

**Report CTF**

**CTF Nemisis Penetration Testing Report**

Date: 26/08/2024

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 | 26/08/2024 | 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 hack sudo thor “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 26th of August to 26th 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 Hacksudo Thor 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 | Nemisis | Machine Url: <https://www.vulnhub.com/entry/ia-nemesis-101,582/>  Machine IP ( 192.168.56.129 ) |

### 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

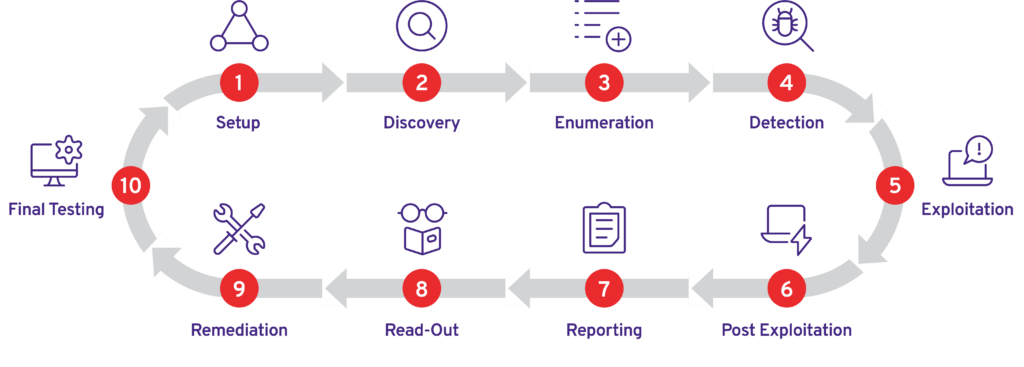
3] Enumeration

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 | 2 |
| 2 | High | 0 |
| 3 | Medium | 0 |
| 4 | Low | 0 |

Table: Representing Severity Level

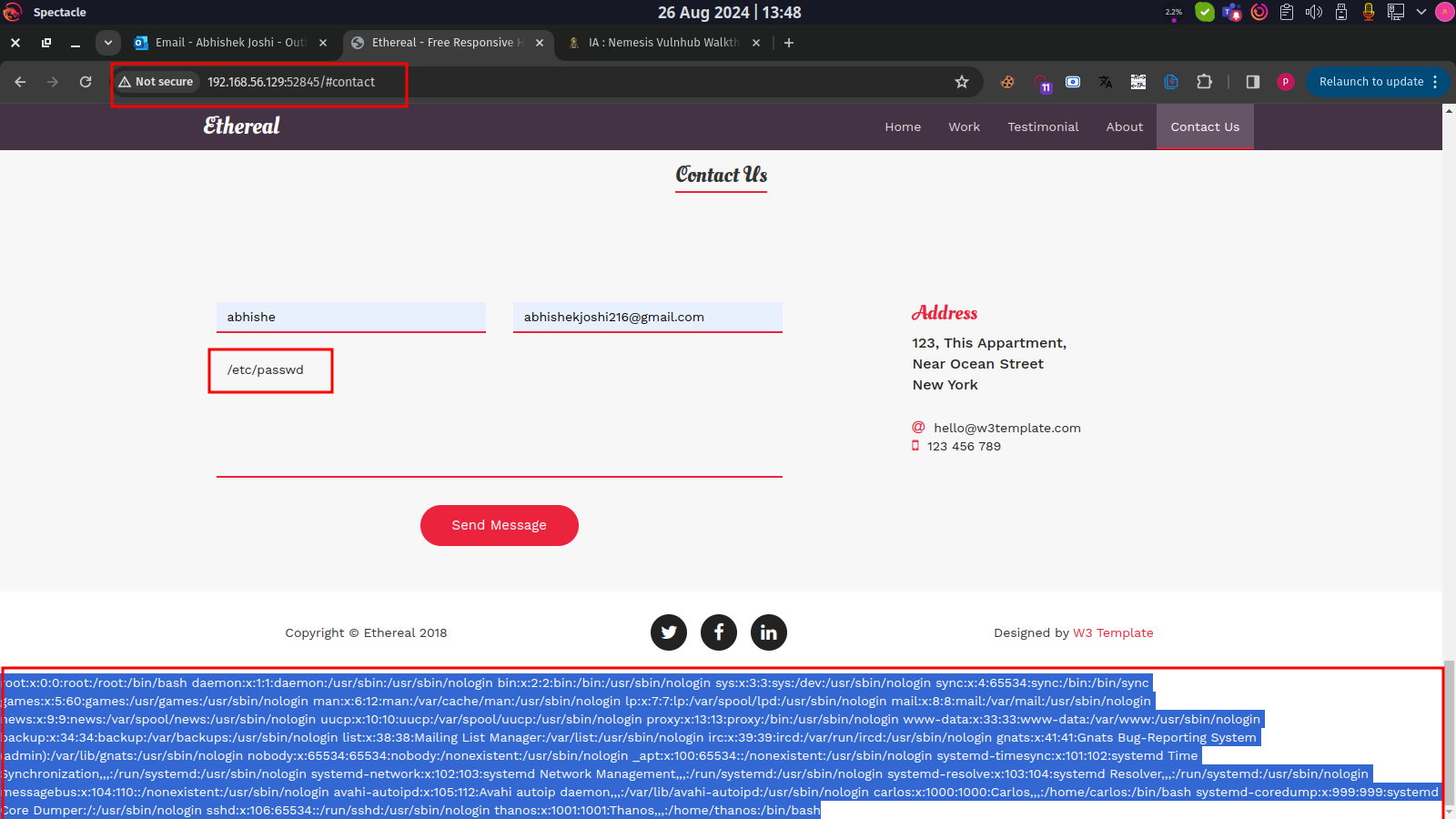
### Findings Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Observed Vulnerability** | **Risk Rating** | **Status** | **Comments** |
| 1. | Command Injection | Critical | Not Fixed | -- |
| 2. | Unauthorized Access to SSH Private Key (Privilege Escalation via Command Injection) | Critical | Not Fixed | -- |

