



Sample Penetration Test

Report

Date: xx-xx-xxxx

Table of Contents

1. INTRODUCTION	3
2. DISCLAIMER	3
3. SCOPE	3
4. CONTACT DETAILS	3
5. SEVERITY RATINGS	4
6. PENETRATION TESTING FINDINGS	5
6.1 Stored Cross Site Scripting	
6.2 Reflected XSS	
6.3 Unfiltered exif data...	
6.4 Session Invalidation	
6.5 No rate limiting leads to ATO	
6.6 Parameter Tempering/Business Logic	
6.7 Direct Object Access (IDOR)/Business Logic	
6.8 SQL injection	
7. DECLARATION	24

1. Introduction

This security report aims to provide a comprehensive assessment of the current security state, vulnerabilities, and potential risks within the system. By conducting a thorough analysis, we aim to identify areas of concern and offer actionable recommendations to enhance the overall security posture. The findings and insights presented in this report will help guide the implementation of effective security measures and mitigate potential threats.

2. Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the penetration test and not any changes or modifications made outside of that period.

3. Scope/Targeted vulnerabilities

1. SQL Injection
2. Insufficient Access Controls
3. Cross-Site Scripting
4. Direct Object Access
5. Insecure File Upload
6. Parameter Tampering
7. Insecure Business Logic

4. Contact Information

Name	Title	Contact Information
Researcher_name	Security analyst	Email: info@vulndetox.com

5. Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

**Tool Used: Burp Suite only.
(Manual Testing)**

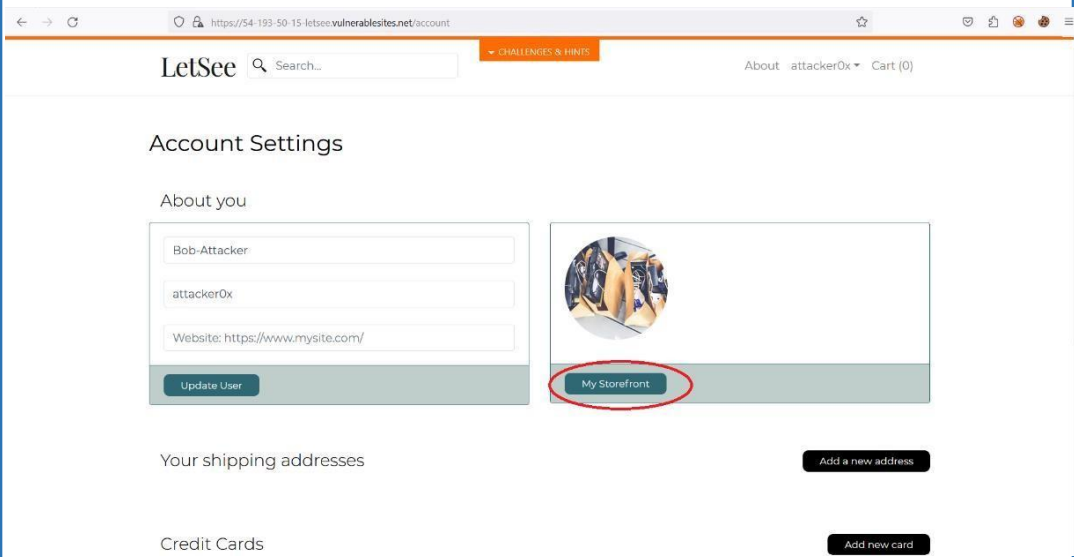
6. Penetration Test Findings

Vulnerability	6.1 Stored Cross Site Scripting (S-XSS)
Severity Level	Critical
Affected IP/URL	https://54-193-50-15-letsee.vulnerablesites.net/products/new
OWASP Category	Injection
Description	The website has a widespread and persistent vulnerability called "Stored Cross-Site Scripting" (XSS). It affects the entire site, causing XSS payloads with malicious code to continuously appear for every visitor. This vulnerability poses significant risks, including unauthorized data access, session hijacking, and potential malware distribution. Immediate action is required to mitigate this vulnerability and protect the website and its users.
Recommendation	Implement strict input validation and output encoding to prevent the execution of malicious scripts.
References	WSTG - v4.1 OWASP Foundation

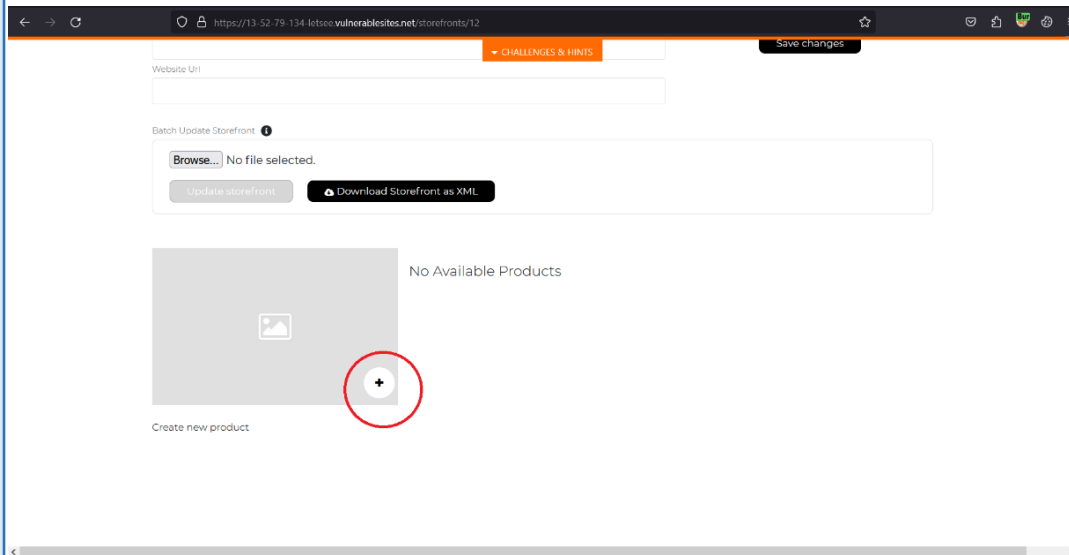
Step 1. Login into your account.

