

**Report CTF**

**CTF Tornado Penetration Testing Report**

Date: 30/08/2024

Version: 1.0

**1 Table Of Contents**

1. Document Revision History………………………………………………………….........………………………...........…2
2. Executive Summary....................................................................................................................................3
3. Scope..........................................................................................................................................................4
4. Risk Categories...........................................................................................................................................5
5. Pentest Methodology..................................................................................................................................6
6. Visual Summary..........................................................................................................................................7
7. Findings Summary......................................................................................................................................8

**1 Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Version** | **Contact Details** |
| Abhishek Joshi | 30/08/2024 | 0.1 | Email: [abhishekjoshi266@gmail.com](mailto:abhishekjoshi266@gmail.com)  Mob No: +91 -- |

# u e

**2 Executive Summary**

An analysis of a black box penetration test conducted on the Tornado “CTF Machine” is presented in this document. Based on a thorough security assessment performed by Internal Security Team in August of 2024.

This assessment was conducted On-Premises by the Security team. An assessment was conducted on the 30th of August to 30th of August 2024. As a comprehensive strategy for this assessment, Security Team concreted the black box penetration testing methodology and technique. To facilitate this, Company provided a walkthrough of the application and provided access to the test environment with valid different privilege accounts.

Testing was carried out by identifying vulnerabilities with the intent of accessing critical information. The objective of performing this activity was to assess the security risks associated with the developed applications and identify vulnerabilities that cybercriminals could leverage to compromise the application. The report summarizes the security findings related to the Company applications and network.

**This assessment aimed to:**

Analyze the application for technical vulnerabilities that an attacker may exploit to compromise the CTF Machine.

Provide recommendations for risk mitigation that may arise on successful exploitation of these vulnerabilities.

**3 Scope**

### Scope

The section defines the scope and boundaries of the project.

### Constraints and Limitations

The assessments, and the result(s) / finding(s) made are highly subjective to target system(s) and service(s) visibility and availability at that given point of time.

### Target Scope

Identify weaknesses that might be exploited by adversaries who have authorized or unauthorized access to Company Technical Skill Test and underlying infrastructure:

Test Perform On Tornado CTF Environment Without Credential as Black Box Testing.

Following Machine was in the scope of the penetration test.

### Machine and Environment Details

|  |  |  |
| --- | --- | --- |
| Sr . No CTF Name Url: | | |
| 1 | Hacksudo Thor | Machine Url: <https://www.vulnhub.com/entry/ia-tornado,639/>  Machine IP ( 192.168.56.131 ) |

### Contact Details

|  |  |
| --- | --- |
| Names Contact Details | |
| Abhishek Joshi (Penetration Tester) | Skype ID: live:.cid.afc2ad6f8029ae21  Mail: [a](mailto:Abhishek.joshi@fendahl.com)bhishekjoshi266@gmail.com |

### 

**4 Risk Categories**

### Risk Categories & Rationales

Pentest use a simple risk categorisation of each vulnerability to focus the triage process at the risks which truly matter. The Common Vulnerability Scoring System (CVSS) is an industry standard formula. It generates a risk score between 0.0 and 10.0.

The table below explains the risk categories and demonstrates rule-of-thumb equivalency with CVSS scores:

|  |  |  |
| --- | --- | --- |
| **Risk Category** | **CVSS Score** | **Rationales** |
| Critical | 8.1 – 10.0 | Poses a severe risk which is easy to exploit. Begin the process of remediating immediately after the issue has been presented. |
| High | 6.1 – 8.0 | Poses a significant risk and can be exploited. Address these as soon as possible after any critical risks have been remediated. |
| Medium | 4.1 – 6.0 | Poses an important risk but may be difficult to exploit. Pentest recommends remedial work within 3 months of discovery. |
| Low | 2.1 – 4.0 | Poses a minor risk or may be exceedingly difficult to exploit. Address these over the long-term during testing cycles |
| Informational | 0.0 – 2.0 | Loss of sensitive information, or a discussion point. These are not directly exploitable but may aid an attacker. Remediate these to create a true defence-in-depth security posture, |

CVSS is not applicable to all risks. For example, it is incapable of capturing the risk of a “flat network design”. Experience has told us that this is a “high” risk in most cases.

For this reason, the reader may find vulnerabilities which have no CVSS rating in our reports.

We endeavour to provide the reason for omitting the risk score when that is the case, and to provide CVSS by default in all applicable cases.

**5 Pentest Methodology**

### Methodology

The penetration testing methodology is typically based on the NIST security methodology. The focus shifts from traditional application security, where the primary threat is from multiple sources over the Internet. The key difference is in the client-side security, file system, hardware, and network security. Traditionally for Thick Client Applications, an end user is in control of the device. Security Team used the NIST & MITRE Attack Framework testing guide for conducting penetration test of the systems and applications. The testing was done to simulate as closely as possible the viewpoint of completely external attacker, the steps involved are

1] Setup

2] Discovery

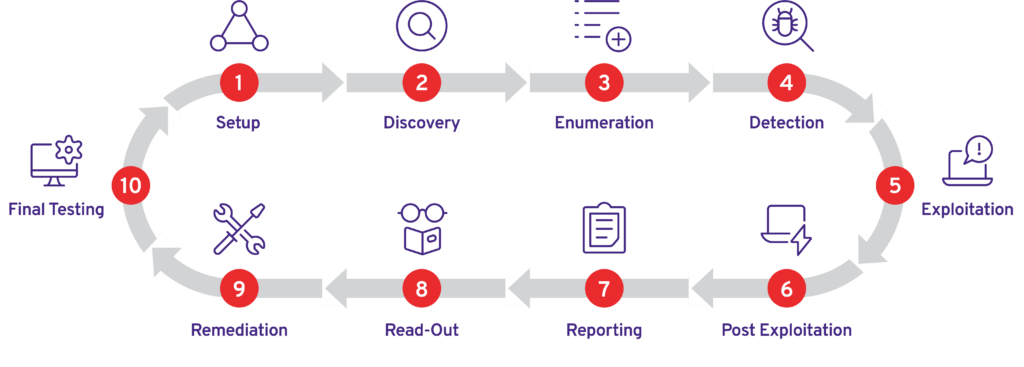
3] Enumeration

4] Detection

5] Exploitation

6] Post-Exploitation

7] Reporting



### Visual Summary

Graphical representation of Identified Vulnerabilities to Severity Risk rating

|  |  |  |
| --- | --- | --- |
| Sr. No. Severity Level Frequency | | |
| 1 | Critical | 1 |
| 2 | High | 1 |
| 3 | Medium | 1 |
| 4 | Low | 0 |

