

Vulnerability Assessment & Penetration Testing

Reported Issued: 19/04/2024

Submitted By: Muhammed Irshad

Sensitive: The information in this document is strictly confidential and is made by Muhammed Irshad

Confidentiality Notice

This report contains sensitive, privileged and confidential information. Precautions should be taken to protect the confidentiality of the information in this document.

Disclaimer

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope of the engagement. This report is a summary of the findings from a “point-in-time” assessment made on six different client environments. Any changes made to the environment during the period of testing may affect the results of the assessment.

Table Of Contents

1. Confidentiality Notice
2. Disclaimer
3. Executive Summary
 - a. Scope
4. Testing Methodology
5. Classification
 - a. Risk Classification
6. Assessment Findings
 - a. Vulnerability #1
 - b. Vulnerability #2
 - c. Vulnerability #3
 - d. Vulnerability #4
 - e. Vulnerability #5
 - f. Vulnerability#6
7. Appendix A: Tools Used
8. Appendix B: Engagement Information
 - a. Contact Information
9. Conclusion

Executive Summary

I performed a security assessment on six different web applications. The purpose of this assessment was to discover and identify vulnerabilities in the six websites' infrastructure and suggest methods to remediate the vulnerabilities and identified a total of six vulnerabilities within the scope of the engagement which are broken down by severity in the table below.

Critical	High	Medium	Low
2	1	2	1

The highest severity vulnerabilities give potential attackers the opportunity in confidential data being deleted, lost or stolen; websites being defaced; unauthorized access to systems or accounts and, ultimately, compromise of individual machines or entire networks. In order to ensure data confidentiality, integrity, and availability, security remediations should be implemented as described in the security assessment findings.

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope. Any changes made to the environment during the period of testing may affect the results of the assessment.

Scope

Security assessment includes testing for security loopholes in the scope defined below. Apart from the following, no other information was provided. Nothing was assumed at the start of the security assessment. The following was the scope covered under the security audit:

Web Application 1: <https://account.lenovo.com/my/en/signin>

Web Application 2: <https://www.visitsaudi.com/en>

Web Application 3: <https://www.1shoppingcart.com>

Web Application 4: <https://oxygendigitalshop.com/>

Web Application 5: <https://mohua.gov.in/>

Web Application 6: <https://www.realme.com/in/>

Testing Methodology

My testing methodology was split into three phases: Reconnaissance, Target Assessment, and Discovering Vulnerabilities. During reconnaissance, we gathered information about the web applications. I gathered evidence of vulnerabilities during this phase of the engagement in a manner that would not disrupt normal business operations.

The following image is a graphical representation of this methodology.



Classification

Risk Classification

Level	Score	Description
Critical	10	The vulnerability poses an immediate threat to the organization. Successful exploitation may permanently affect the organization. Remediation should be immediately performed.
High	7-9	The vulnerability poses an urgent threat to the organization, and remediation should be prioritized.
Medium	4-6	Successful exploitation is possible and may result in notable disruption of business functionality. This vulnerability should be remediated when feasible.
Low	1-3	The vulnerability poses a negligible/minimal threat to the organization. The presence of this vulnerability should be noted and remediated if possible.
Informative	0	These findings have no clear threat to the organization, but may cause business processes to function differently than desired or reveal sensitive information about the company.

Assessment Findings

Number	Findings	CVSS	Severity
1	Authentication Bypass	9.1	Critical
2	Cross Site Scripting (XSS)	9	Critical
3	No Rate Limit	7	High
4	Parameter Tampering	6.5	Medium
5	Secure Flag Missing	4.2	Medium
6	Clickjacking	3.1	Low

Vulnerability #1

Critical Risk (9.1/10)	
Name of Vulnerability	Authentication Bypass
Security Impact	Severe

Vulnerable URL

<https://account.lenovo.com/my/en/signin>

Security Implications

Authentication bypass, which is often a consequence of broken authentication and session management, poses significant security implications for applications. Authentication bypass vulnerabilities allow an attacker to gain access to an account without having to go through the application's authentication procedure.

Authentication bypass often occurs through logic flaws and incomplete implementation of authentication mechanisms. Bypassing the authentication mechanisms of this application allows an attacker to view or edit data or other user's permissions, take over user accounts, access unauthorized endpoints, or expose critical data, depending on the authorization of the account they gain access to.

Here are some specific security risks associated with authentication bypass:

- **Unauthorized Access:**
Authentication bypass allows attackers to gain access to the application or system without providing valid credentials. This can lead to unauthorized use of sensitive features, data, or functionalities.
- **Data Exposure:**

Attackers gaining unauthorized access through authentication bypass may have access to sensitive user data, confidential information, or intellectual property. This can result in data breaches and compromise the confidentiality of information.

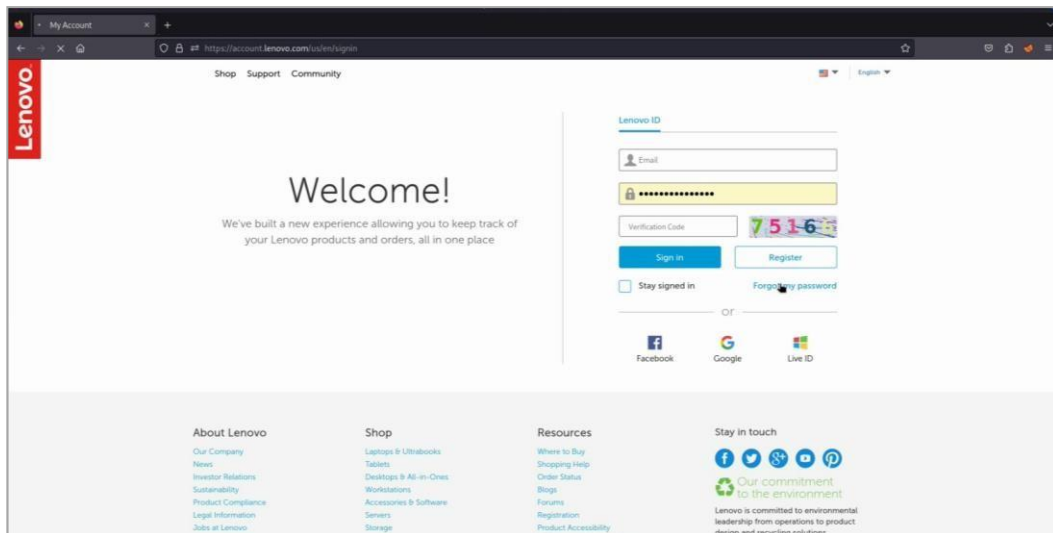
- **Account Takeover:**
Authentication bypass may lead to account takeover, where attackers gain control over user accounts. This can enable them to impersonate legitimate users, make unauthorized changes, or perform malicious actions on behalf of the compromised accounts.
- **Privilege Escalation:**
Once authenticated, attackers may attempt to escalate their privileges within the application or system. This could lead to unauthorized access to administrative features, sensitive data, or other high-privilege functionalities.
- **Financial Implications:**
Authentication bypass can have financial consequences for both users and organizations. For example, attackers may initiate fraudulent transactions or make unauthorized purchases using compromised accounts.
- **Reputation Damage:**
Security breaches resulting from authentication bypass can severely damage the reputation of an organization. Users and customers may lose trust if their accounts are compromised, leading to a negative impact on the organization's brand.

- **Legal Consequences:**
Organizations may face legal consequences if authentication bypass leads to the exposure of sensitive personal information or if it violates data protection regulations. Non-compliance with data protection laws can result in legal actions and financial penalties.
- **Data Integrity Issues:**
Attackers with unauthorized access may tamper with data, leading to integrity issues. This can have serious consequences, especially in applications where the accuracy and reliability of data are crucial.
- **Disruption of Services:**
Authentication bypass can be exploited to disrupt the normal functioning of an application or service. For example, attackers may gain access to administrative controls and disrupt services, leading to downtime and loss of availability.
- **Long-Term Consequences:**
Authentication bypass vulnerabilities, if left unaddressed, can have long-term consequences for an organization. As attackers continue to exploit such vulnerabilities, the overall security posture of the application may deteriorate.

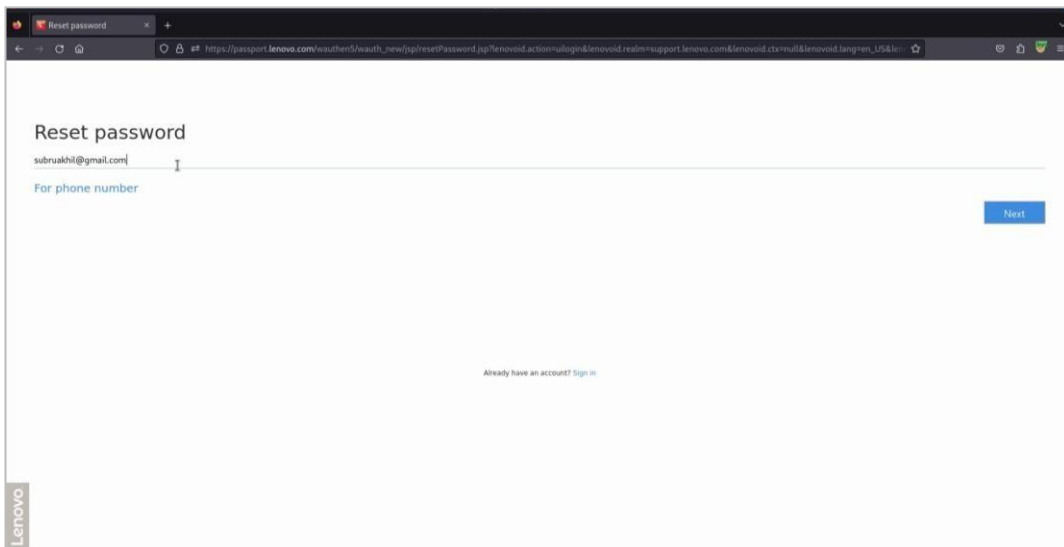
To mitigate the security implications of authentication bypass and broken authentication, it is essential to regularly assess and update authentication mechanisms, enforce secure session management practices, conduct security audits, and promptly address identified vulnerabilities. Additionally, organizations should stay informed about emerging security threats and continuously improve their security measures to adapt to evolving risks.