Step 2. Go to Profile, in Account Settings click on My Storefront.

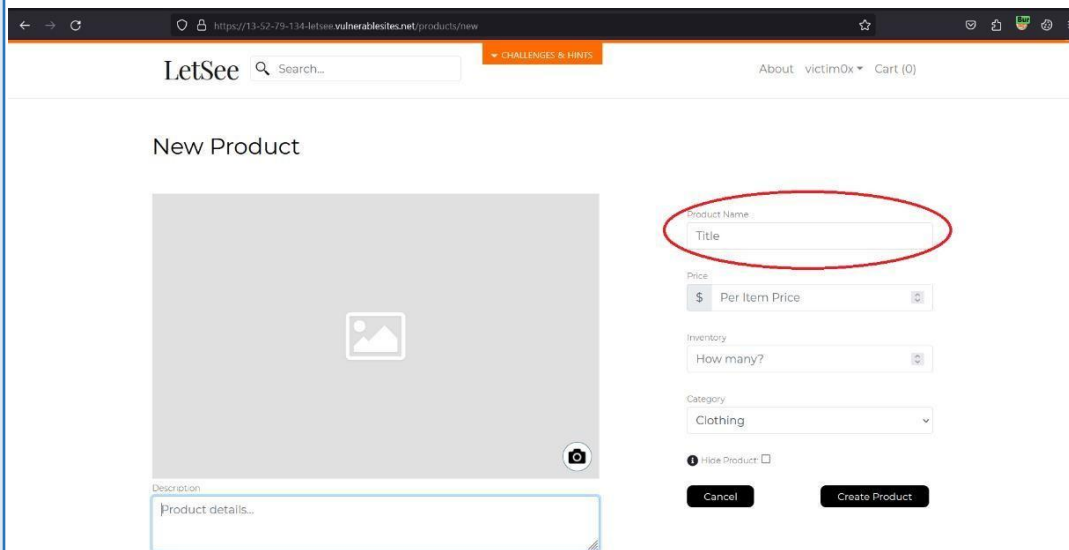
Proof of Concept



Step 3. Click on Manage Shop, and click on the + icon on the image tab.



Step 4. Use this payload in the Product Name parameter:
`</Textarea/</Noscript/</Pre/</Xmp><Svg /Onload=confirm(1)>`



Vulnerability	6.2 Reflected Cross Site Scripting (XSS)
Severity Level	High
Affected IP/URL	https://54-193-50-15-letsee.vulnerablesites.net/search?search=
OWASP Category	Injection
Description	Cross-Site Scripting (XSS) is a type of web vulnerability where attackers inject malicious scripts into a website, which are then executed by unsuspecting users, potentially compromising their sensitive information or enabling unauthorized activities.
Recommendation	Implement strict input validation and output encoding to prevent the execution of malicious scripts.
References	Cross-Site Scripting (XSS) OWASP Foundation

Step1. Go to the search box.



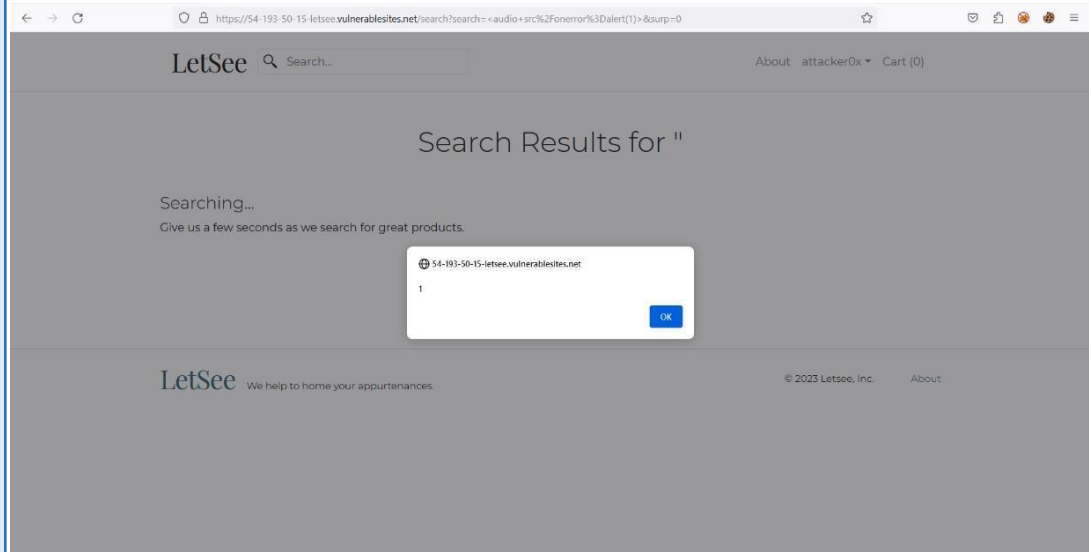
Proof of Concept



Step2. Use the payload:

```
<audio src/onerror=alert(1)>
```

Step3. The payload will pop-up!