Table: Representing Severity Level

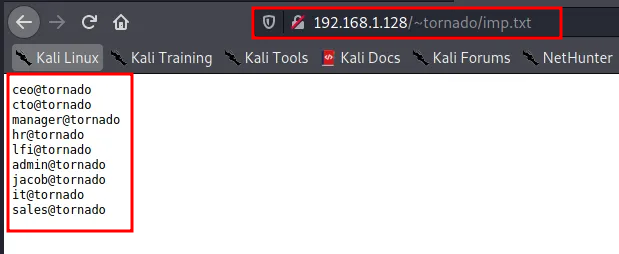
### Findings Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Observed Vulnerability** | **Risk Rating** | **Status** | **Comments** |
| 1. | Account Takeover | Critical | Not Fixed | -- |
| 2. | Reflected XSS | High | Not Fixed | -- |
| 3. | Email Enumeration | Medium | Not Fixed | -- |

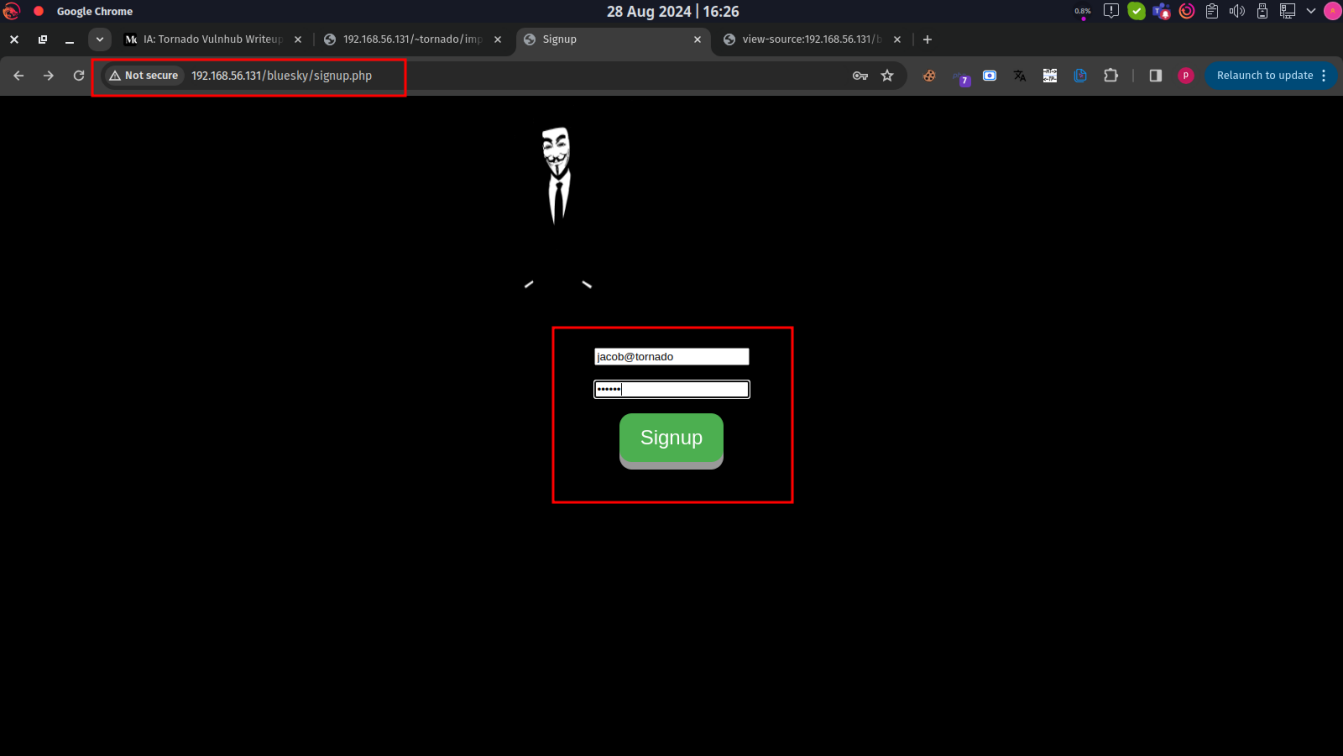
#### Account Takeover

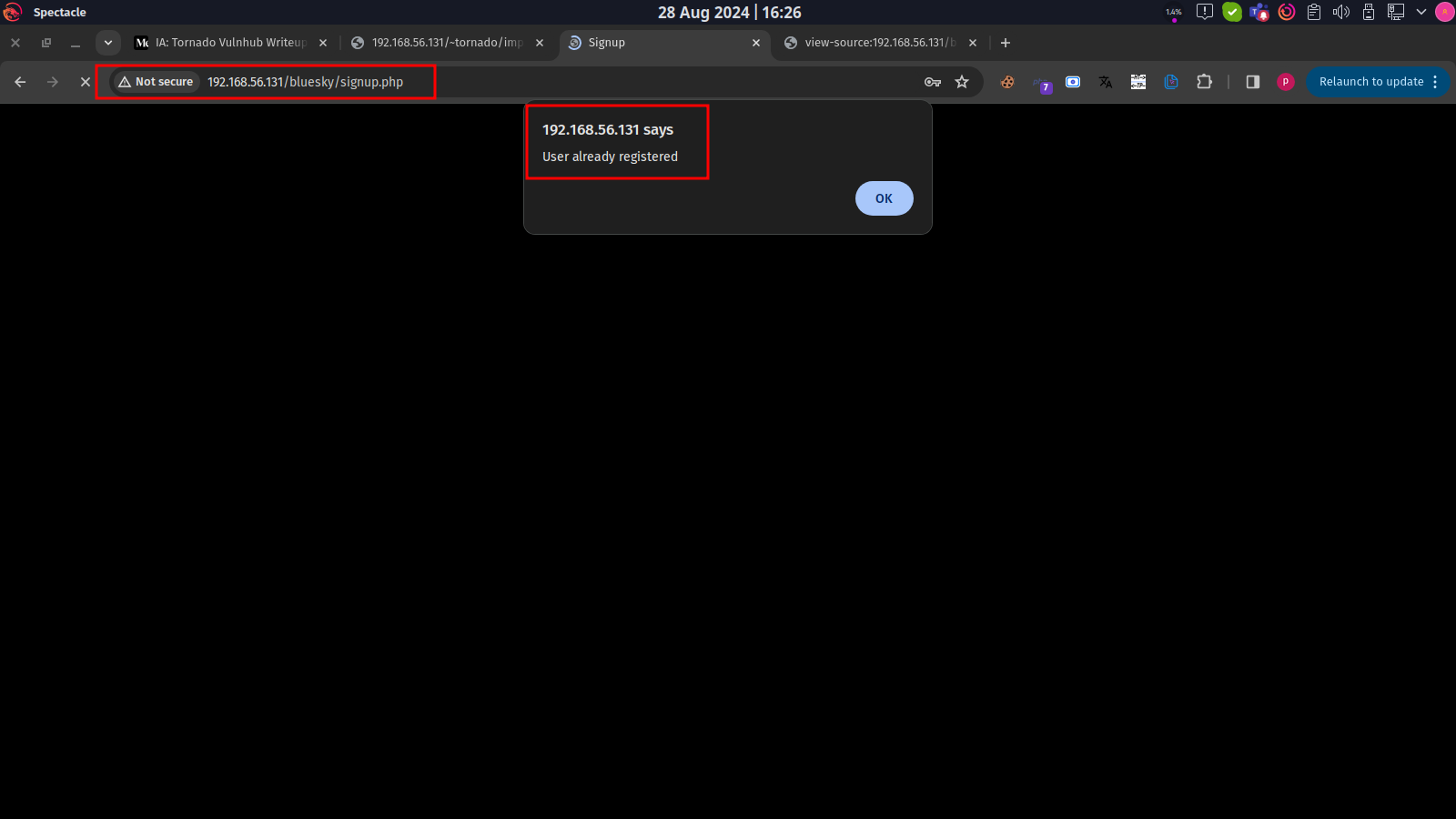
|  |  |
| --- | --- |
| **Vulnerability** | Account Takeover |
| **Description** | Account Takeover lead to unauthorized access to user data, including personal information, financial data, or other sensitive information. If the application allows password resets via email, an attacker could also gain full control of the account. |
| **Risk/Impact** | An attacker could potentially compromise existing accounts by registering new accounts with the same email address. Attacker use inspect element to increase lenght of email form and register successfully with existing email . Now attacker can access their account. |
| **CVSS Score** | 9.5 Critical |
| **Path:** | 192.168.56.131 |
| **Remidiation / Solution** | Remidiation:  - To mitigate this vulnerability, the application should implement proper input validation and sanitization. This includes checking the length of input fields and ensuring that they match expected patterns (e.g., a valid email address format).  - The application should also implement server-side validation. This means that even if an attacker bypasses client-side validation (such as by using inspect element), the server will still reject the input if it is invalid. |
| **Refrence Url:** | <https://nvd.nist.gov/vuln/detail/CVE-2024-22179> |

