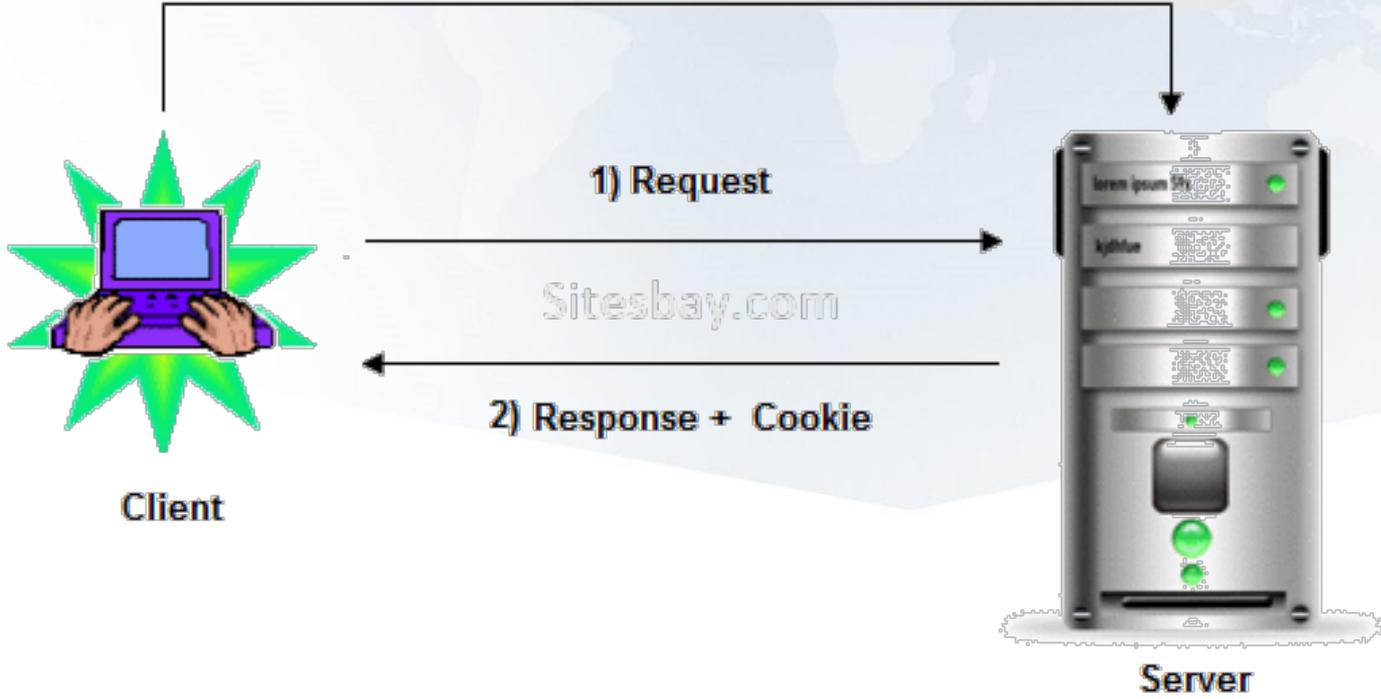
1. Informational responses (100–199)
2. Successful responses (200–299)
3. Redirects (300–399)
4. Client errors (400–499)
5. Server errors (500–599)