Vulnerability	6.3 Unfiltered Exif Data (File upload)
Severity Level	Low
Affected IP/URL	https://54-193-50-15-letsee.vulnerablesites.net/storefronts/7
OWASP Category	File Upload
Description	The website has a vulnerability where unfiltered EXIF (Exchangeable Image File Format) data is exposed through image uploads. This means that when users upload images to the website, any potentially harmful or malicious data embedded in the EXIF metadata of the images is not properly filtered or sanitized.
Recommendation	It is crucial to implement proper validation and sanitization mechanisms for any uploaded images. This includes thoroughly checking and removing any potentially harmful or untrusted EXIF data before storing or using it within the website.
References	File Upload - OWASP Cheat Sheet Series

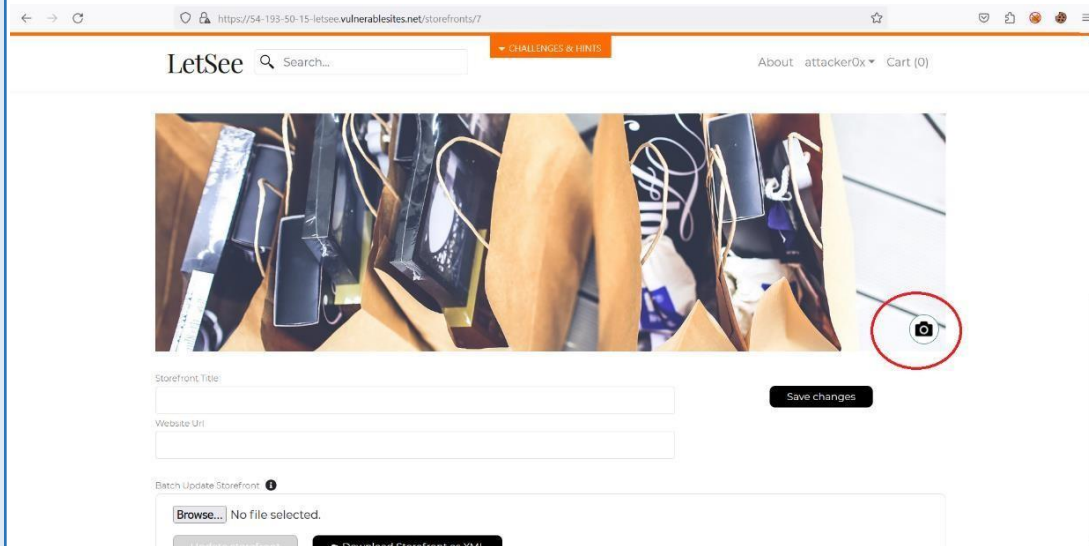
Step1. Go to your account.

Step2. Go to Storefront and click on Manage shop.

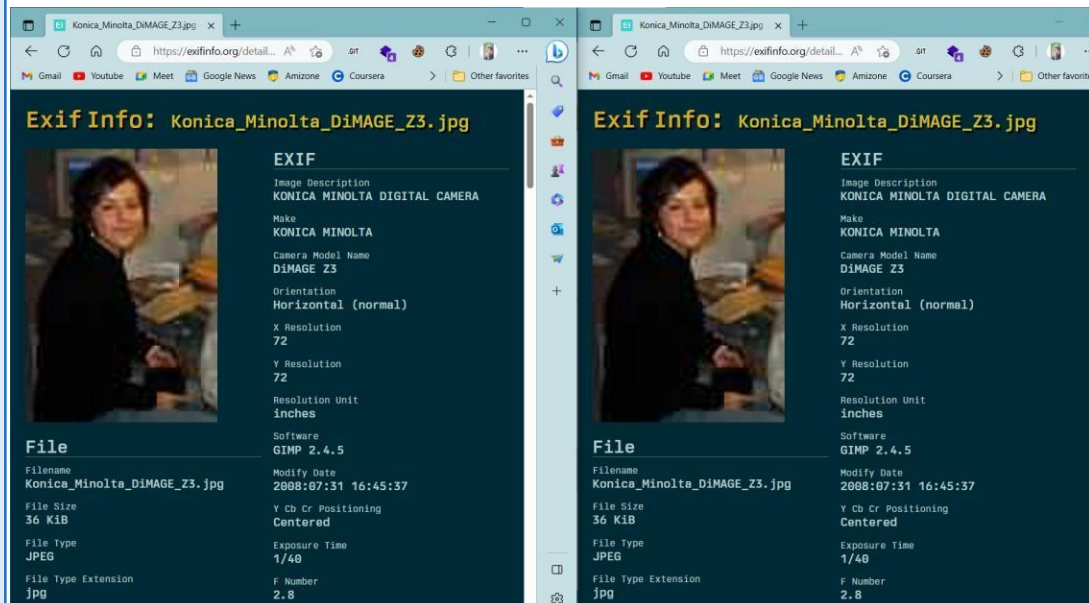
Proof of Concept

The screenshot shows a web browser window with the URL `https://54-193-50-15-letsee.vulnerablesites.net/account`. The page header includes the LetSee logo, a search bar, a navigation menu with 'CHALLENGES & HINTS', and user information 'About attacker0x' and 'Cart (0)'. The main content area is titled 'Account Settings' and contains three sections: 'About you', 'Your shipping addresses', and 'Credit Cards'. The 'About you' section has input fields for 'Name' (Bob-Attacker), 'Username' (attacker0x), and 'Website' (https://www.mysite.com/), along with an 'Update User' button. To the right of these fields is a profile picture placeholder with a 'My Storefront' button below it, which is circled in red. The 'Your shipping addresses' section has an 'Add a new address' button, and the 'Credit Cards' section has an 'Add new card' button.

Step3. Click on camera button and upload the file which may contains meta data. Reference: [GitHub - ianare/exif-samples: Sample images for testing Exif metadata retrieval.](https://github.com/ianare/exif-samples)



Step4. After uploading, come back the to the accounts section & right click on image > Copy link.