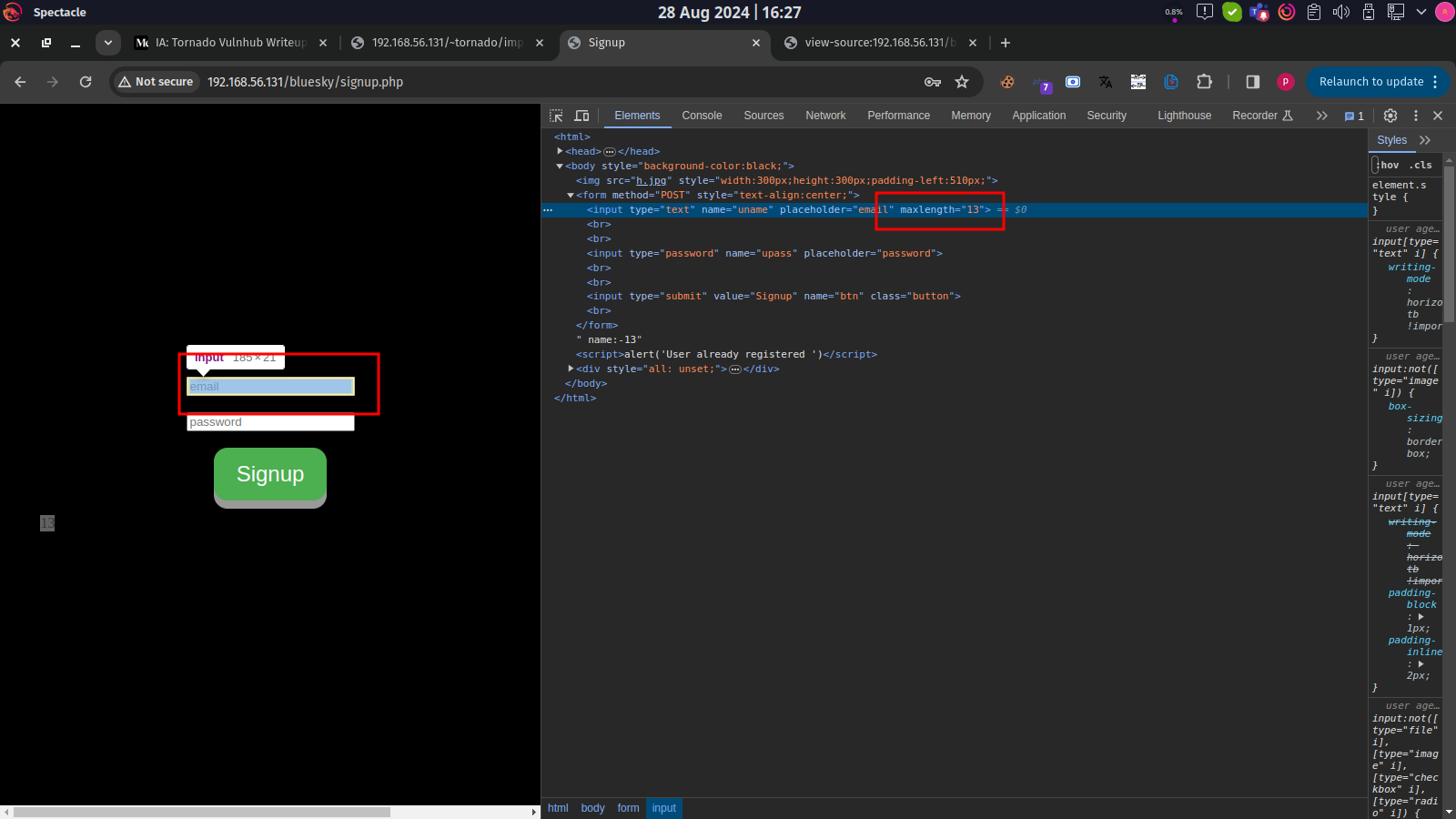
Below Screenshots shows that attacker found email address.