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

# Burp Suite

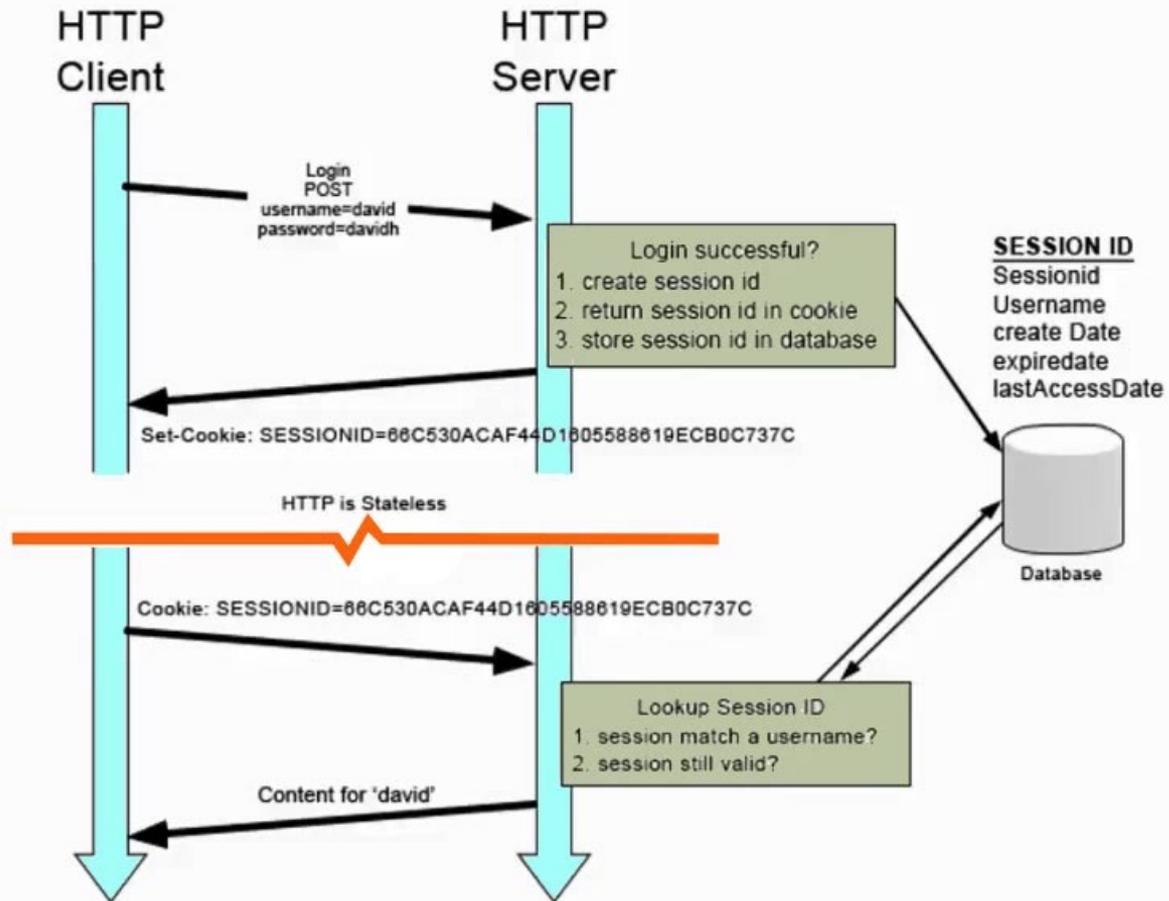


3) Request + Cookie



# Cookies and Sessions

Burp Suite



# SubList3r



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

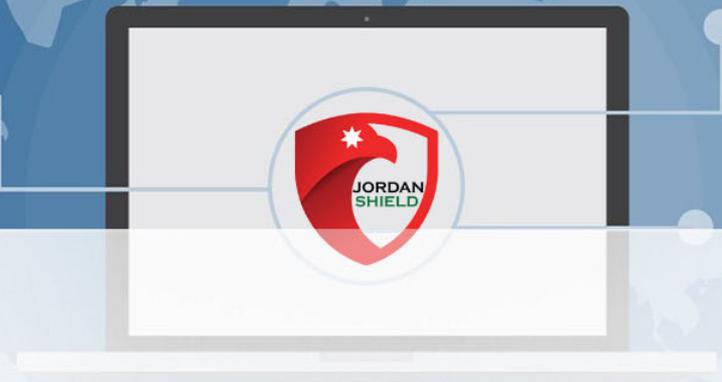
```
Sublist3r : python - Konsole
File Edit View Bookmarks Settings Help
[ahmed@secgeek ~/Sublist3r]$ python sublist3r.py -d yahoo.com -b -t 50 -p 80,443,21,22

Sublist3r
# Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for yahoo.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[-] Starting bruteforce module now using subbrute..
[-] Total Unique Subdomains Found: 14015
[-] Start port scan now for the following ports: 80,443,21,22
1d.yahoo.com - Found open ports: 80
2010.yearinreview.yahoo.com - Found open ports: 80

Sublist3r : python
```