Step5. Visit, [Exif Info: view meta-data in your files](https://exifinfo.org/) to check the exif data.

Vulnerability	6.4 Session invalidation
Severity Level	Moderate
Affected IP/URL	https://54-193-50-15-letsee.vulnerablesites.net/account
OWASP Category	Insufficient/Broken Access Control
Description	The website fails to properly invalidate user sessions upon logout, allowing an active session to remain valid and accessible even after the user explicitly logs out. This vulnerability poses significant risks as an attacker could potentially hijack the active session and impersonate the logged-out user, gaining unauthorized access to their account and sensitive data.
Recommendation	It is essential to implement proper session management mechanisms. This includes ensuring that the session is invalidated, and all associated session tokens and cookies are properly cleared upon logout. Additionally, implementing session timeout measures, such as setting an appropriate idle session expiration time, can further enhance security by automatically terminating inactive sessions.
References	1. A01 Broken Access Control - OWASP Top 10:2021 2. Session Management - OWASP Cheat Sheet Series

Step1. Go to your account.

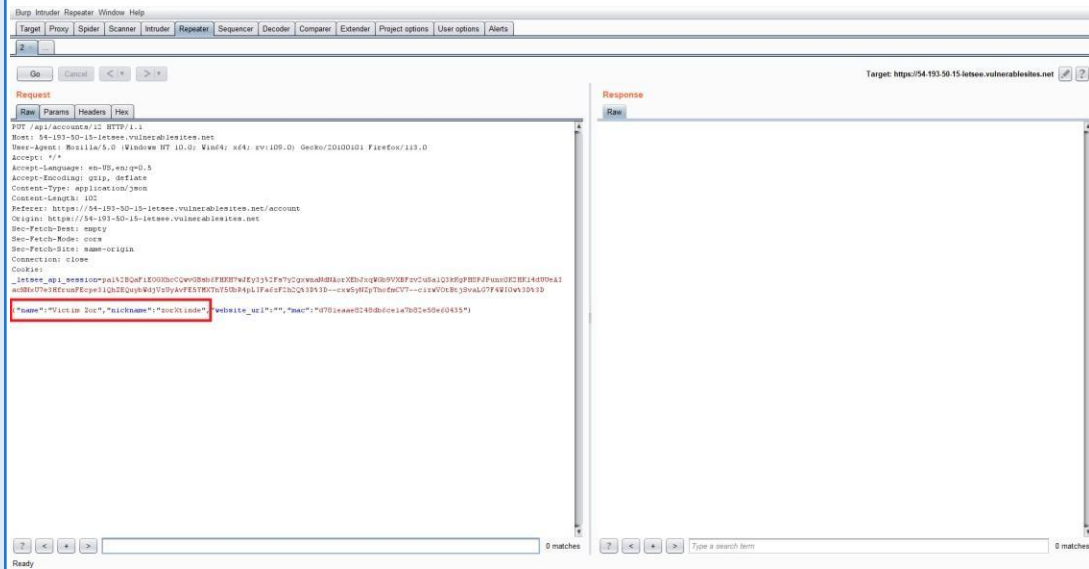
Step2. Update the name, nickname, etc.

Proof of Concept

The screenshot shows a web browser window with the URL `https://54-193-50-15-letsee.vulnerabilities.net/account`. The page title is "LetSee" and it features a search bar and a "CHALLENGES & HINTS" button. The main content area is titled "Account Settings" and is divided into three sections:

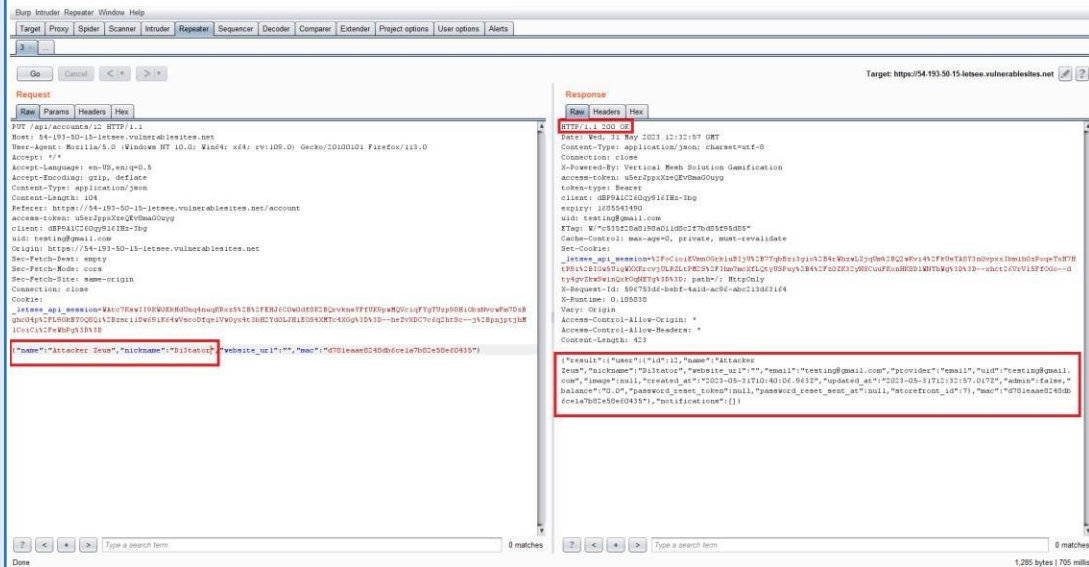
- About you:** Contains three input fields with the following values: "Victim Zor", "zorXtinde", and "Website: https://www.mysite.com/". Below these fields are two buttons: "Update User" and "My Storefront".
- Your shipping addresses:** Includes a button labeled "Add a new address".
- Credit Cards:** Includes a button labeled "Add new card".