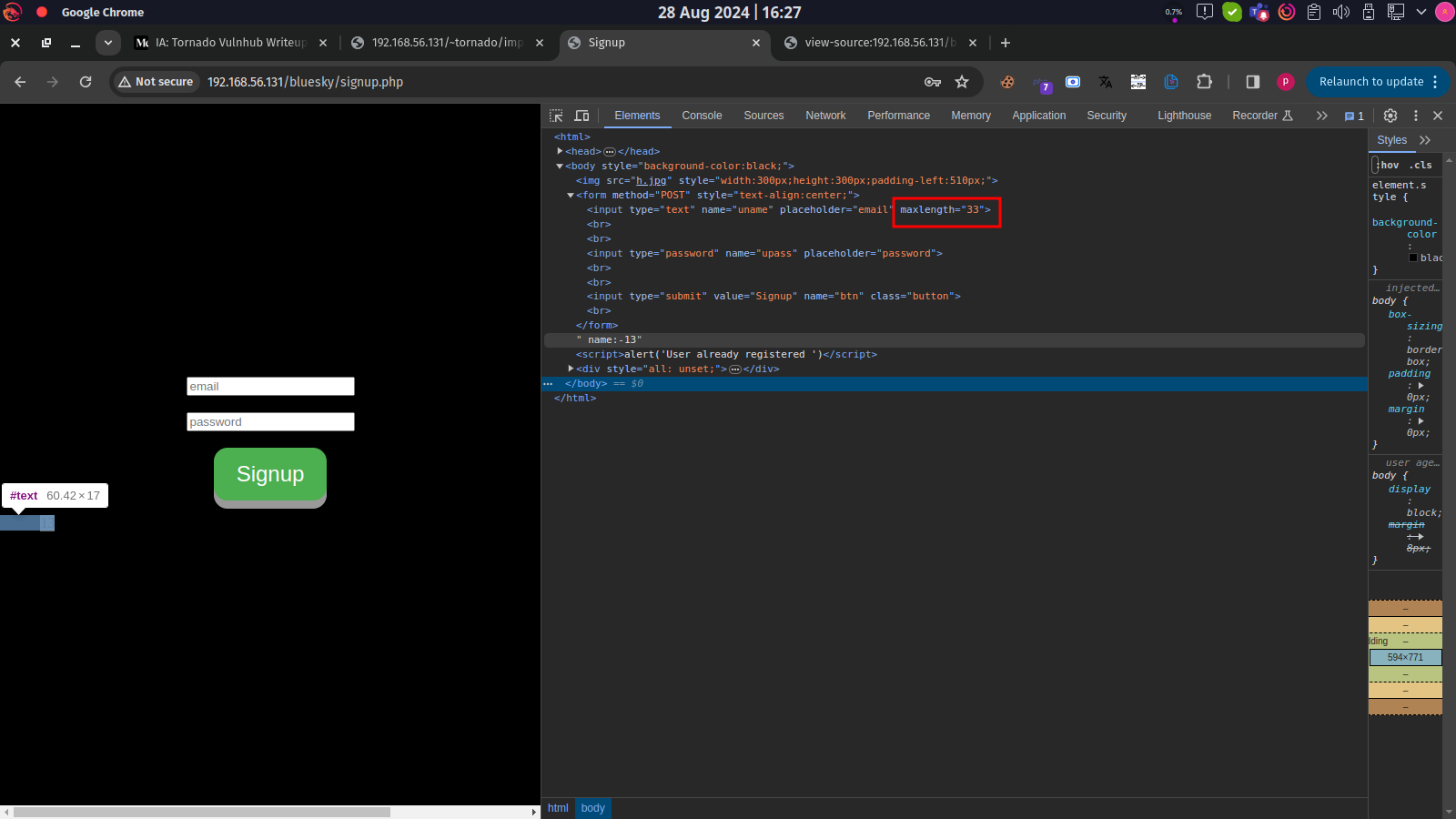


Below Screenshots shows that attacker found all email addressess is alredy register then attacker use inspect element and check length of email address and increase lenght for performing Account Takeover.