Steps to Reproduce

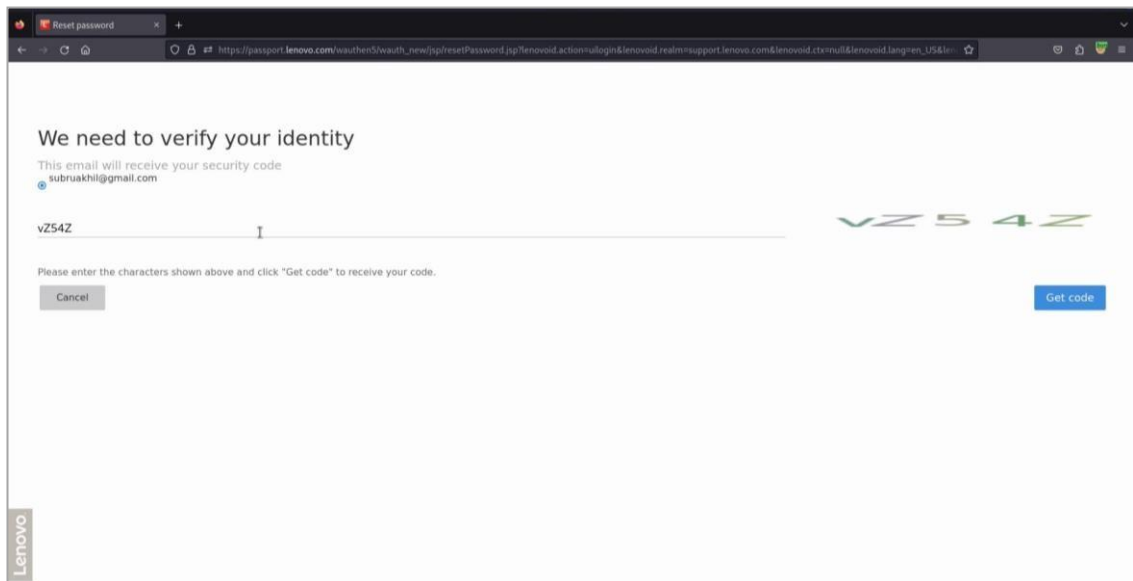
1. Navigate to: { <https://account.lenovo.com/us/en/signin> } and click 'Forgot Password' option.



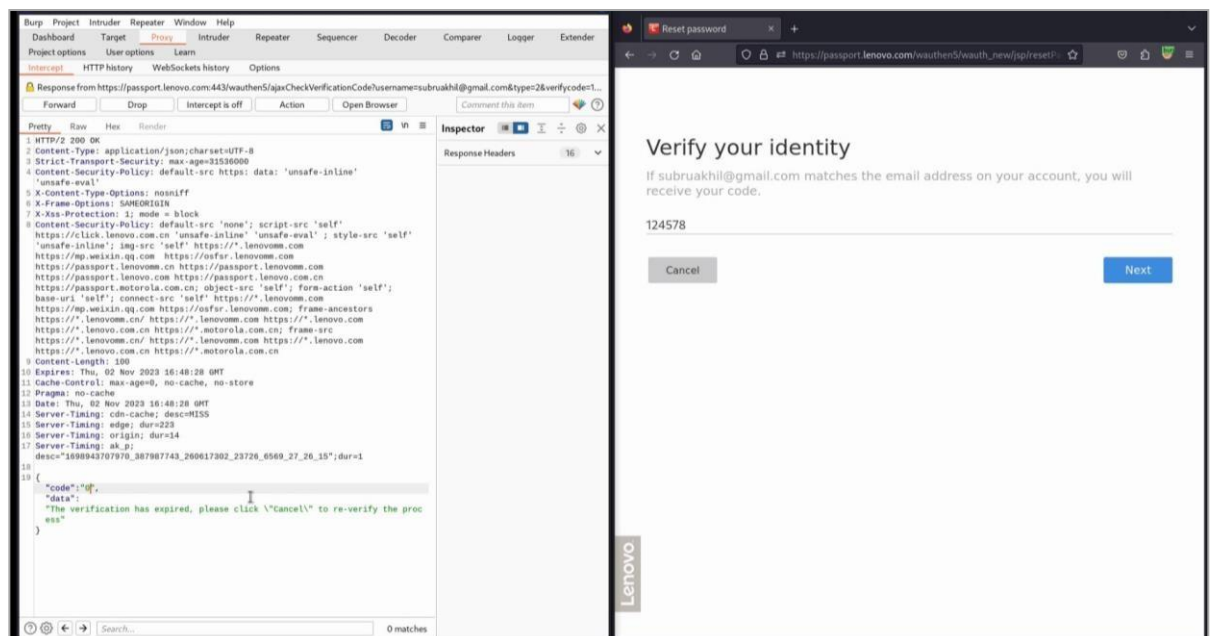
2. It will redirect to Reset Password page:
{ https://passport.lenovo.com/wauthen5/wauth_new/jsp/resetPassword.jsp?lenovoid.action=uilogin&lenovoid.realm=support.lenovo.com&lenovoid.ctx=null&lenovoid.lang=en_US&lenovoid.uinfo=null&lenovoid.cb=https%3A%2F%2Faccount.lenovo.com%2Fus%2Fen%2Fsignin }
3. After entering target email, click 'Next' option.



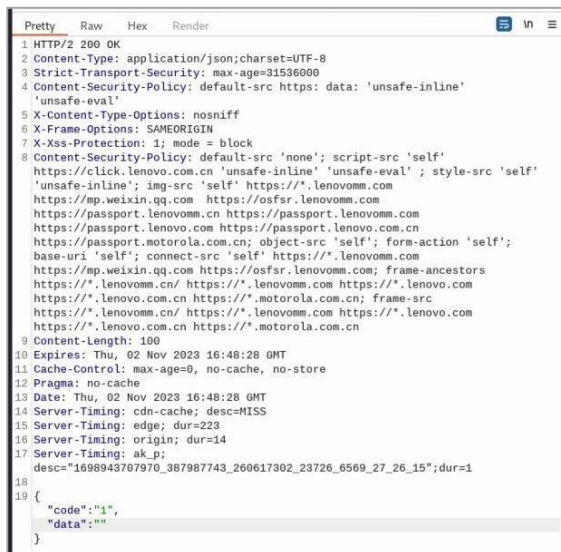
4. Then enter given captcha and click 'Get code' option.



5. Then they asked for entering the code we received on mail, so we enter a random code.
6. Enable a HTTP intercept proxy, such as Burp Suite to record and intercept web traffic from my browse.



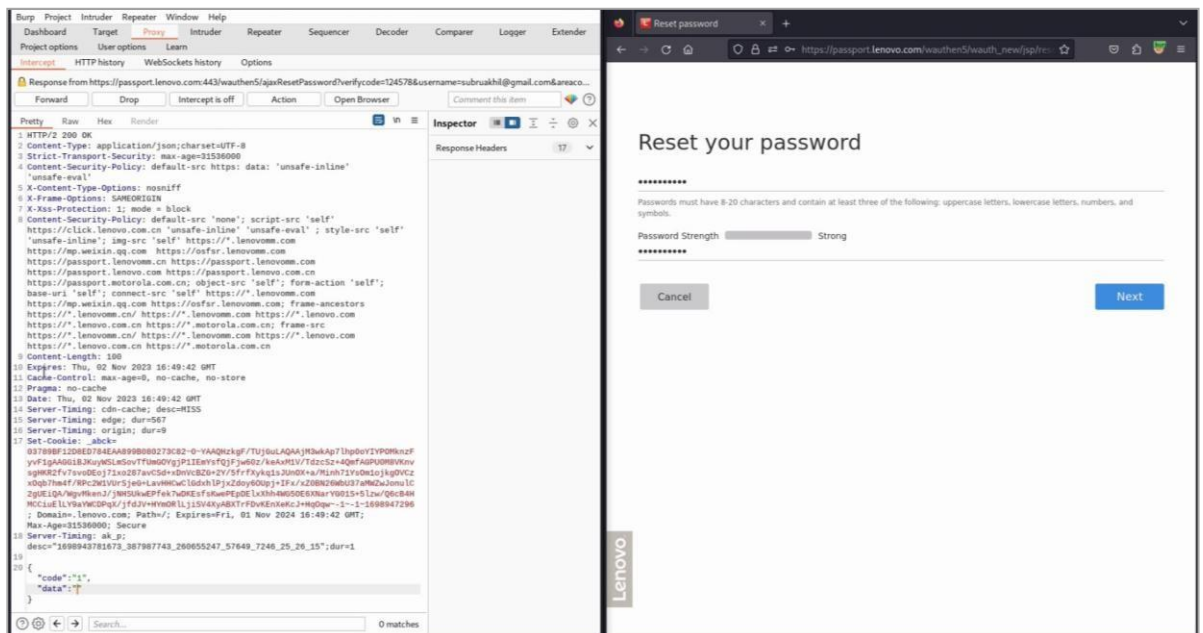
7. Observe the request captured on Burp-Suite and changed to "code": "1" & "data": "" into the JSON in the request body and forward it.