# DNS Record Types



**A** : IPv4

**AAAA** : IPv6

**PTR** : IP pointer multi domains

**CNAME** : alias

**MX** : mail server

# Subdomain Take Over



```
app.neworg.com 60 IN CNAME newproject.herokuapp.com
```

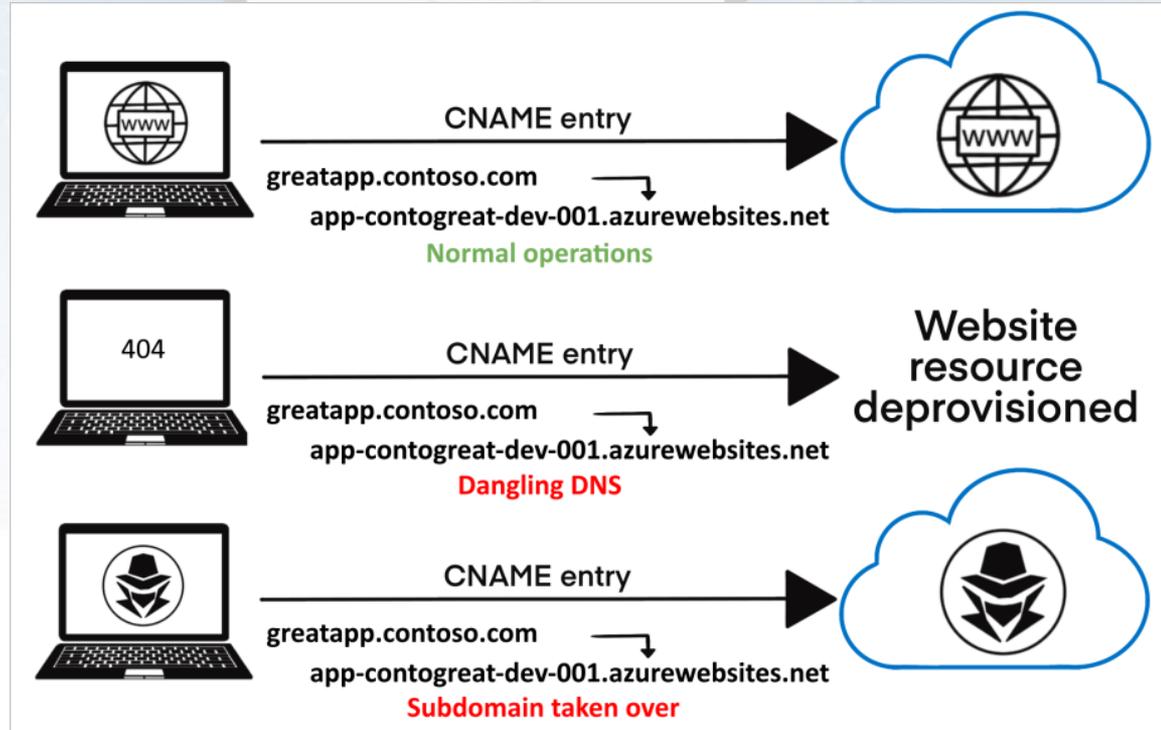
User visits `app.neworg.com` in his browser which has a CNAME pointing to `newproject.herokuapp.com`

User actually browses content from the web app at `newproject.herokuapp.com`

# 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



## 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

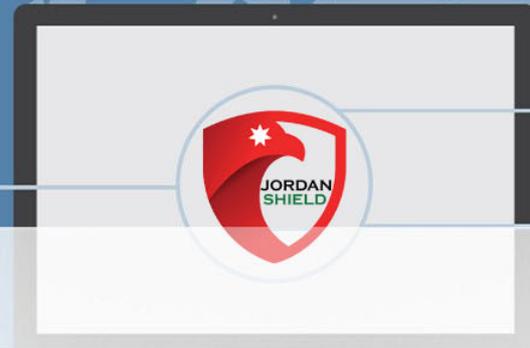
to that of a GET request, but without the response body.]