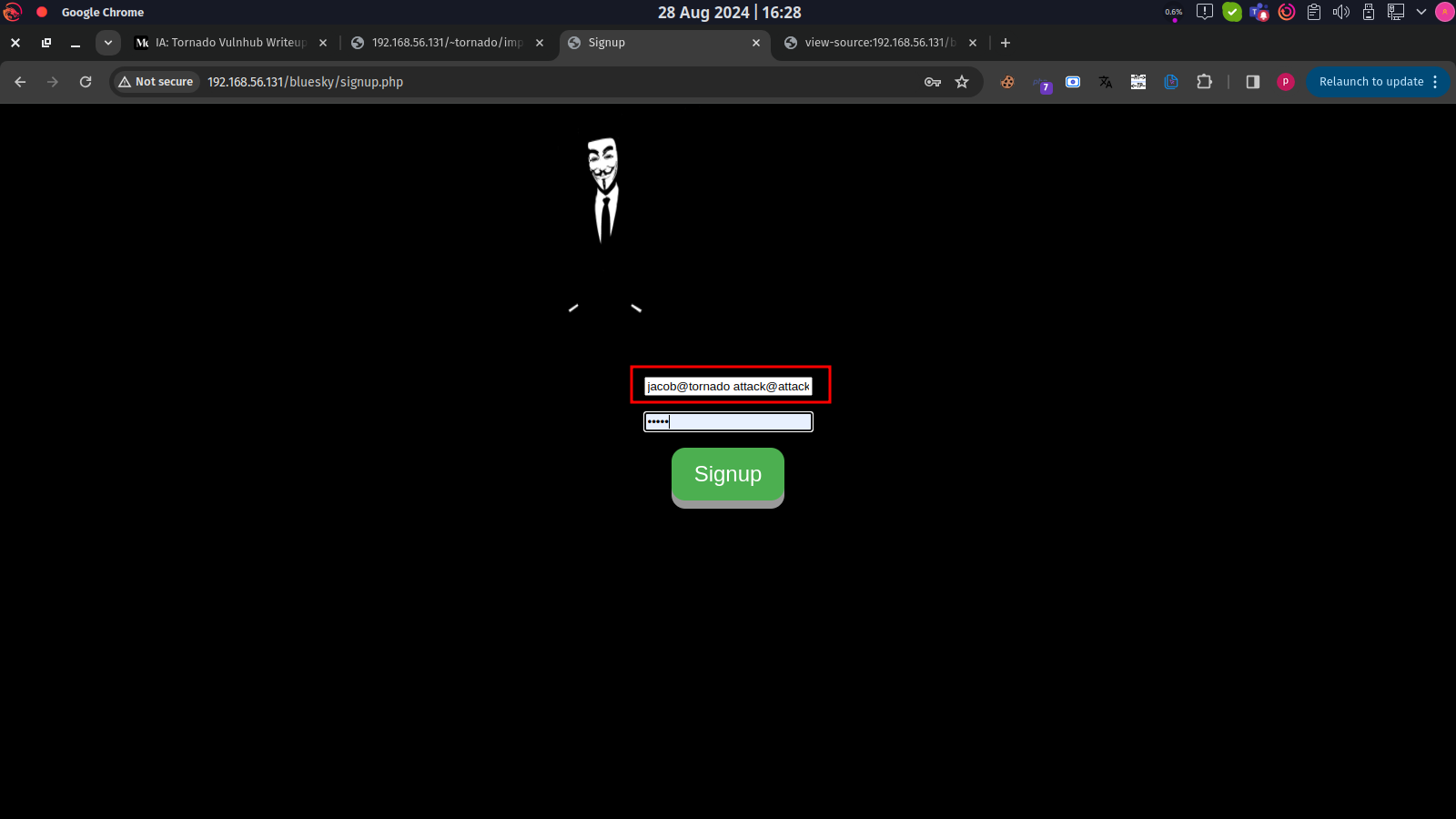


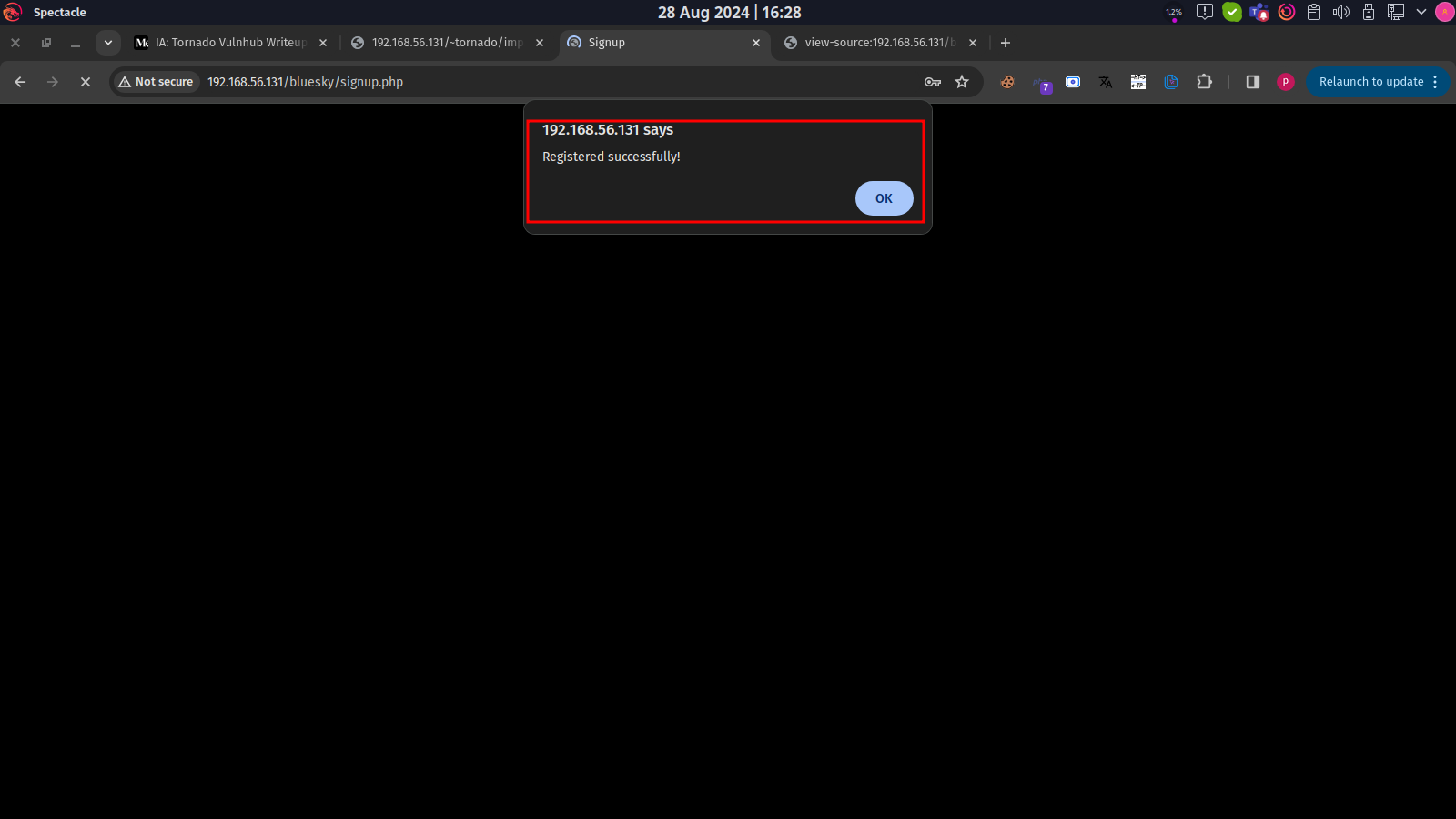


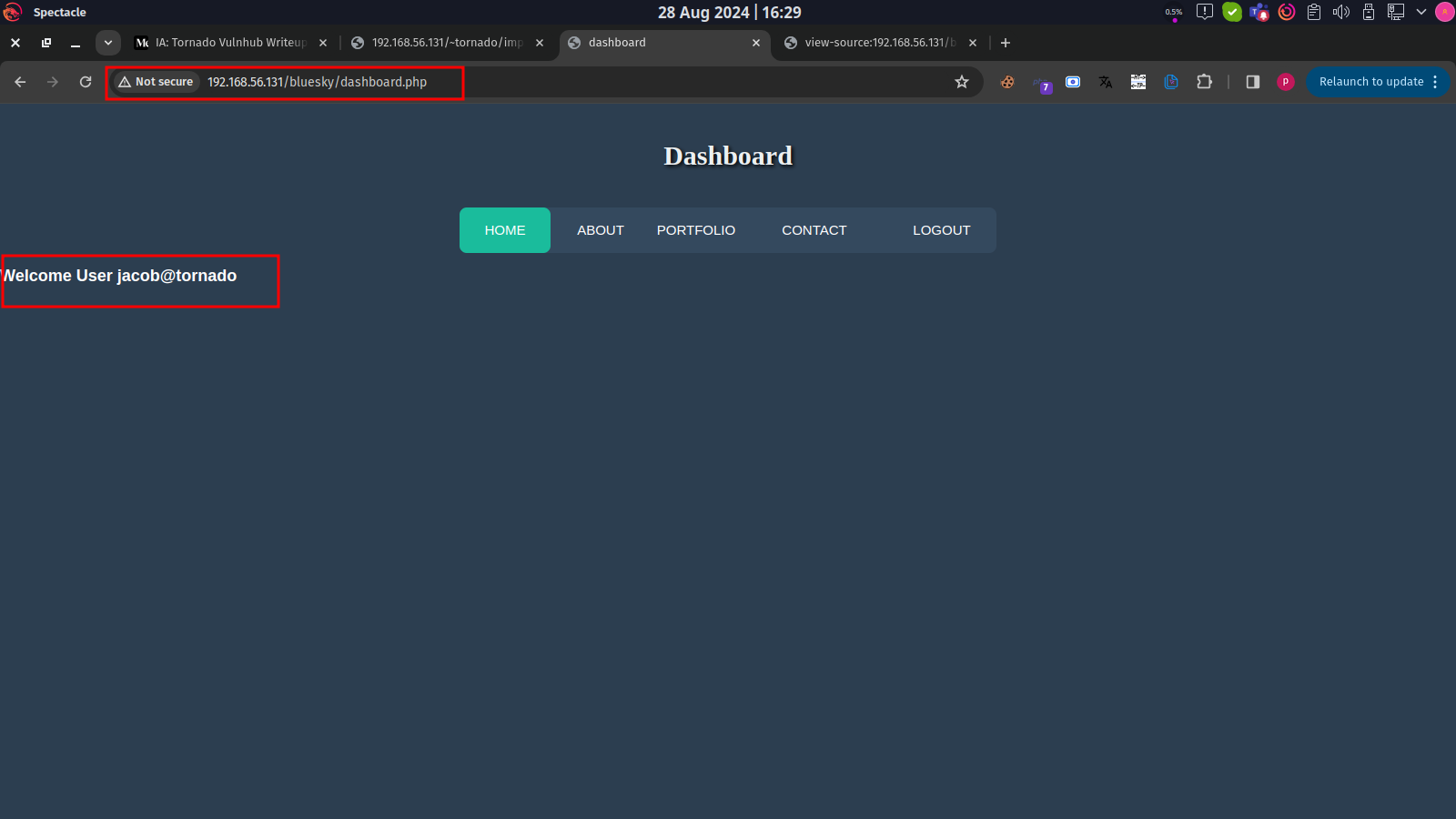
= 



Below Screenshots shows that attacker manuplate registration and register again with alredy register account. And gaining access