Step3. Open Burpsuite and Capture the request while updating your account information.



Step4. Now, Logout from your account.

Step5. Change the name, etc. and forward the request. Boom! You'll get 200 OK response from the web server.



Step6. Login into your account and observe, Account details has been changed without proper account access.

Vulnerability	6.5 No Rate Limiting leads to Account Take Over
Severity Level	High
Affected IP/URL	https://54-193-50-15-letsee.vulnerablesites.net/login
OWASP Category	Throttling and Rate Limiting
Description	The website lacks rate limiting on its login page, which leads to a significant security vulnerability that can result in account takeover attacks. Without rate limiting, an attacker can repeatedly attempt to guess user credentials without any restrictions, exploiting weak passwords or using brute-force techniques to gain unauthorized access to user accounts. This vulnerability allows malicious actors to automate login attempts, making it easier for them to bypass authentication mechanisms and gain control over user accounts.
Recommendation	Implement Strong Password Policy, CAPCHA, 2FA and account lockout to avoid rate limiting vulnerability.
References	OWASP LA Robert Lee Combating Account Takeover 2017 11.pdf

Step1. Visit the above URL.

Step2. Enter the email address of the victim & enter wrong password.

Step3. Now, capture the request in Burpsuite & send request into Repeater tab.

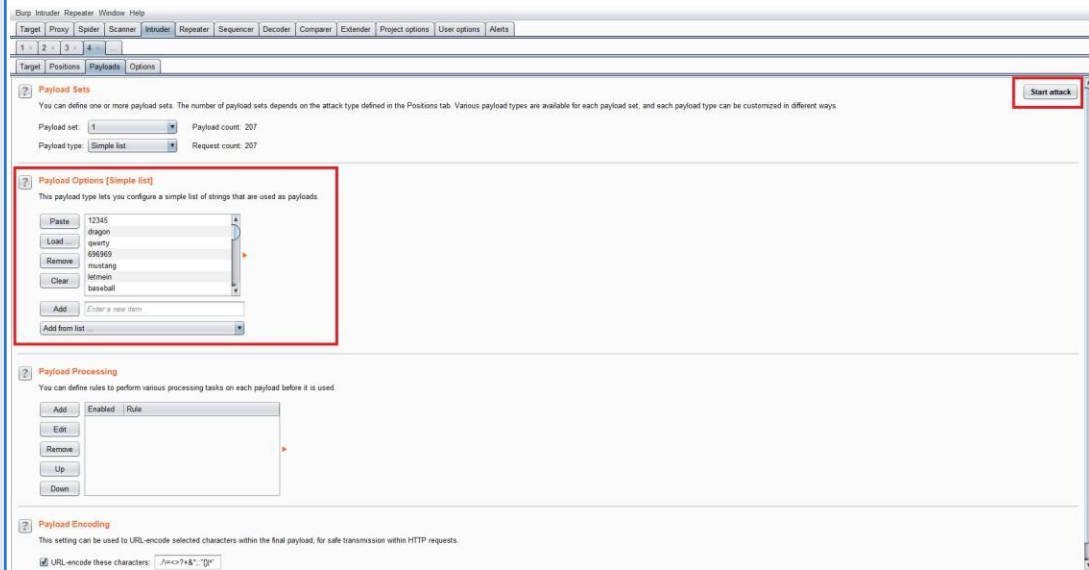
Proof of Concept

The screenshot shows the Burp Suite Repeater interface. The 'Payload Positions' tab is active, displaying a captured HTTP request. The request is a POST to the endpoint `/api/auth/sign_in HTTP/1.1`. The request headers include `Host: 14-193-50-15-letssee.vulnweb.in:8080`, `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/113.0`, `Accept: */*`, `Accept-Language: en-US,en;q=0.5`, `Accept-Encoding: gzip, deflate`, `Content-Type: application/json`, and `Content-Length: 54`. The request body is a JSON object: `{\"email\": \"hellotokarep10@gmail.com\", \"password\": \"123\"}`. The interface also shows a search bar at the bottom with '0 matches' and 'Length: 752'.

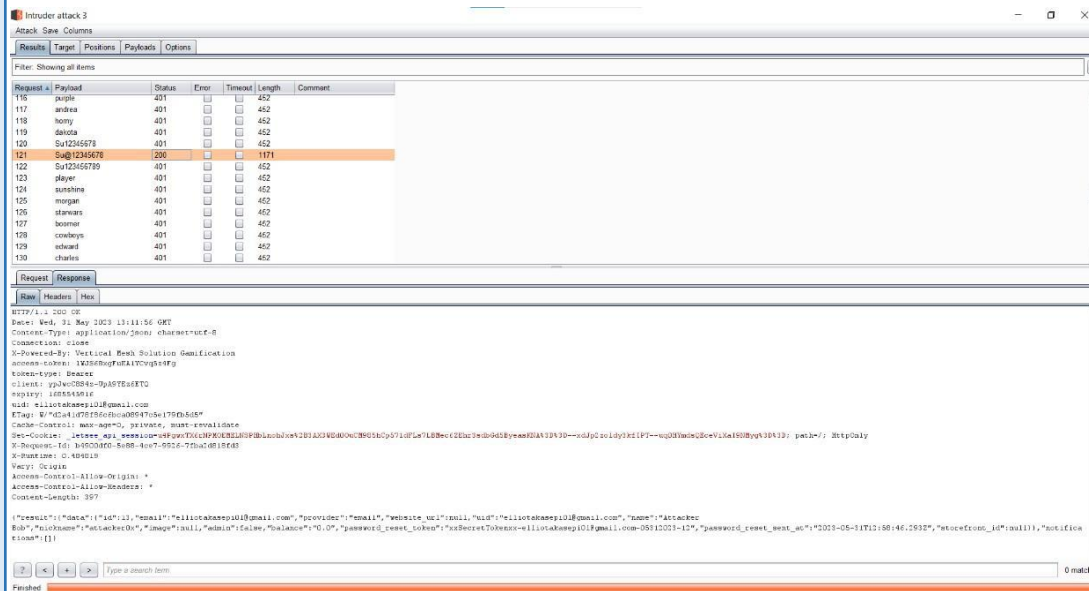
Step4. Select the "password" parameter, add as brute force target.