### 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.]

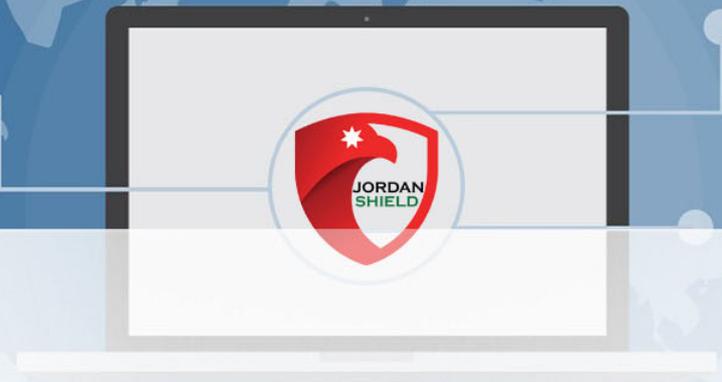
## TRACE

[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

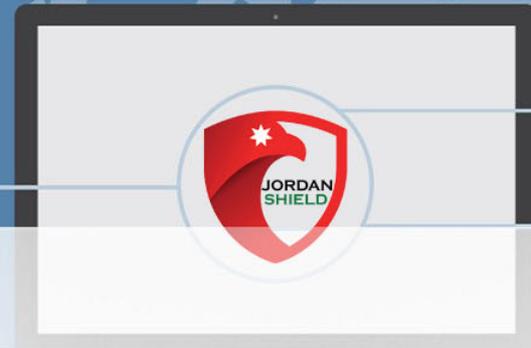


## Introduction To HTML

```
<!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

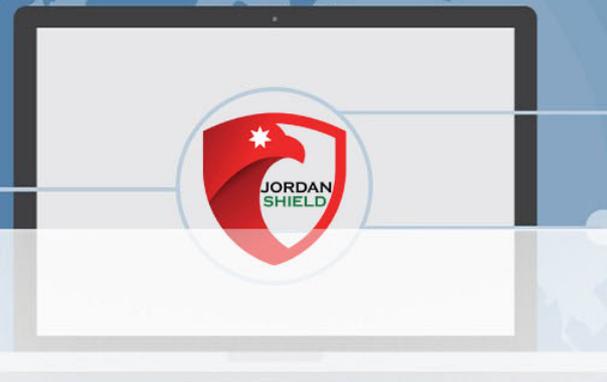
The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.

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:

```

```

Ref:W3School



## 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"></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>
```