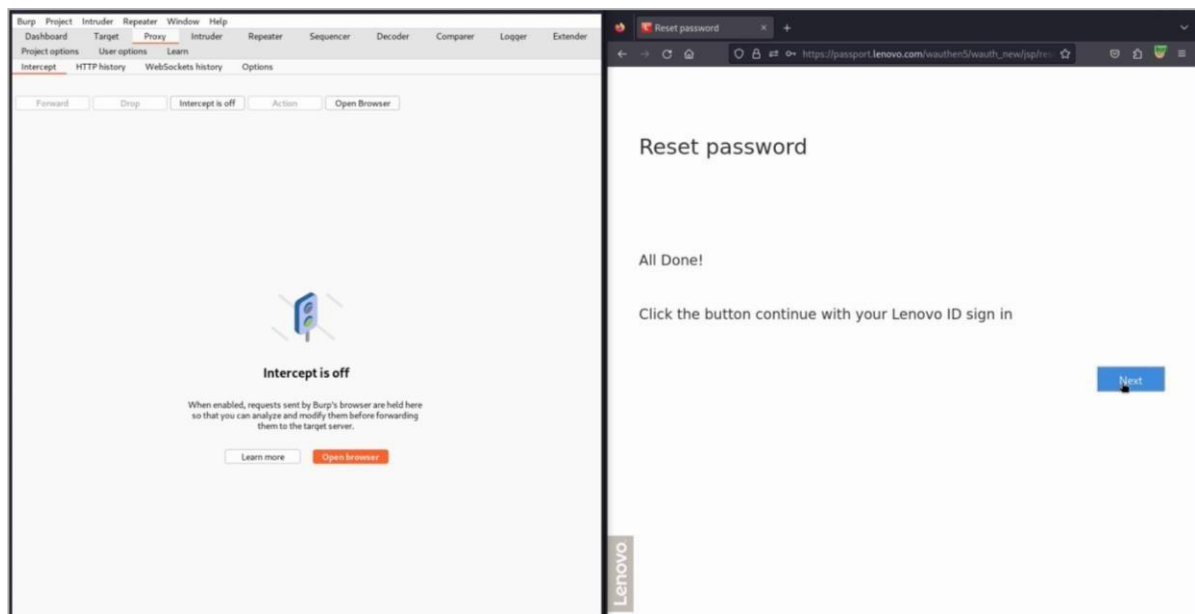
```
1 HTTP/2 200 OK
2 Content-Type: application/json;charset=UTF-8
3 Strict-Transport-Security: max-age=31536000
4 Content-Security-Policy: default-src https: data: 'unsafe-inline' 'unsafe-eval'
5 X-Content-Type-Options: nosniff
6 X-Frame-Options: SAMEORIGIN
7 X-Xss-Protection: 1; mode = block
8 Content-Security-Policy: default-src 'none'; script-src 'self' https://click.lenovo.com.cn 'unsafe-inline' 'unsafe-eval'; style-src 'self' 'unsafe-inline'; img-src 'self' https://*.lenovomm.com https://mp.weixin.qq.com https://osfsr.lenovomm.com https://passport.lenovomm.cn https://passport.lenovomm.com https://passport.lenovo.com https://passport.lenovo.com.cn https://passport.motorola.com.cn; object-src 'self'; form-action 'self'; base-uri 'self'; connect-src 'self' https://*.lenovomm.com https://mp.weixin.qq.com https://osfsr.lenovomm.com; frame-ancestors https://*.lenovomm.cn/ https://*.lenovomm.com https://*.lenovo.com https://*.lenovo.com.cn https://*.motorola.com.cn; frame-src https://*.lenovomm.cn/ https://*.lenovomm.com https://*.lenovo.com https://*.motorola.com.cn https://*.motorola.com.cn
9 Content-Length: 100
10 Expires: Thu, 02 Nov 2023 16:48:28 GMT
11 Cache-Control: max-age=0, no-cache, no-store
12 Pragma: no-cache
13 Date: Thu, 02 Nov 2023 16:48:28 GMT
14 Server-Timing: cdn-cache; desc=MIS
15 Server-Timing: edge; dur=223
16 Server-Timing: origin; dur=14
17 Server-Timing: ak_p; desc="1698943787970_387987743_260617362_23726_6569_27_26_15";dur=1
18
19 {
20   "code": "1",
21   "data": ""
22 }
```

8. After forwarding the response request, the page redirect to 'Reset password page'. There we can add new password and click 'Next' option.

9. Again, we capture the request on Burp-Suite and changed to "code": "1" & "data": "" into the JSON in the request body & then forward it.



10. Reset password successfully done page will appear.



Impact

Authentication bypass can lead to data loss or theft through an attacker's access to data. The severity of which is dependent on the sensitivity of the data within the application. It can also result in reputational damage to the application or the company due to legitimate users not trusting the security of the application if the application's data becomes publicly available.

References

- https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication
- <https://portswigger.net/web-security/authentication>

Vulnerability #2

Critical Risk (9/10)	
Name of Vulnerability	Cross Site Scripting (XSS)
Security Impact	Severe

Vulnerable URL

<https://www.visitsaudi.com/en>

Security Implications

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.

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.

Cross-Site Scripting (XSS) attacks occur when:

1. Data enters a Web application through an untrusted source, most frequently a web request.
2. The data is included in dynamic content that is sent to a web user without being validated for malicious content.

The malicious content sent to the web browser often takes the form of a segment of JavaScript, but may also include HTML, Flash, or any other type of code that the browser may execute. The variety of attacks based on XSS is almost limitless, but they commonly include transmitting private data, like cookies or other session information, to the attacker, redirecting the victim to web content controlled by the attacker, or performing other malicious operations on the user's machine under the guise of the vulnerable site.

Reflected XSS Attacks

Reflected attacks are those where the injected script is reflected off the web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. Reflected attacks are delivered to victims via another route, such as in an e-mail message, or on some other website. Reflected XSS is also sometimes referred to as Non-Persistent or Type-I XSS (the attack is carried out through a single request / response cycle).

Stored XSS Attacks

Stored attacks are those where the injected script is permanently stored on the target servers, such as in a database, in a message forum, visitor log, comment field, etc. The victim then retrieves the malicious script from the server when it requests the stored information. Stored XSS is also sometimes referred to as Persistent or Type-II XSS.

Blind Cross-site Scripting

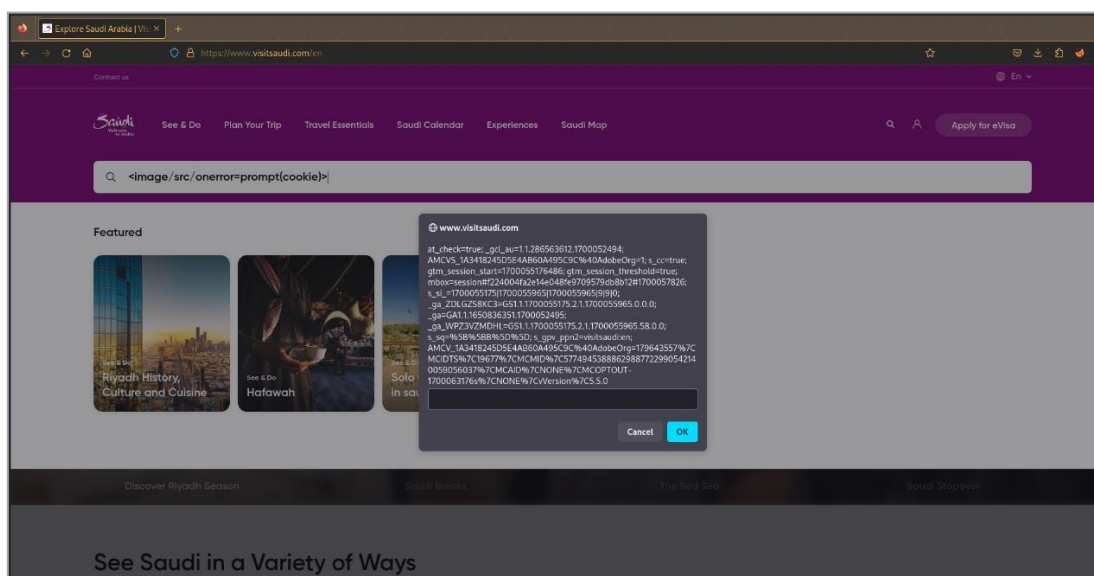
Blind Cross-site Scripting is a form of persistent XSS. It generally occurs when the attacker's payload is saved on the server and reflected back to the victim from the backend application. For example, in feedback forms, an attacker can submit the malicious payload using the form, and once the backend user/admin of the application will open the attacker's submitted form via the backend application, the attacker's payload will get executed.