Step5. Now create a wordlist or for reference use my wordlist with the testing password in the list. [s3ctar0r/test-wordlist \(github.com\)](https://github.com/s3ctar0r/test-wordlist)

Step6. Set Payload type as Simple, paste the wordlist and click on Start Attack button.



Step7. After completion of the attack check for 200 Status code and bigger length of the response.



Step8. Now go and confirm the password by logging into the web application.

Vulnerability	6.6 Parameter tampering / Business Logic
Severity Level	Critical
Affected IP/URL	https://54-193-50-15-letsee.vulnerablesites.net/products/5 (Choose any product you want)
OWASP Category	Price manipulation
Description	<p>The website is vulnerable to parameter tampering, specifically product price manipulation. Parameter tampering refers to the unauthorized modification of data parameters passed between a client and a server in a web application. In this case, the vulnerability allows attackers to manipulate product prices by altering the parameters associated with the price calculation or display. Attackers exploit this vulnerability by modifying the parameters sent to the server, which controls the calculation or display of product prices. By manipulating these parameters, attackers can fraudulently lower or increase the prices of products, potentially leading to financial loss for the website or unfair pricing for customers.</p> <p>This affects business so it is also known to be Business Logic vulnerability.</p>
Recommendation	<p>Implement strict access controls to ensure that only authorized personnel can modify or update pricing-related parameters. Perform server-side validation of all input parameters related to pricing calculations or display. Validate and sanitize user input to ensure that only authorized and expected values are accepted.</p>
References	Web Parameter Tampering OWASP Foundation

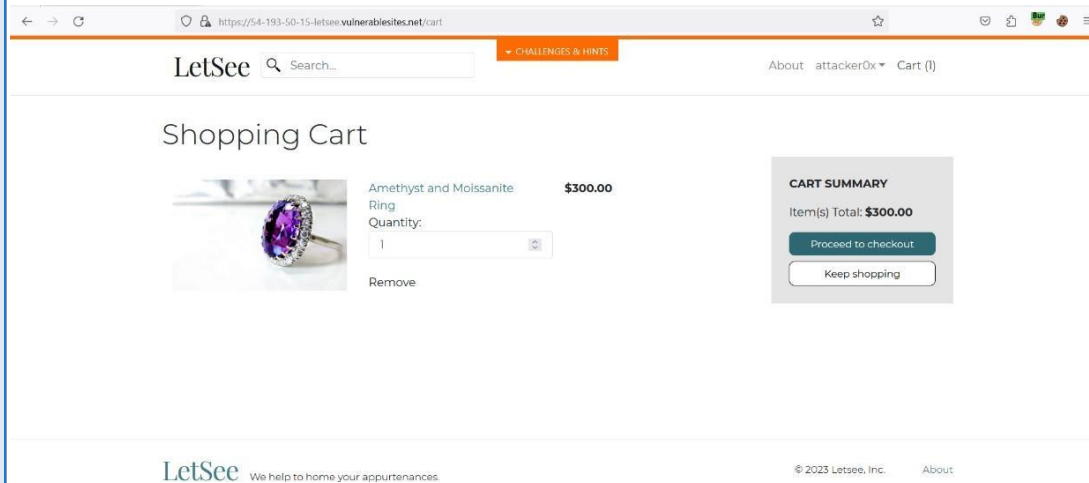
Step1. Go to the website. Click on any product you wish.

Step2. Add the product into Cart.

Proof of Concept

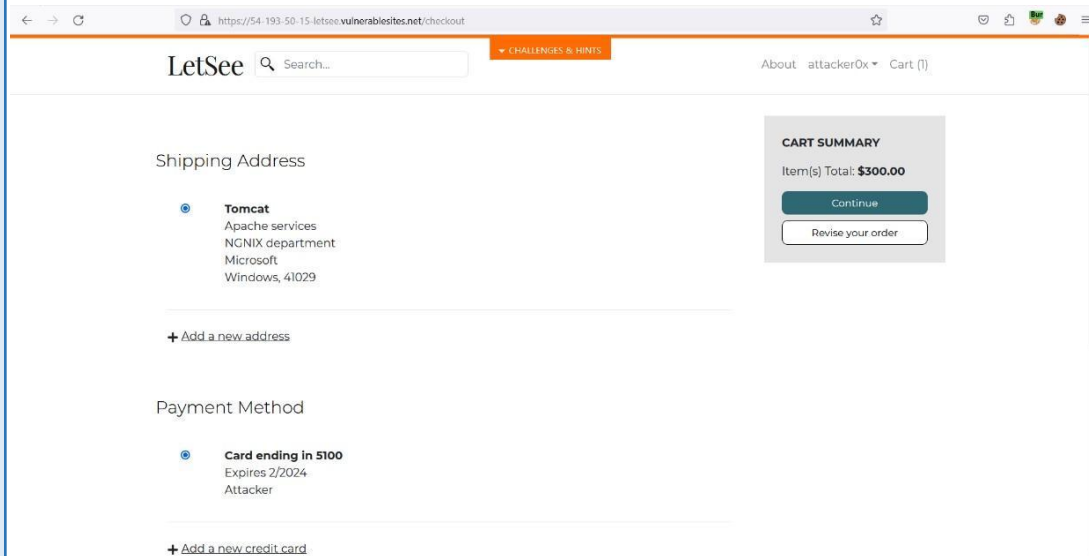
The screenshot shows a web browser window with the address bar displaying `https://54-193-50-15-letsee.vulnerablesites.net/products/5`. The website header includes the 'LetSee' logo, a search bar, a navigation menu with 'CHALLENGES & HINTS', and a user profile 'attacker0x' with a cart containing 0 items. The main content area features the heading 'Incredible Fashion and Jewelry' with a sub-heading 'SHOP BY EDNA MODE'. A large image of a ring with a purple amethyst centerstone and a diamond halo is shown. To the right of the image, the product is titled 'Amethyst and Moissanite Ring' with a price of '\$300.00' and a note '3 in stock'. A quantity selector is set to '1', and an 'Add To Cart' button is visible. Below the image, a 'Description' section contains the text: 'Don't miss this opportunity to own this gorgeous gemstone ring crafted in silver with a large amethyst centerstone. The ring is accented with Moissanite, an alternative to Diamonds that is harder and more lustrous than cubic zirconia.'