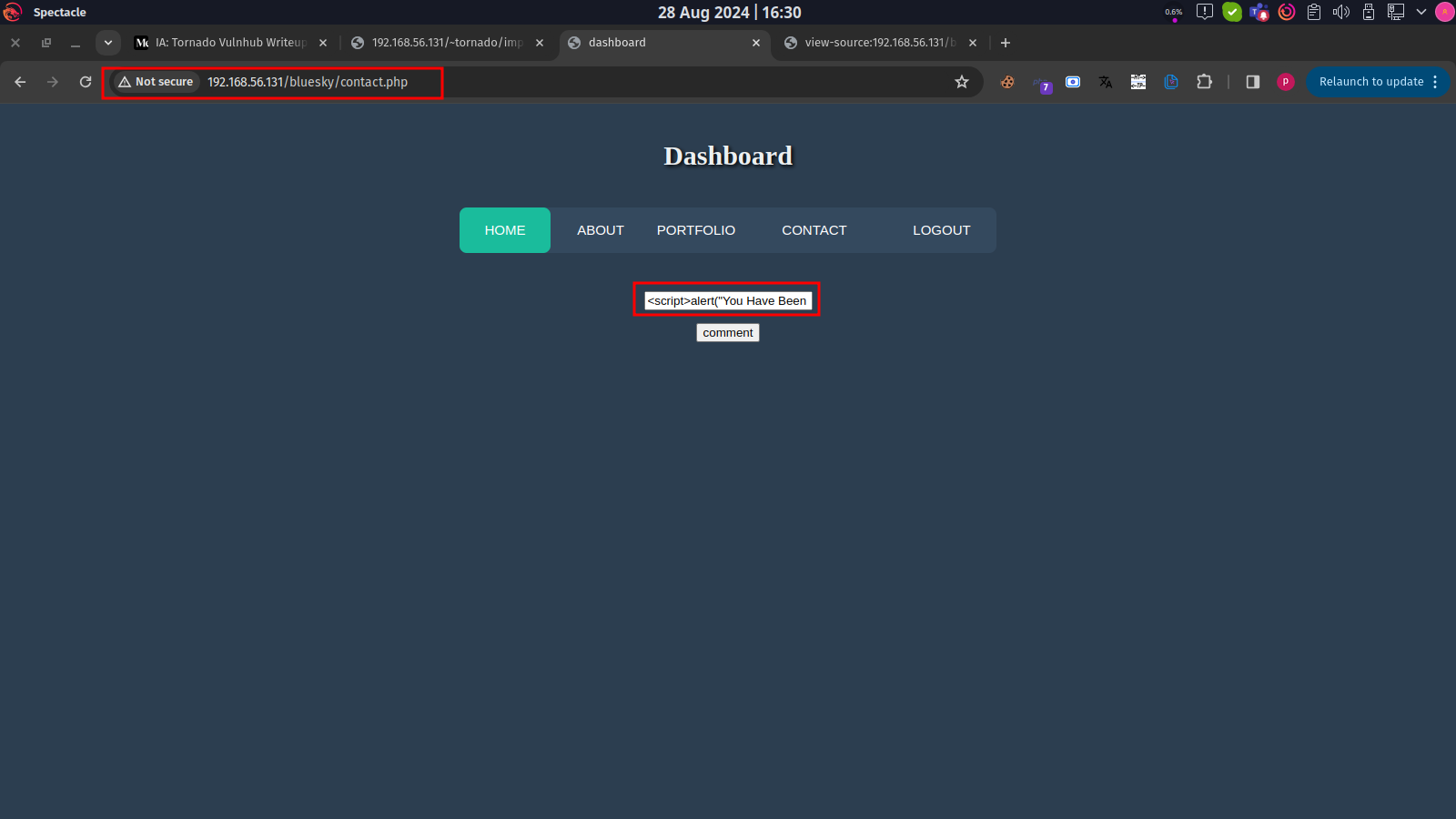


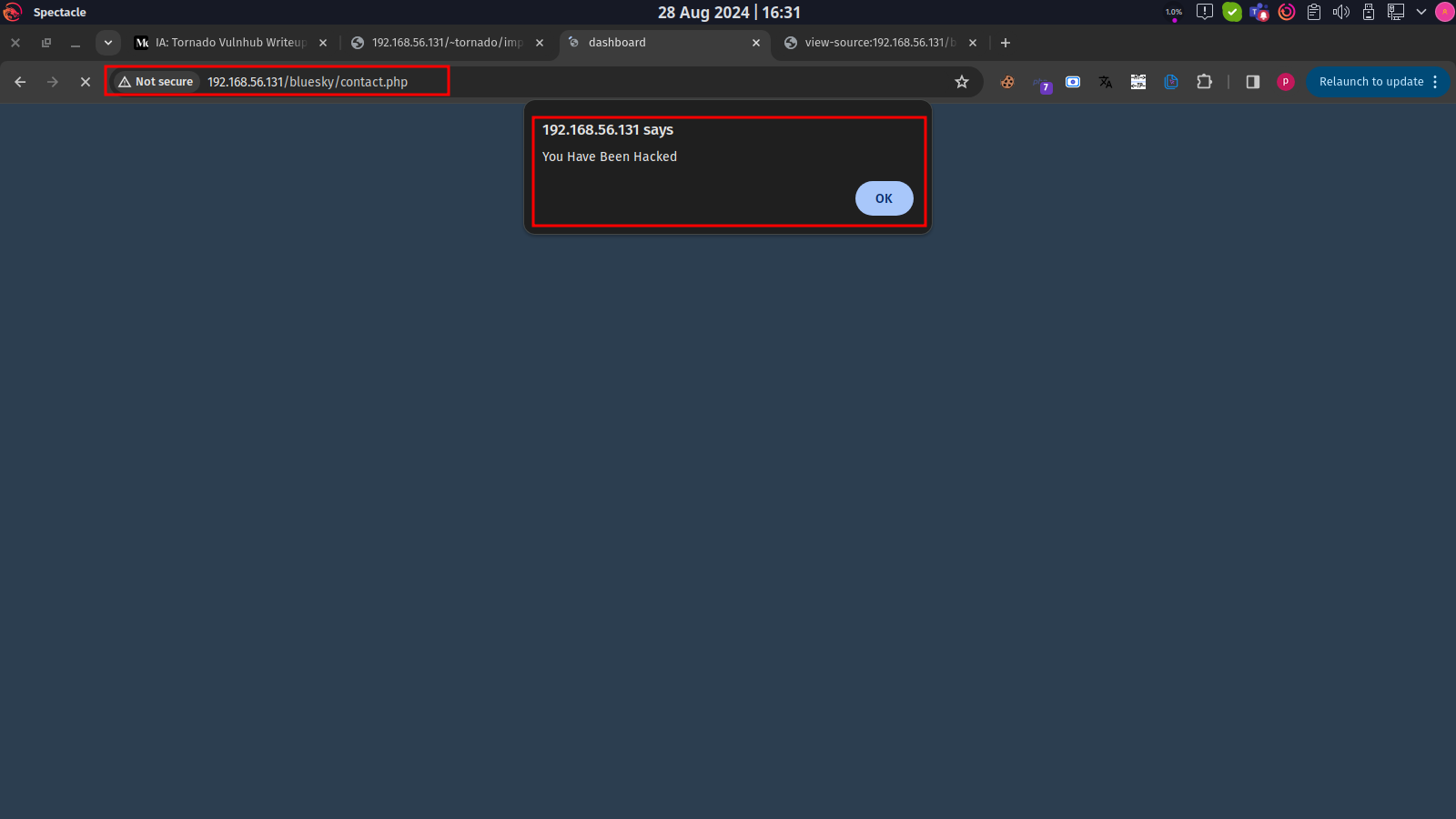


#### Reflected XSS

|  |  |
| --- | --- |
| **Vulnerability** | Reflected XSS |
| **Description** | Reflected XSS is a type of web security vulnerability that allows an attacker to inject malicious scripts into a web application. This occurs when user input (such as URL parameters, form submissions, or HTTP headers) is immediately echoed back to the web page without proper validation or sanitization. The injected script is executed in the context of the victim's browser, often leading to unauthorized actions or data theft. |
| **Risk/Impact** | An attacker might craft a URL that includes a malicious script as a query parameter. When a user clicks on this link, the application reflects the script back in the response, causing the user's browser to execute it. |
| **CVSS Score** | 7.5 High |
| **Path:** | 192.168.56.131 |
| **Remidiation / Solution** | Remidiation:  - Input Validation: Validate and sanitize all user inputs. Ensure that data is in the expected format and reject any unexpected characters or patterns.  - Output Encoding: Encode output before rendering it in the browser. Use context-specific encoding (e.g., HTML, JavaScript, URL encoding) to prevent the execution of injected scripts.  - Content Security Policy (CSP): Implement a robust CSP to restrict the sources from which scripts can be loaded and executed. This can help mitigate the impact of XSS attacks. |
| **Refrence Url:** | <https://owasp.org/www-community/attacks/xss/> |

Below Screenshots shows that attacker input their malicious script and successfuly exploited.