Other Types of XSS Vulnerabilities

In addition to Stored and Reflected XSS, another type of XSS, DOM Based XSS was identified by Amit Klein in 2005

Steps to Reproduce

1. Go to the target URL <https://www.visitsaudi.com/en>
2. Click on the search tab and enter the following:
<image/src/onerror=prompt(cookie)>
3. These shows a popup in the corresponding website.



Impact

An attacker could steal credentials. An attacker could exfiltrate sensitive data. An attacker can steal cookies and Sessions. An attacker can quickly obtain access to your other client's computers.

Reflected XSS could lead to data theft through the attacker's ability to manipulate data through their access to the application, and their ability to interact with other users, including performing other malicious attacks, which would appear to originate from a legitimate user. These malicious actions could also result in reputational damage for the business through the impact to customers trust.

References

- <https://owasp.org/www-community/attacks/xss/>
- <https://portswigger.net/web-security/cross-site-scripting>

Vulnerability #3

High Risk (7/10)	
Name of Vulnerability	No Rate Limit
Security Impact	High

Vulnerable URL

<https://www.1shoppingcart.com>

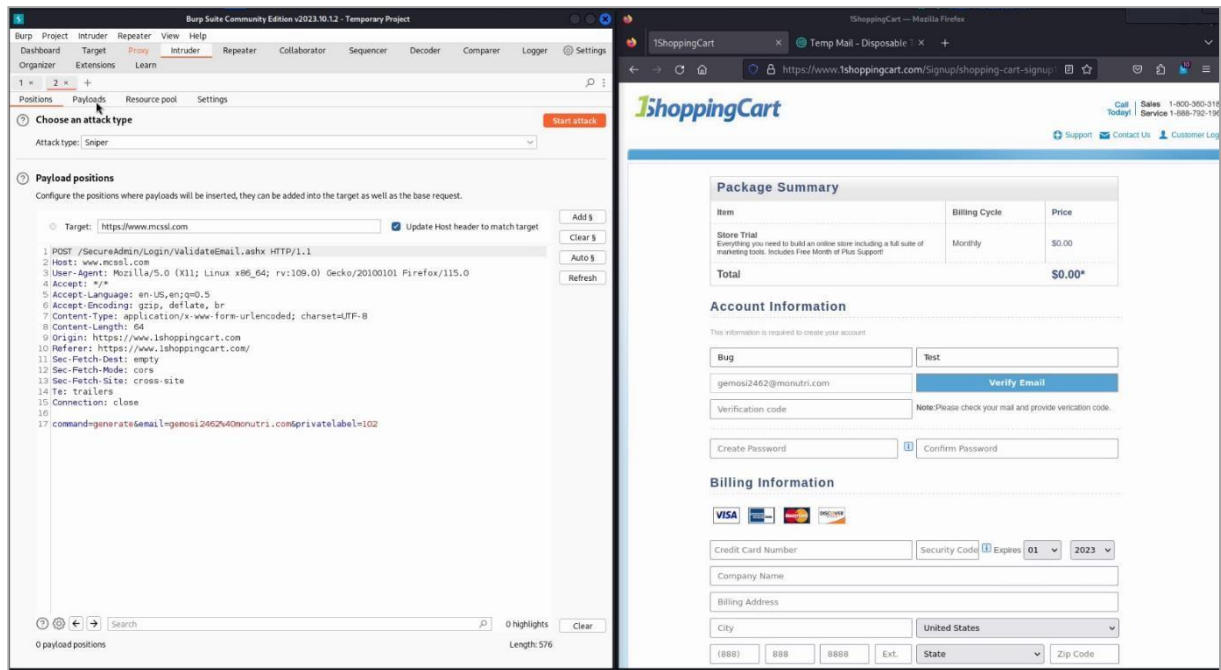
Security Implications

Rate Limiting prevents an application from becoming unresponsive or unavailable due to too many requests exhausting the application's resources. A lack of rate limiting on an email triggering endpoint was identified. This allows an attacker to create a large amount of emails to any email address, which they could use to spam a target with emails.

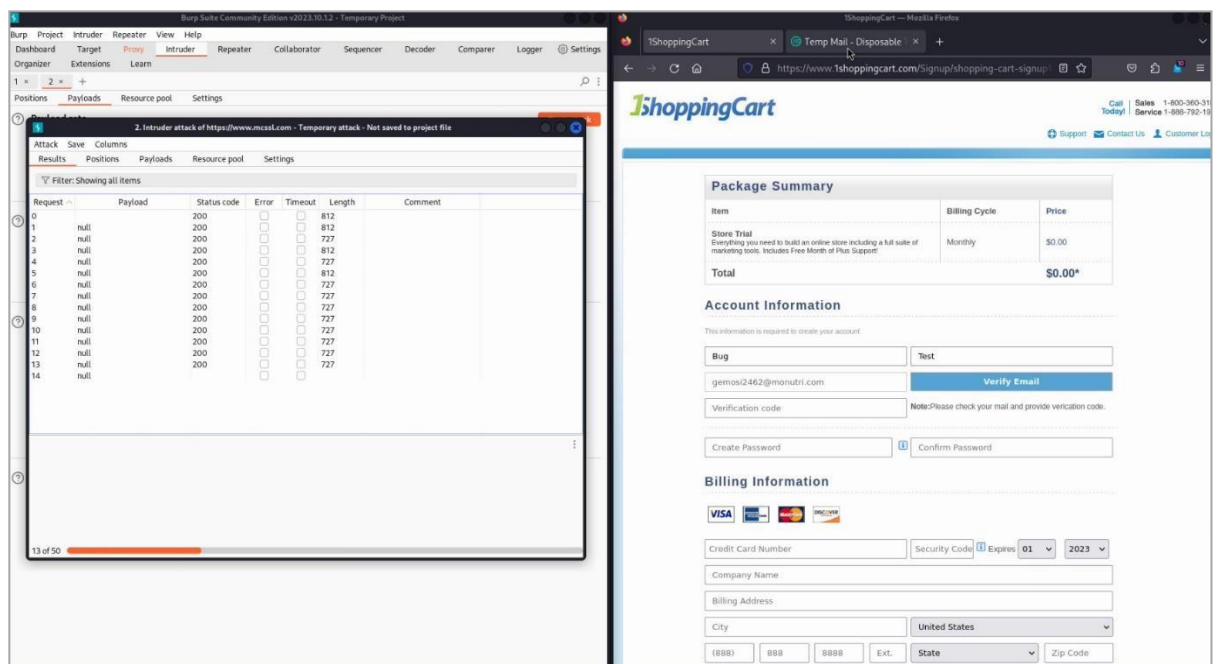
Steps to Reproduce

1. Go to the URL <https://www.1shoppingcart.com> and click 'Try It Free!' option.
2. It will redirect to Sign Up page. Enter First & Last name and enter the email address, here I have used temporary email from {<https://temp-mail.org/en/>}: “gemosi2462@monutri.com”.

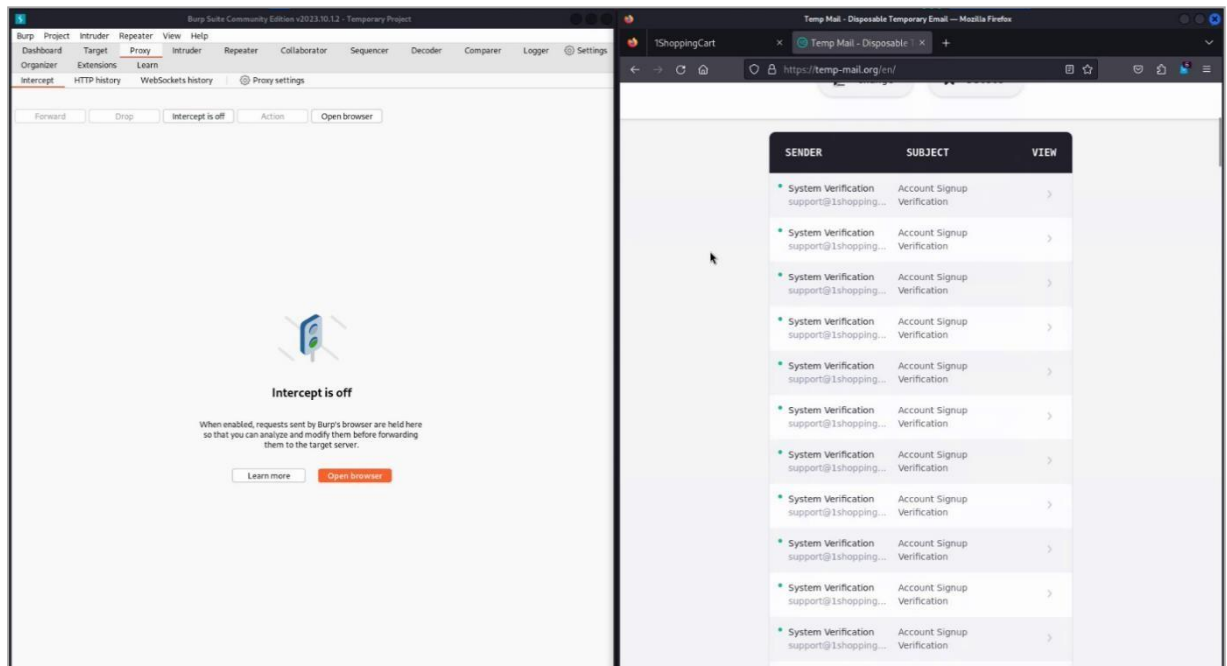
- Now turn on the Burp Suite, then click “Verify Email” option and intercept the request, and set a null payload using the intruder and test for a no rate limit vulnerability.