Step3. Click on "Proceed to checkout"



The screenshot shows the LetSee shopping cart page. The browser address bar displays <https://54-193-50-15-letsee.vulnerablesites.net/cart>. The page header includes the LetSee logo, a search bar, a "CHALLENGES & HINTS" button, and navigation links for "About", "attacker0x", and "Cart (1)". The main content area is titled "Shopping Cart" and features a product listing for an "Amethyst and Moissanite Ring" priced at \$300.00. The quantity is set to 1. A "Remove" link is provided below the product. To the right, a "CART SUMMARY" box shows the total item price as \$300.00 and contains two buttons: "Proceed to checkout" and "Keep shopping". The footer includes the LetSee logo with the tagline "We help to home your appurtenances", the copyright notice "© 2023 Letsee, Inc.", and an "About" link.

Step4. Click Continue.



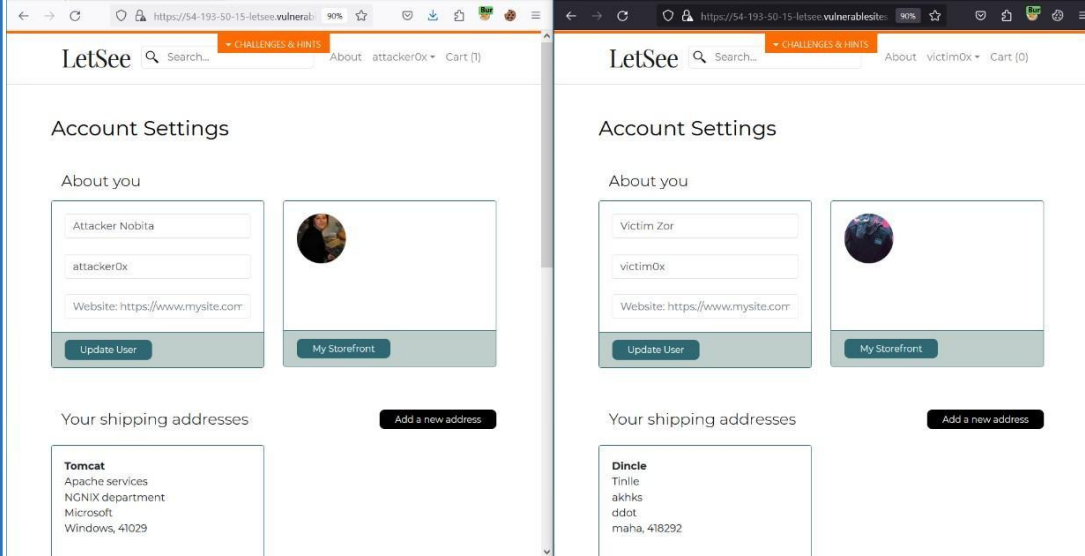
The screenshot shows the LetSee checkout page. The browser address bar displays <https://54-193-50-15-letsee.vulnerablesites.net/checkout>. The page header is identical to the shopping cart page. The main content area is titled "Shipping Address" and shows a selected address: "Tomcat", "Apache services", "NGNIX department", "Microsoft", "Windows, 41029". Below this is a link to "+ Add a new address". The "Payment Method" section shows a selected card: "Card ending in 5100", "Expires 2/2024", "Attacker". Below this is a link to "+ Add a new credit card". On the right, a "CART SUMMARY" box shows the total as \$300.00 and contains two buttons: "Continue" and "Revise your order".

Vulnerability	6.7 Direct Object Access (IDOR) / Business Logic
Severity Level	Critical
Affected IP/URL	https://54-193-50-15-letsee.vulnerablesites.net/account
OWASP Category	Insecure Direct Object Reference
Description	The website is susceptible to an Insecure Direct Object Reference (IDOR) vulnerability in the account details change functionality. IDOR occurs when an attacker can directly manipulate or access internal object references, such as user accounts or sensitive data, by tampering with input parameters or request payloads.
Recommendation	Implement Proper Authorization and Access Controls: Ensure that appropriate authorization checks are in place to restrict access to sensitive resources. Users should only be able to access and modify their own resources, preventing unauthorized access to other users' data. Validate User Permissions: Implement server-side validation to verify that the authenticated user has the necessary permissions to access or modify specific resources. This prevents unauthorized actions on sensitive objects.
References	Insecure Direct Object Reference Prevention - OWASP Cheat Sheet Series

Step1. Create multiple accounts, login into the accounts & go to profile

Account. Step2. While making a purchase you can capture “user_id” and “mac”
id and save it.