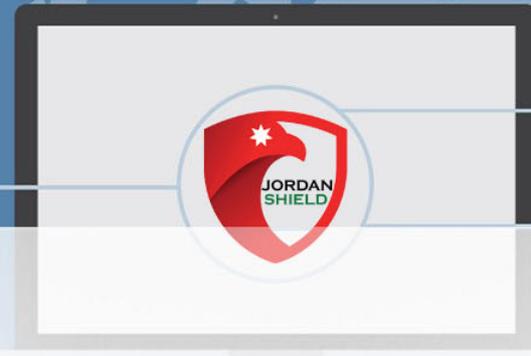
```
<script>confirm(1)</script>
```

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

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

Ref:W3School

# Vulnerability



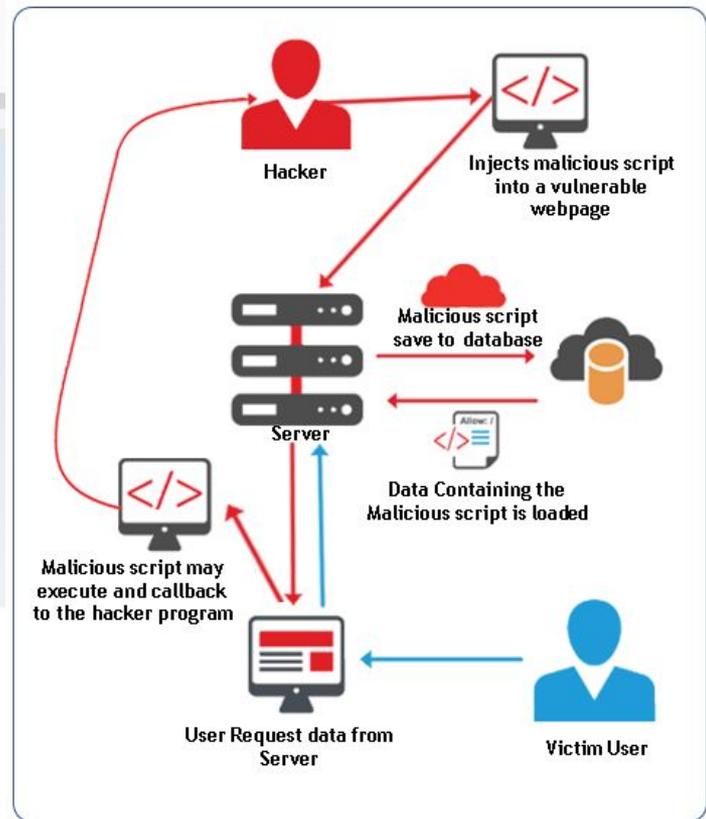
```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Ref:W3School

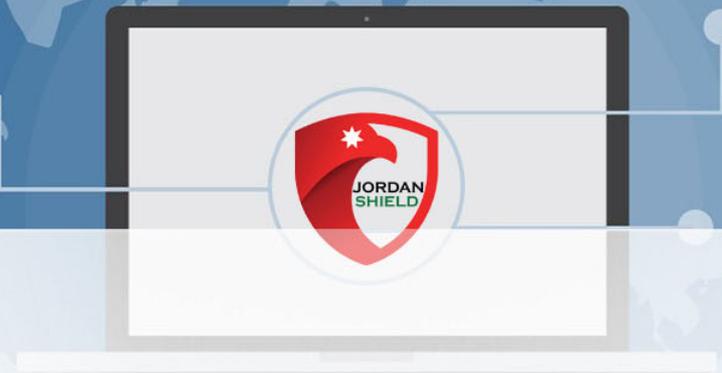
# Vulnerability

**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



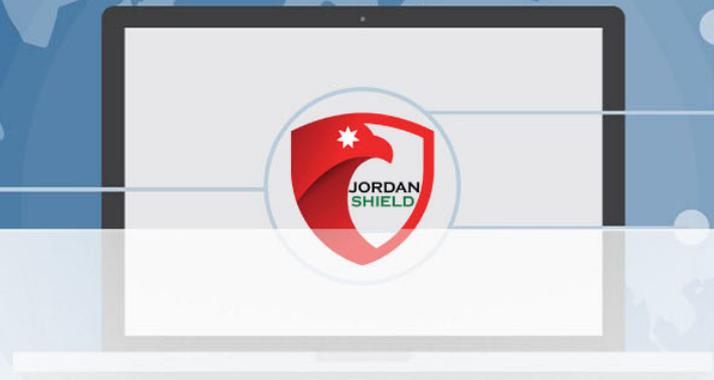
# Vulnerability



Types of XSS:

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

# Vulnerability



## 1- XSS (REFLECTED)

Nothing saved in database.

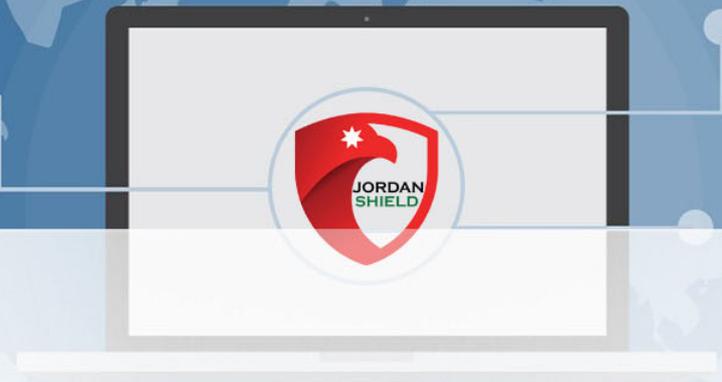
Try This Payload: `<script>alert(1)</script>`

``

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

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

# Vulnerability



## 2- XSS (STORED)

Payloads are saved in database.

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

``

`<scriPt>confirm(1)</sCriPt>`

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

# Vulnerability



Payloads to test:

<code>&lt;script&gt;alert(1)&lt;/script&gt;</code>	<code>&lt;h3&gt;Moh&lt;/h3&gt;</code>
<code>&lt;scriPt&gt;confirm(1)&lt;/scriPt&gt;</code>	<code>&lt;h6&gt;Moh&lt;/h6&gt;</code>
<code>&lt;sCriPt&gt;alert(1)&lt;/sCriPt&gt;</code>	<code>&lt;button onclick="document.location='default.asp'"&gt;HTML Tutorial&lt;/button&gt;</code>
<code>&lt;ScRiPt&gt;prompt(1)&lt;/ScRiPt&gt;</code>	
<code>&lt;a href="https://google.com"&gt;moh&lt;/a&gt;</code>	

# Vulnerability