- After setting payload, click on ‘Start Attack’ and after finishing the attack, forward the request to run the exploit.



5. Then turn off the Burp suite Interceptor and check the inbox for the mails.



Impact

The impact can range from something like DOS up to enable authentication attacks, these are all in the higher end of the impact range because they have some serious potential to disrupt the normal workings of an API.

References

- <https://snapsec.co/blog/Attacking-Rate-limit/#:~:text=I%20mpact,critical%20when%20misused%20by%20attackers>
- <https://www.geeksforgeeks.org/no-rate-limiting-flaw-in-cyber-security/>

Vulnerability #4

Medium Risk (6.5/10)	
Name of Vulnerability	Parameter/Price Tampering
Security Impact	Medium

Vulnerable URL

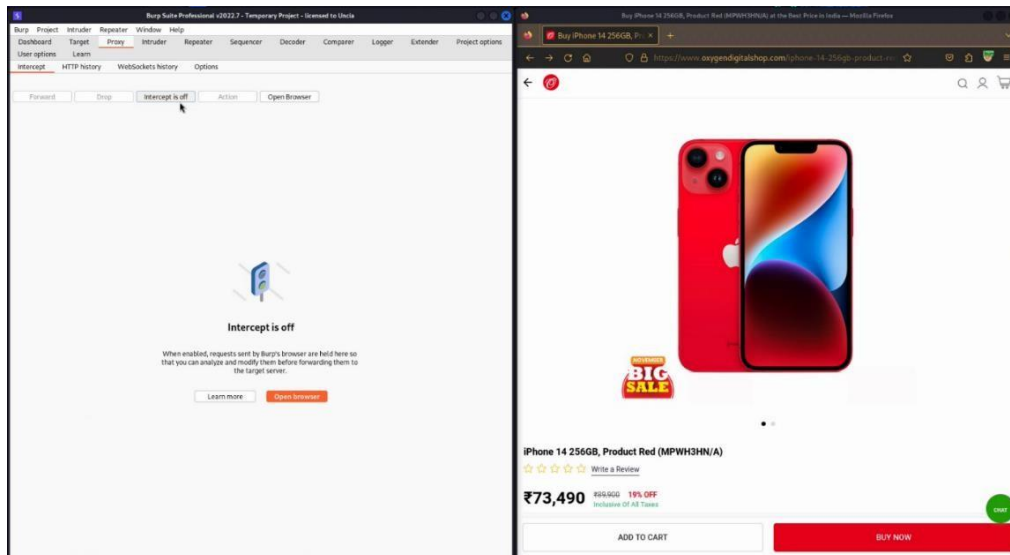
<https://oxygendigitalshop.com/>

Security Implications

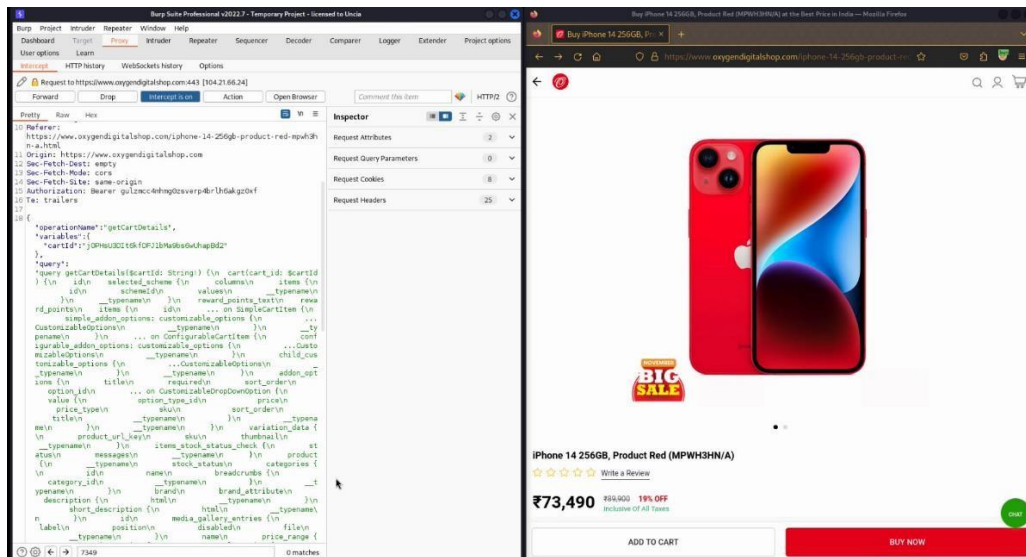
Parameter tampering, also known as input manipulation, is a type of attack where an attacker alters parameters in a request to a web application in order to gain unauthorized access, bypass security mechanisms, or manipulate the application's behavior. Price tampering or manipulation, often associated with parameter tampering in e-commerce poses serious security risks. Unauthorized changes to product prices can lead to financial losses, erode customer trust, and damage the reputation of the business. Fraudulent activities, such as unauthorized purchases at manipulated prices, may result in chargeback issues and legal consequences. The operational impact includes disruptions in inventory management and order processing. Violation of contracts, regulatory compliance issues, and potential legal actions amplify the business risks. Additionally, the compromise of business logic and negative customer experiences can lead to a loss of customer loyalty. Mitigation strategies should encompass robust security measures, regular audits, and transparent communication to maintain customer trust and protect the integrity of the e-commerce platform.

Steps to Reproduce

1. Use a browser to navigate to: { <https://oxygendigitalshop.com/> }
2. Login as User Account and select a product.



3. Enable a HTTP intercept proxy, such as Burp Suite to record and intercept web traffic from my browse.
4. Click “Buy Now” option and capture the response and do intercept.



5. Observe the insecure price parameters and change the price values & forward the request.

The screenshot displays the Burp Suite Professional v2022.7 interface on the left and a web browser on the right. The browser shows the product page for an iPhone 14 256GB, Product Red (MPWH3HN/A) on the website oxygendigitalshop.com. The price is listed as ₹73,490 with a 19% OFF discount. The Burp Suite interface shows the intercepted response from https://www.oxygendigitalshop.com:443/graphql. The JSON response includes a 'final_price' field with a value of 73490, which is highlighted by a red arrow. The 'Inspector' panel on the right shows the selected text '73490' and the 'Decoded from' dropdown set to 'Select'. The 'Response Headers' panel shows 23 headers.

The screenshot displays the Burp Suite Professional v2022.7 interface. The 'Inspector' panel shows the selected text '10' and the 'Decoded from' dropdown set to 'Select'. The 'Response Headers' panel shows 23 headers. The JSON response includes a 'grand_total' field with a value of 10, which is highlighted by a red arrow. The 'Inspector' panel shows the selected text '10' and the 'Decoded from' dropdown set to 'Select'. The 'Response Headers' panel shows 23 headers.

6. The page is redirected to Delivery Address and Price Details page. Here we can observe that the total price changed. Then click “Continue” option.

Delivery Address

☒ Same billing address
☐ Different billing address

☐ Use GST Invoice

Apply Discount Code

Enter Coupon Code

Price Details

Total Price	₹0
Estimated Tax	₹0
Estimated Shipping	FREE
Total	₹10

Prices are inclusive of all taxes

Ways to Pay

7. The Page will redirect to Payment Option page. Here also the Total price is changed. Then Select payment method.
8. Click “Pay___” option and capture the response and do intercept.
9. Observe the insecure amount parameter and change the amount value.