Proof of Concept



Step3. We have to change userid on the top with HTTP PUT request as well as mac parameter.

The screenshot shows a web browser displaying the 'Account Settings' page for a user named 'Attacker Nobita'. The page includes fields for 'Attacker Nobita', 'attacker0x', and 'Website: https://www.mysite.com'. A network tool on the right shows an intercepted HTTP PUT request to 'https://54-193-50-15-letsee.vulnerablelabs.net/443'. The request body contains a JSON object with 'name' and 'mac' fields. The 'mac' field value is highlighted in red: `"mac": "7f0c1e9a5c40b1e1e702e450e60435"`.

Step4. Here, we can find the user_ids and mac ids for both the accounts.

The screenshot shows the 'Account Settings' page for a user named 'Victim Zor'. The page includes fields for 'Victim Zor', 'victim0x', and 'Website: https://www.mysite.com'. A network tool on the right shows an intercepted HTTP PUT request to 'https://54-193-50-15-letsee.vulnerablelabs.net/443'. The request body contains a JSON object with 'name' and 'mac' fields. The 'mac' field value is highlighted in red: `"mac": "d781eaae8240df6ce1a7b82e59e60435"`. A Notepad window is open in the foreground, displaying the following text:

```
attacker0x - "user_id":13  
"mac": "d09079d794a6ee60d836f884739f7196"  
  
victim0x - "user_id":12  
"mac": "d781eaae8240df6ce1a7b82e59e60435"
```

Step5. Now, replace the attacker's ids with victim ids.

The image shows two browser windows. The top window displays the 'Account Settings' page for a user named 'Attacker Nobita' with the username 'attacker0x'. The bottom window shows the same page but with the username changed to 'Victim Zor' and the profile picture updated. To the right, a proxy tool (Burp Suite) is shown intercepting an HTTP request. The request body contains a JSON object with fields for 'username' and 'password'. The 'username' field is highlighted in red and contains the value 'attacker0x'. The 'password' field is also highlighted in red and contains a long alphanumeric string.

Step6. Here, you can see the victim details have been updated successfully!

The image shows two side-by-side screenshots of the 'LetSee' website's 'Account Settings' page. The left screenshot shows the account details for 'Attacker Nobita' with the username 'attacker0x'. The right screenshot shows the account details for 'Victim Zor' with the username 'victim0x'. Both screenshots have a red box highlighting the 'About you' section, which includes the name, username, profile picture, and website. Below the 'About you' section, there are buttons for 'Update User' and 'My Storefront'. The 'Your shipping addresses' section is also visible, showing a list of addresses for 'Tomcat' and 'Dincle'.

Vulnerability	6.8 SQL injection
Severity Level	High
Affected IP/URL	https://54-193-50-15-letsee.vulnerablesites.net/search?search=
OWASP Category	Injection
Description	SQL Injection is a web application vulnerability that allows attackers to manipulate the underlying SQL queries executed by the website's database. In this case, the vulnerability exists specifically within the search functionality, enabling attackers to inject malicious SQL code into the search parameter. Exploiting this vulnerability, attackers can manipulate the SQL queries to perform unauthorized actions, gain unauthorized access to sensitive data, modify or delete data, or even execute arbitrary commands on the database server.
Recommendation	Implement strict input validation and sanitization techniques to ensure that user-supplied input is properly validated and does not contain any malicious SQL code. This includes using parameterized queries or prepared statements to separate data from the SQL query structure. Use of Web Application Firewall that includes SQL Injection detection and prevention capabilities. This can help identify and block malicious SQL Injection attempts before they reach the database.
References	<ol style="list-style-type: none">1. SQL Injection OWASP Foundation2. SQL injection cheat sheet Web Security Academy (portswigger.net)

Step1. Go to the search box.

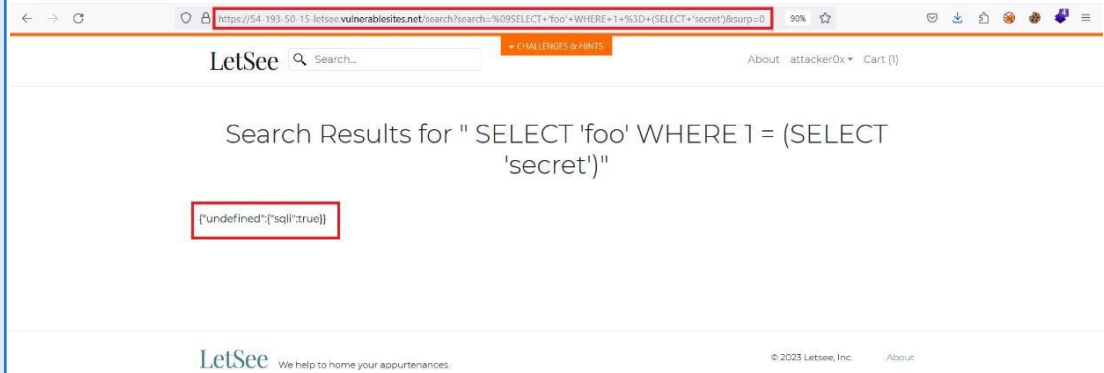


Proof of Concept



Step2. Enter the SQLi payload:

```
SELECT 'foo' WHERE 1 = (SELECT 'secret')
```



Here, you got the SQL injection vulnerability used for extracting data via visible errors.

I hereby declare that all the findings presented in this document are the result of my diligent research and analysis, representing my authentic and legitimate discoveries.

VulnDetox

re

Validated and Verified by