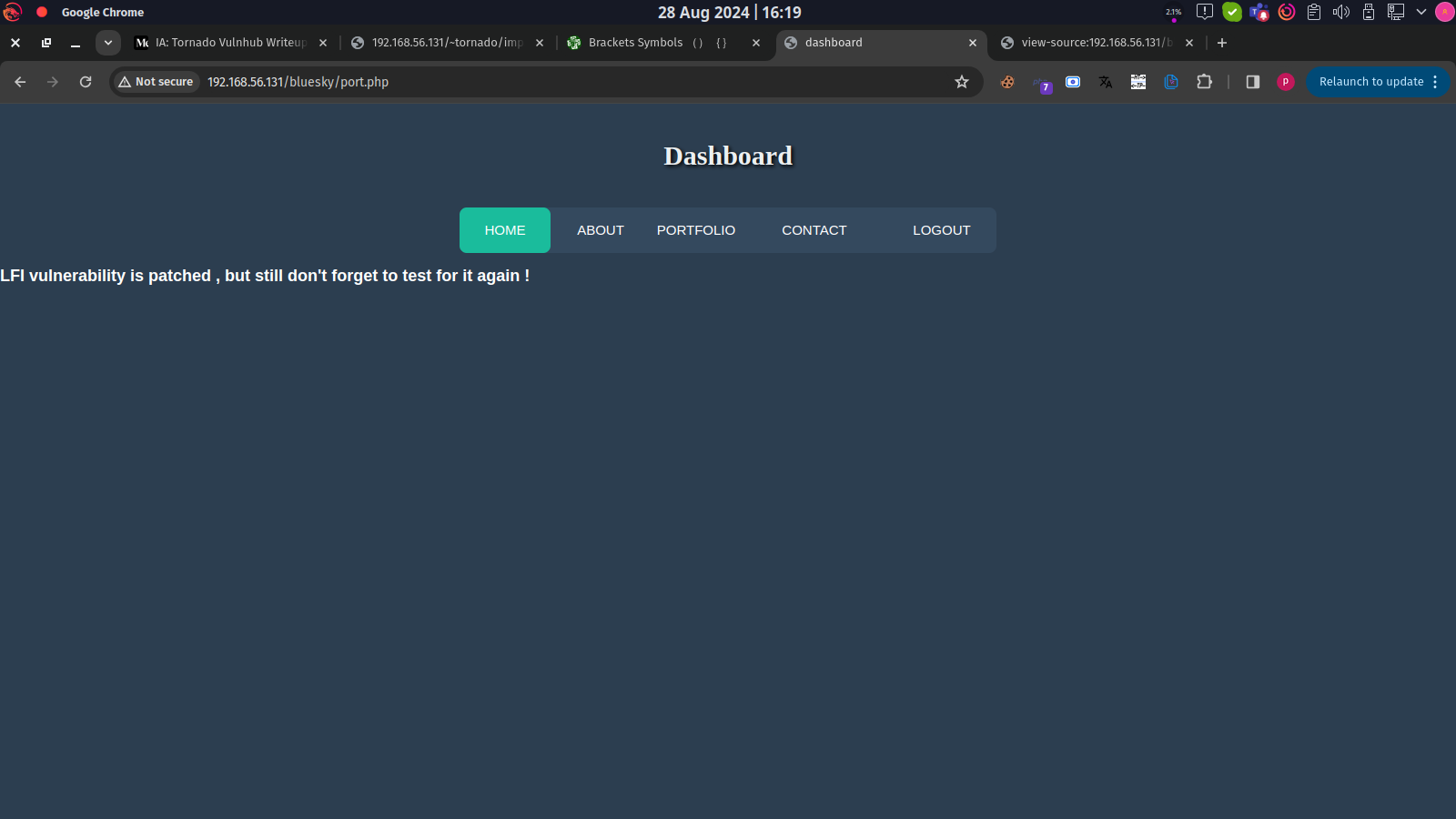


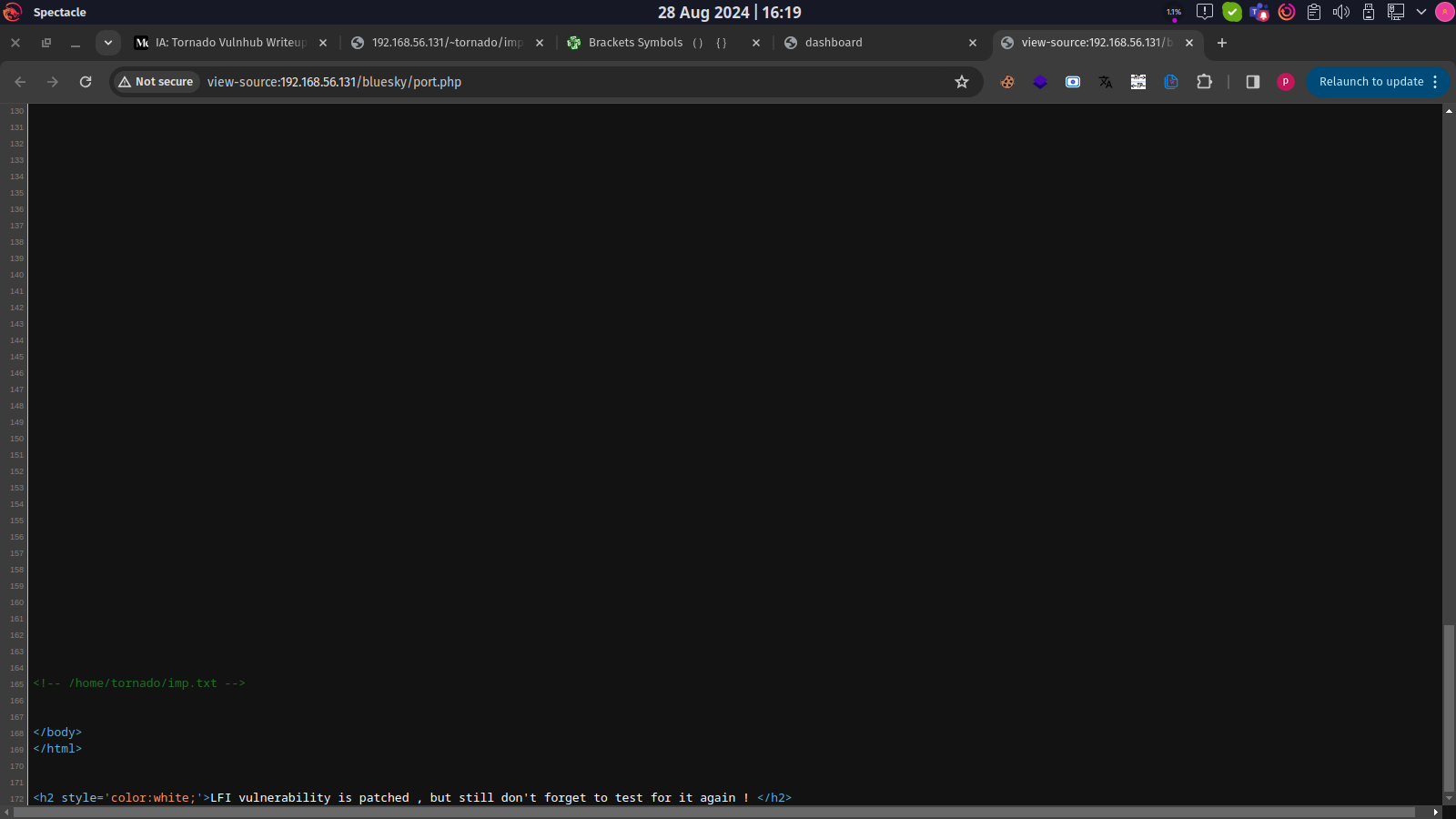


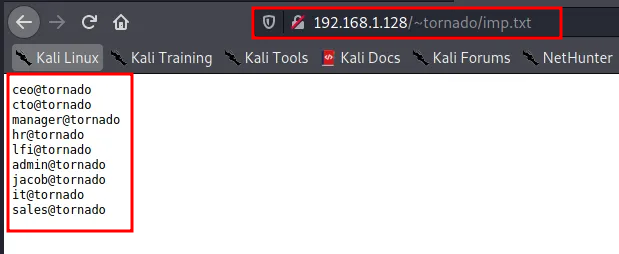
#### Email Enumeration

|  |  |
| --- | --- |
| **Vulnerability** | Email Enumeration |
| **Description** | Email enumeration is a vulnerability that occurs when an attacker can determine whether an email address is registered with a web application based on the application's responses to login or registration attempts. In this case, if an attacker discovers a directory that discloses registered users' email addresses, it indicates a lack of proper access controls and input validation, allowing unauthorized access to sensitive user information. |
| **Risk/Impact** | An attacker submits an email address during registration or password recovery and the application responds differently for registered versus unregistered emails (e.g., "Email sent" for registered emails and "Email not found" for unregistered ones), this provides the attacker with a way to verify which email addresses are valid. |
| **CVSS Score** | 5.0 Medium |
| **Path:** | 192.168.56.131 |
| **Remidiation / Solution** | Remidiation:  - Rate Limiting: Implement rate limiting on login and registration attempts to prevent automated enumeration attacks. This can slow down attackers and make enumeration more difficult.  -Account Lockout Mechanisms: Consider implementing account lockout mechanisms that temporarily disable accounts after a certain number of failed login attempts, but be cautious to avoid denial-of-service (DoS) scenarios.  - Input Validation: Validate email addresses during registration and ensure that error messages do not disclose whether an email is registered or not. |
| **Refrence Url:** | <https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/03-Identity_Management_Testing/04-Testing_for_Account_Enumeration_and_Guessable_User_Account> |

Below Screenshots shows that attacker found hidden directory on source code and they disclouse registerd user emails.

4 (0)161 233 0100





**END OF REPORT**