Burp Suite Professional v2022.7 - Temporary Project - licensed to Uncia

Response from https://www.oxygendigitalshop.com:443/graphql [172.67.155.83]

```
{
  "data": {
    "placeOrder": {
      "order": {
        "order_id": "000016666",
        "order_number": "000016666",
        "action_url": "https://oxygendigitalshop.com/payu/standard/payfororder_id=16666",
        "payu": {
          "url": "https://secure.payu.in/_payment",
          "timestamp": "8026e9c723fcea1120",
          "key": "qk1xkh",
          "hash": "a0641703131e3ab041ea0d9f69d2998c570989a4c4aeafdb1be41db7890aee62e77436379b63ae999abb8057a3650caa8c983a1423ad9397a17c43a133187b",
          "amount": 10,
          "productinfo": "000016666",
          "firstname": "AkhiL",
          "lastname": "Subrahmanyam",
          "city": "Thiruvananthapuram",
          "state": "Kerala",
          "zip": "695001",
          "country": "IN",
          "email": "akhisu@proton.me",
          "udf5": "000016666",
          "phone": "91904805438",
          "curl": "https://oxygendigitalshop.com/payu/standard/cancel/",
          "furl": "https://oxygendigitalshop.com/payu/standard/response/",
          "eurl": "https://oxygendigitalshop.com/payu/standard/response/"
        },
        "typename": "payu_biz"
      },
      "typename": "Order"
    },
    "typename": "PlaceOrderOutput"
  }
}
```

Payment Options

☐ PineLabs GateWay - EMI, Credit/Debit Cards

☒ HDFC GateWay - EMI, Credit/Debit Cards, UPI, NET banking

Apply Discount Code

Enter Coupon Code

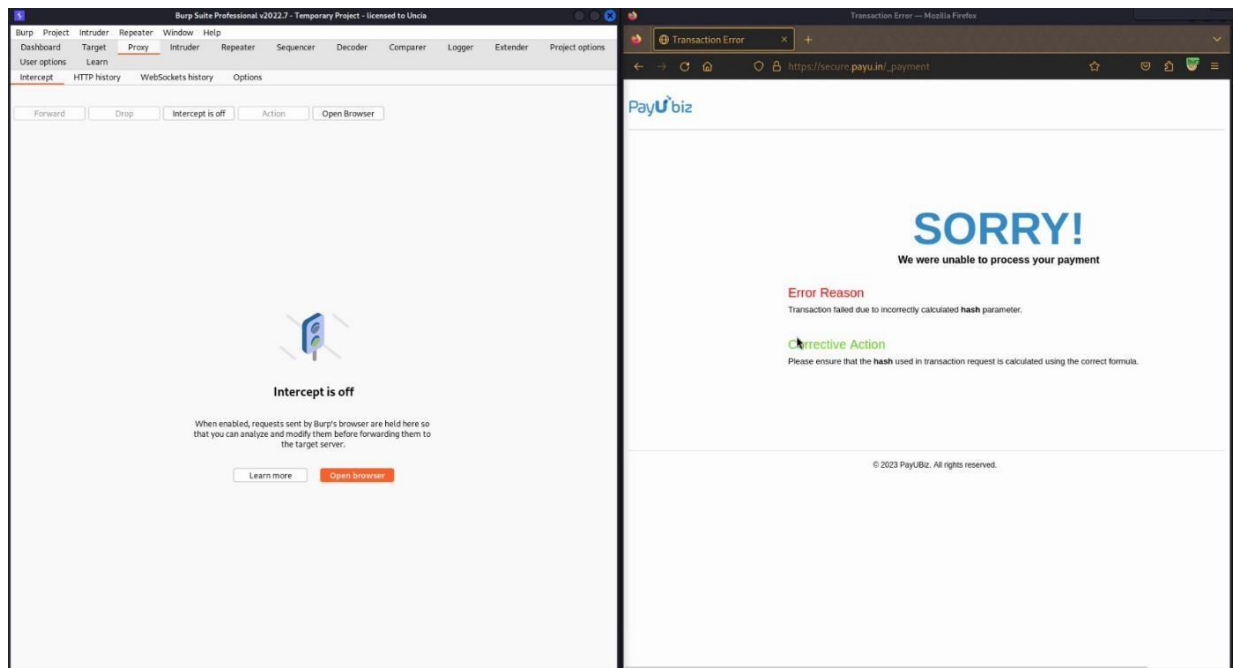
Price Details

Total Price	₹0
Estimated Tax	₹0
Estimated Shipping	FREE
Total	₹10

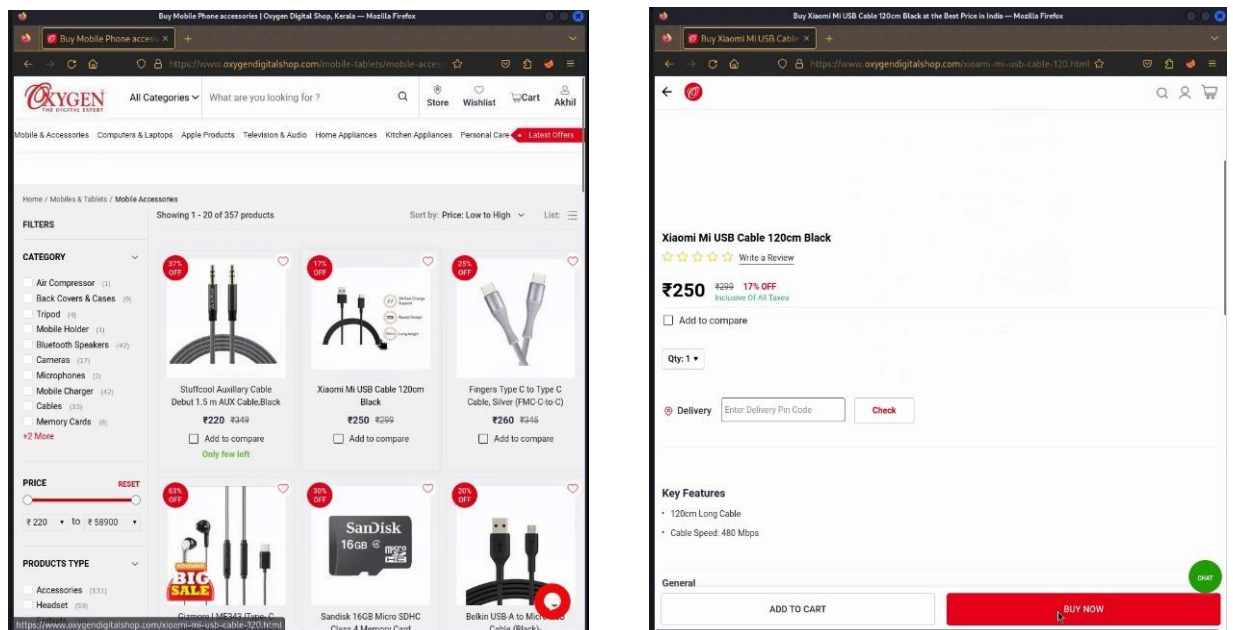
Prices are inclusive of all taxes

Ways to Pay

10. Forward, the request and so an error action occurred.



11. Back to home page and select a low-priced product and click “Buy Now”.



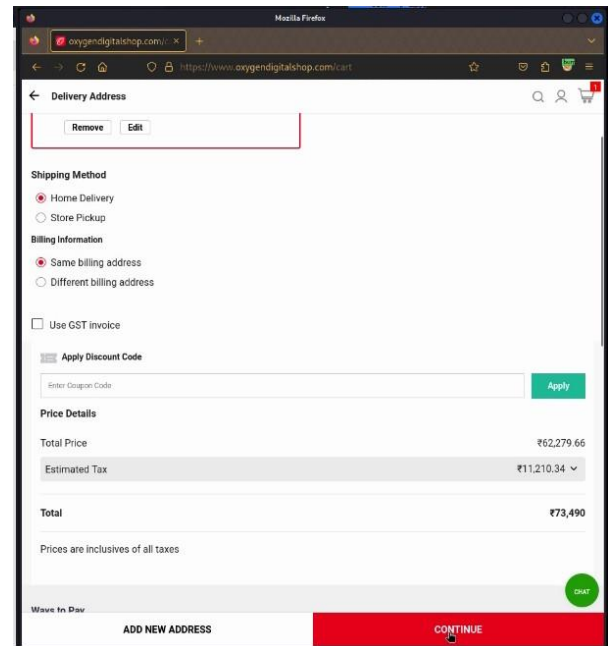
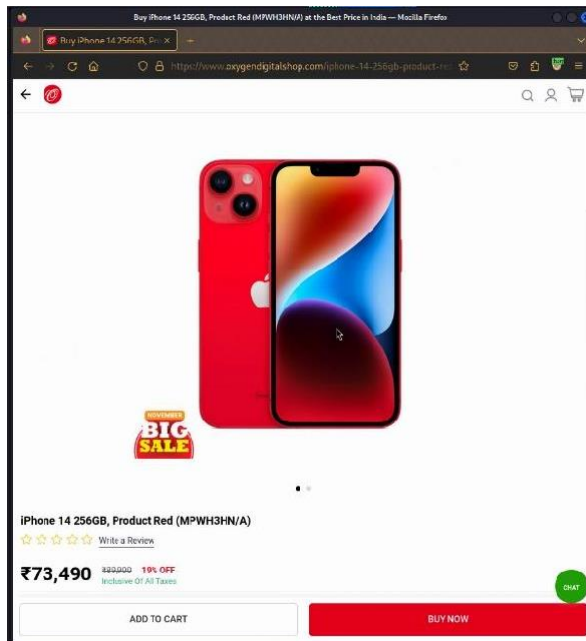
12. On payment option, click “Pay___” option and capture the response and do intercept.

The screenshot shows two windows. On the left is Burp Suite Professional v2022.7, with the 'Intercept' tab selected. The 'Request' pane shows a GET request to https://www.oxygendigitalshop.com/443. The 'Inspector' pane shows the request body, which is a JSON object. The 'Do intercept' option is checked. On the right is a Mozilla Firefox browser window showing the 'Payment Options' page of oxygendigitalshop.com. The page has a 'Pay ₹250' button. The 'Price Details' section shows a Total Price of ₹211.86 and an Estimated Tax of ₹38.14, totaling ₹250. The 'Ways to Pay' section lists various payment methods including Visa, RuPay, and HDFC Bank.

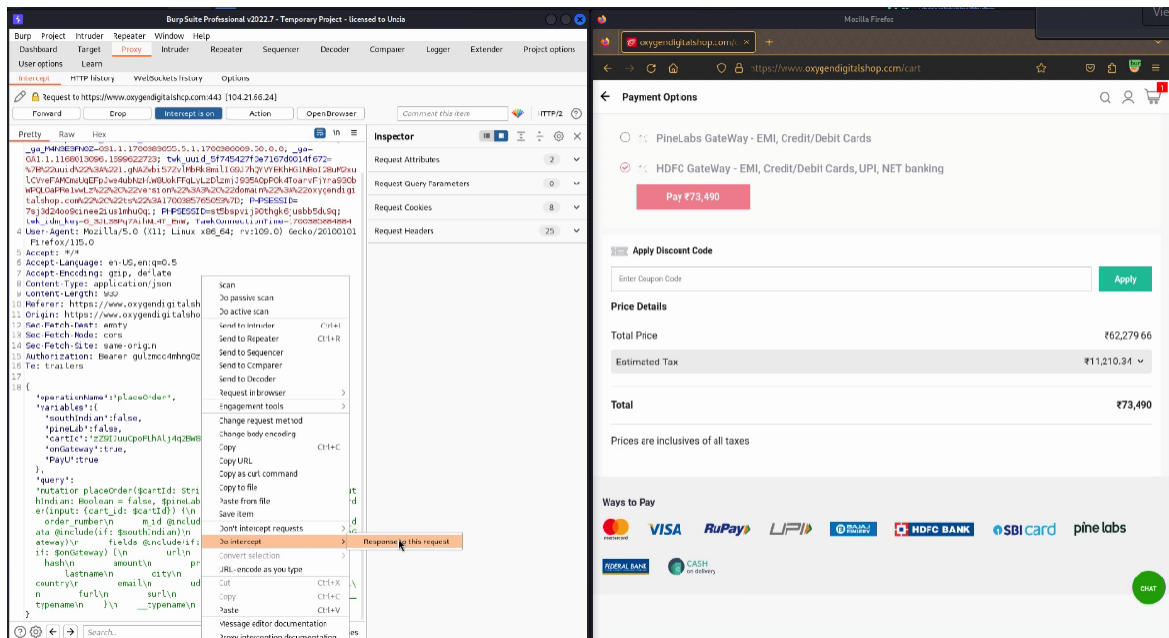
13. Copy the JSON code from the “Pay” option’s request body

The screenshot shows two windows. On the left is Burp Suite Professional v2022.7, with the 'Intercept' tab selected. The 'Response' pane shows a 200 OK response from https://www.oxygendigitalshop.com/443. The 'Inspector' pane shows the response body, which is a JSON object. The 'Decoded from' section shows the response body decoded from URL encoding. The 'Copy' button is highlighted. On the right is a Mozilla Firefox browser window showing the 'Payment Options' page of oxygendigitalshop.com. The page has a 'Pay ₹250' button. The 'Price Details' section shows a Total Price of ₹211.86 and an Estimated Tax of ₹38.14, totaling ₹250. The 'Ways to Pay' section lists various payment methods including Visa, RuPay, and HDFC Bank.

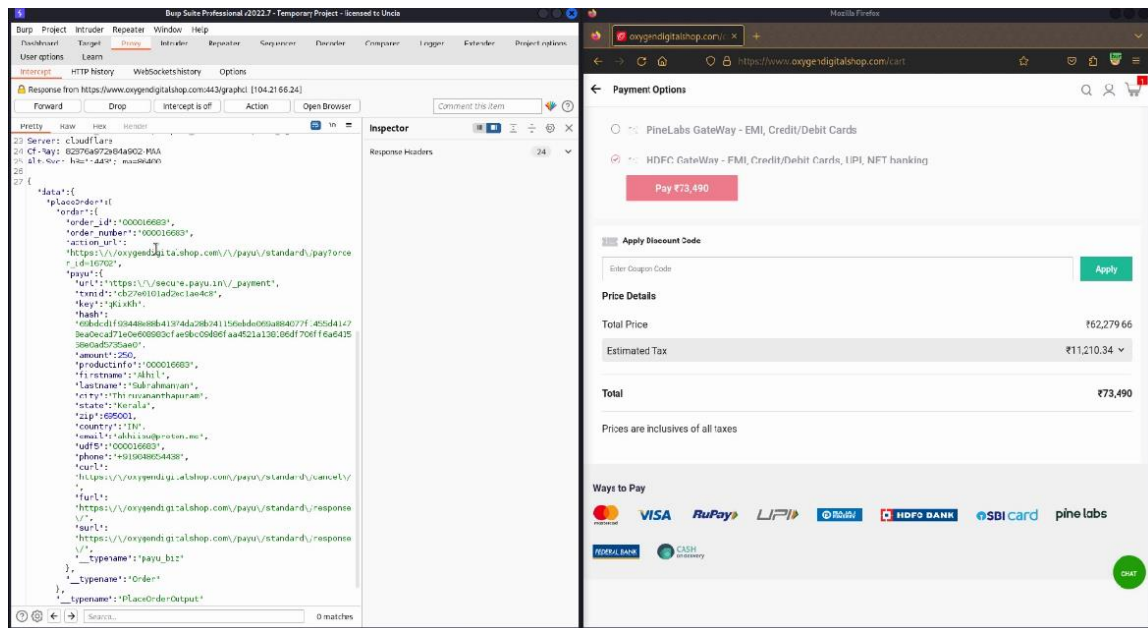
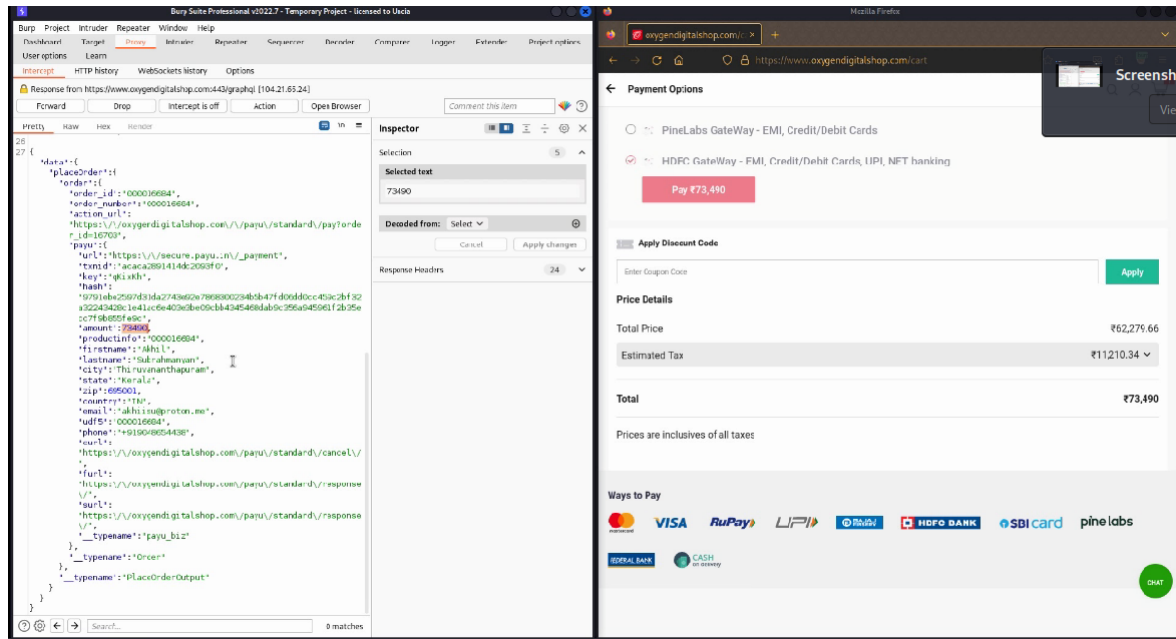
14. Then go home page and select a high-priced product and Click “Buy Now” option. The page is redirected to Delivery Address and Price Details page. Then click “Continue” option.



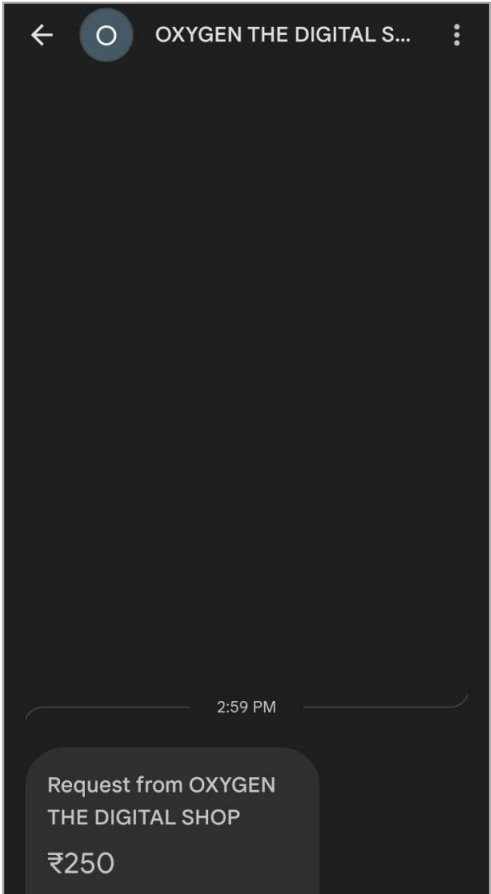
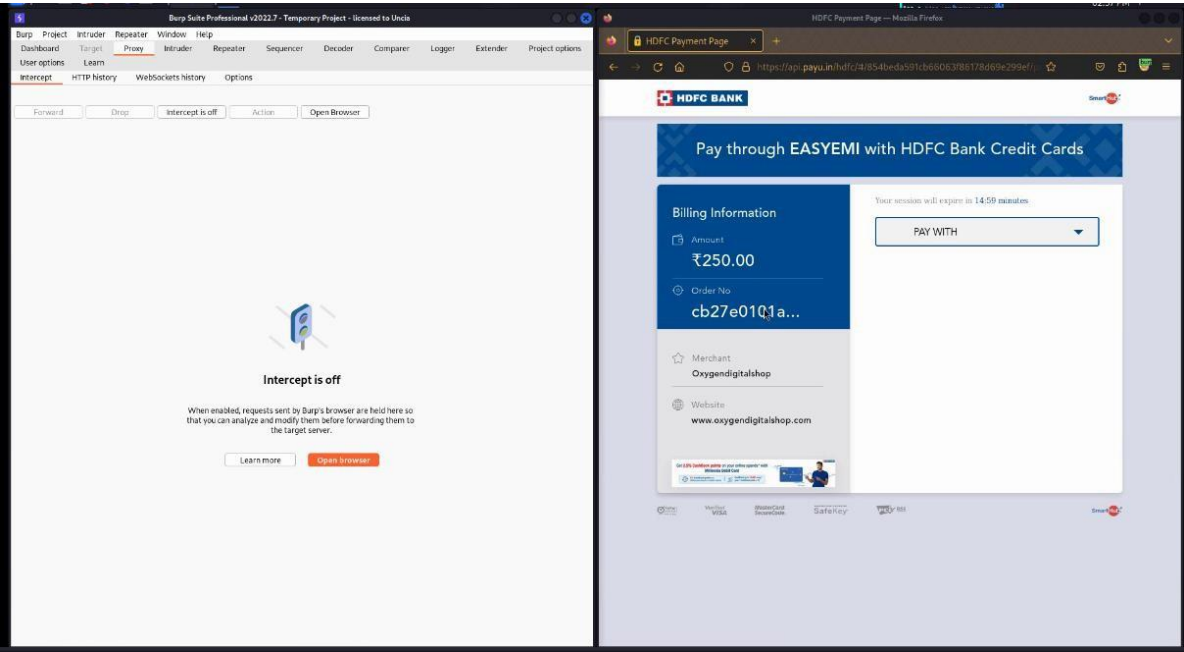
15. Click “Pay ” option and capture the response and do intercept.



16. Paste the copied code into high-priced product's JSON code on the “Pay” option's request body.



17. Forward the request and observes that the payment portal amount is changed to lower priced product's amount.



Impact

The impact of price tampering in an e-commerce context can be significant, affecting both the business and its customers. Price tampering can result in financial losses for the e-commerce platform as products may be sold at lower prices than intended. Customers rely on accurate and fair pricing. If they discover price manipulation, it can erode trust in the platform and they may choose to shop elsewhere, leading to a loss of customer loyalty. Price manipulation can damage the overall reputation of the e-commerce brand. Trust is a critical component of a successful e-commerce business. Unauthorized purchases at lower prices, leading to financial losses and potential chargeback issues. Price manipulation can disrupt inventory management, as the system may not accurately reflect the demand for products at manipulated prices also it can affect the normal flow of order processing, leading to operational challenges. Price manipulation may violate contracts with vendors or partners, leading to legal consequences and unfair pricing practices might violate consumer protection laws or regulations, resulting in legal action. News of price tampering can result in negative publicity, potentially spreading through social media and other channels, further harming the brand's reputation.

References

- https://owasp.org/www-community/attacks/Web_Parameter_Tampering
- <https://portswigger.net/web-security/access-control>

Vulnerability #5

Medium Risk (4.2/10)	
Name of Vulnerability	Secure Flag Missing
Security Impact	Medium

Vulnerable URL

<https://mohua.gov.in/>

Security Implications

If the secure flag is set on a cookie, then browsers will not submit the cookie in any requests that use an unencrypted HTTP connection, thereby preventing the cookie from being trivially intercepted by an attacker monitoring network traffic. If the secure flag is not set, then the cookie will be transmitted in clear-text if the user visits any HTTP URLs within the cookie's scope. An attacker may be able to induce this event by feeding a user suitable links, either directly or via another web site. Even if the domain that issued the cookie does not host any content that is accessed over HTTP, an attacker may be able to use links of the form `http://example.com:443/` to perform the same attack.

Steps to Reproduce

1. Go to the URL: `https://mohua.gov.in/`
2. After entering into the website perform inspect elements.
3. Then selected the storage field and evaluated the cookies.
4. Found `PHPSESSID:"2b15cf6b5012048d8bff34077a03592259b7cdab"`
Http Only and Secure flag were set as false.



- https://portswigger.net/kb/issues/00500200_tls-cookie-without-secure-flag-set
- <https://support.detectify.com/support/solutions/articles/%2048001048982-cookie-lack-secure-flag>

Vulnerability #5

Low Risk (3.1/10)	
Name of Vulnerability	Clickjacking
Security Impact	Low

Vulnerable URL

<https://www.realme.com/in/>

Security Implications

Clickjacking, also known as a “UI redress attack”, is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top-level page. Thus, the attacker is “hijacking” clicks meant for their page and routing them to another page, most likely owned by another application, domain, or both.

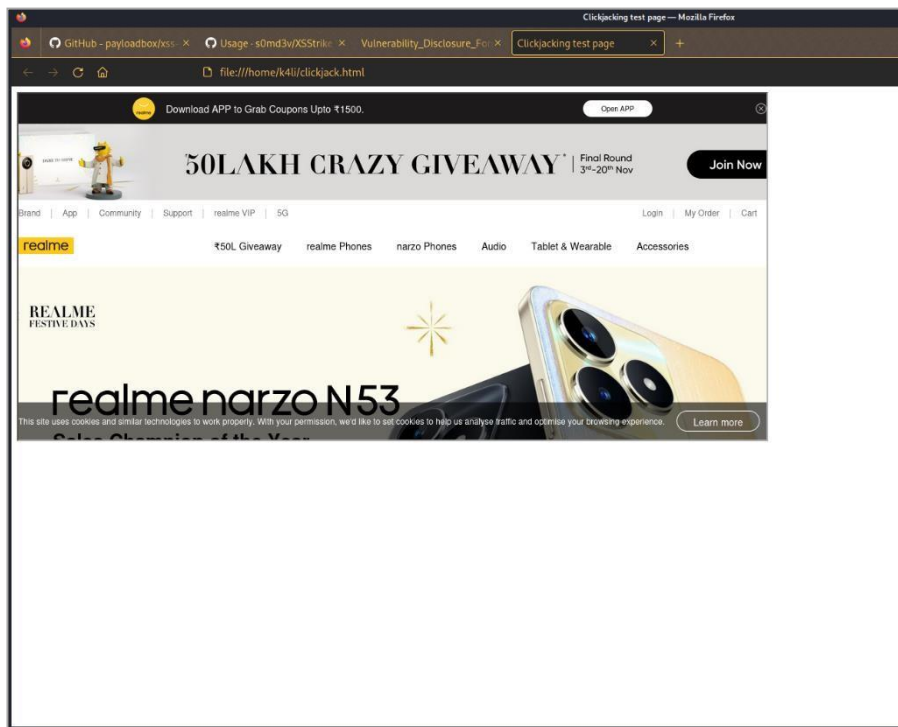
Steps to Reproduce

1. Get the URL of the site that needs to be tested for clickjacking vulnerability. Here it is: <https://www.realme.com/in/>
2. Now paste the URL as the ‘src’ on the clickjacking html code as below:



```
File Actions Edit View Help
GNU nano 7.2 clickjack.html
<html>
  <head>
    <title> Clickjacking test page</title>
  </head>
  <body>
    <iframe src="https://www.realme.com/in/" width="1080" height="500"></iframe>
  </body>
</html>
```

3. Now open this html file and we can see that the resolution of the webpage is changed.
4. Thus, it is vulnerable to clickjacking
5. The result will display as:



Impact

- Keystrokes can also be hijacked.
- With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

References

- <https://owasp.org/www-community/attacks/Clickjacking>
- <https://www.imperva.com/learn/application-security/clickjacking/>

Appendix A: Tools Used

Tool	Description
Burp Suite Professional	Used for Web Application Penetration Testing

TableA.1: Tools used during assessment

Appendix B: Engagement Information

Contact Information

Name	Muhammed Irshad
Phone	9633631740
Email	irshadmhd228@gmail.com , startechb0@eamsurn.com – temp mail

Conclusion

The primary goal is the identification of specific, documented vulnerabilities and their timely remediation. It's important to an organization with an Internet presence because attackers are able to take advantage of any loophole or flaw that may be present.

Vulnerability assessments also provide an organization with the necessary knowledge, awareness and risk backgrounds to understand and react to threats to its environment. A vulnerability assessment process is intended to identify threats and the risks they pose.