#### Command Injection

|  |  |
| --- | --- |
| **Vulnerability** | Command Injection |
| **Description** | Command Injection is a type of security vulnerability that allows an attacker to execute arbitrary commands on the host operating system via a vulnerable application. This typically occurs when an application passes unsafe user-supplied data (input) to a system shell or command interpreter without proper validation or sanitization.. |
| **Risk/Impact** | Attackers can exploit this vulnerability to execute malicious commands, leading to unauthorized access, data manipulation, or even complete system compromise.. |
| **CVSS Score** | 9.0 Critical |
| **Path:** | 192.168.56.129 |
| **Remidiation / Solution** | Remidiation:  - Implement Input Validation  - Use of Parameterized Commands  - Implement Web Application Firewalls (WAF) |
| **Refrence Url:** | <https://owasp.org/www-community/attacks/Command_Injection> |

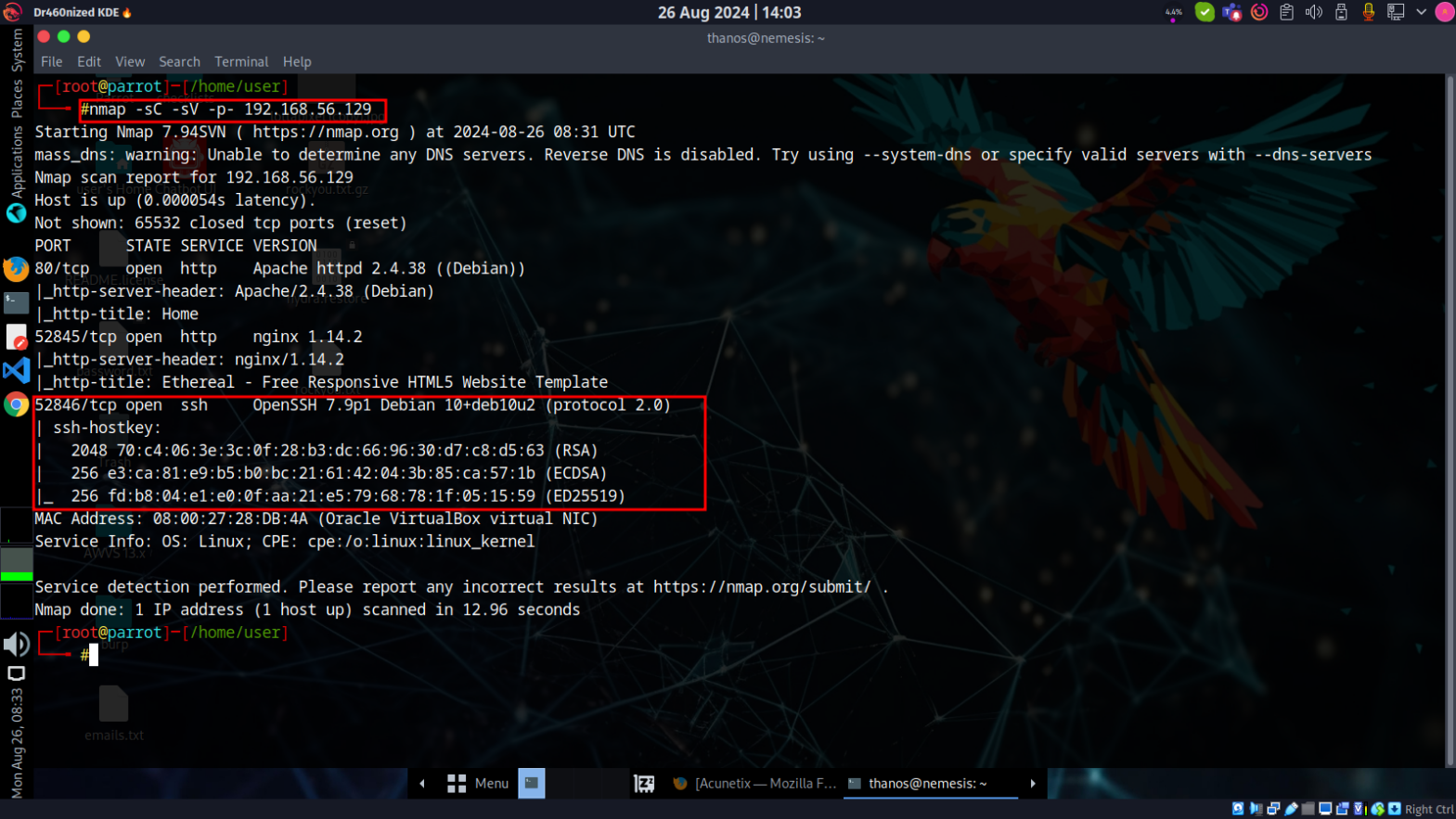
Below Screenshots shows that web app allow special characters and there is no input validation and WAF implemented.



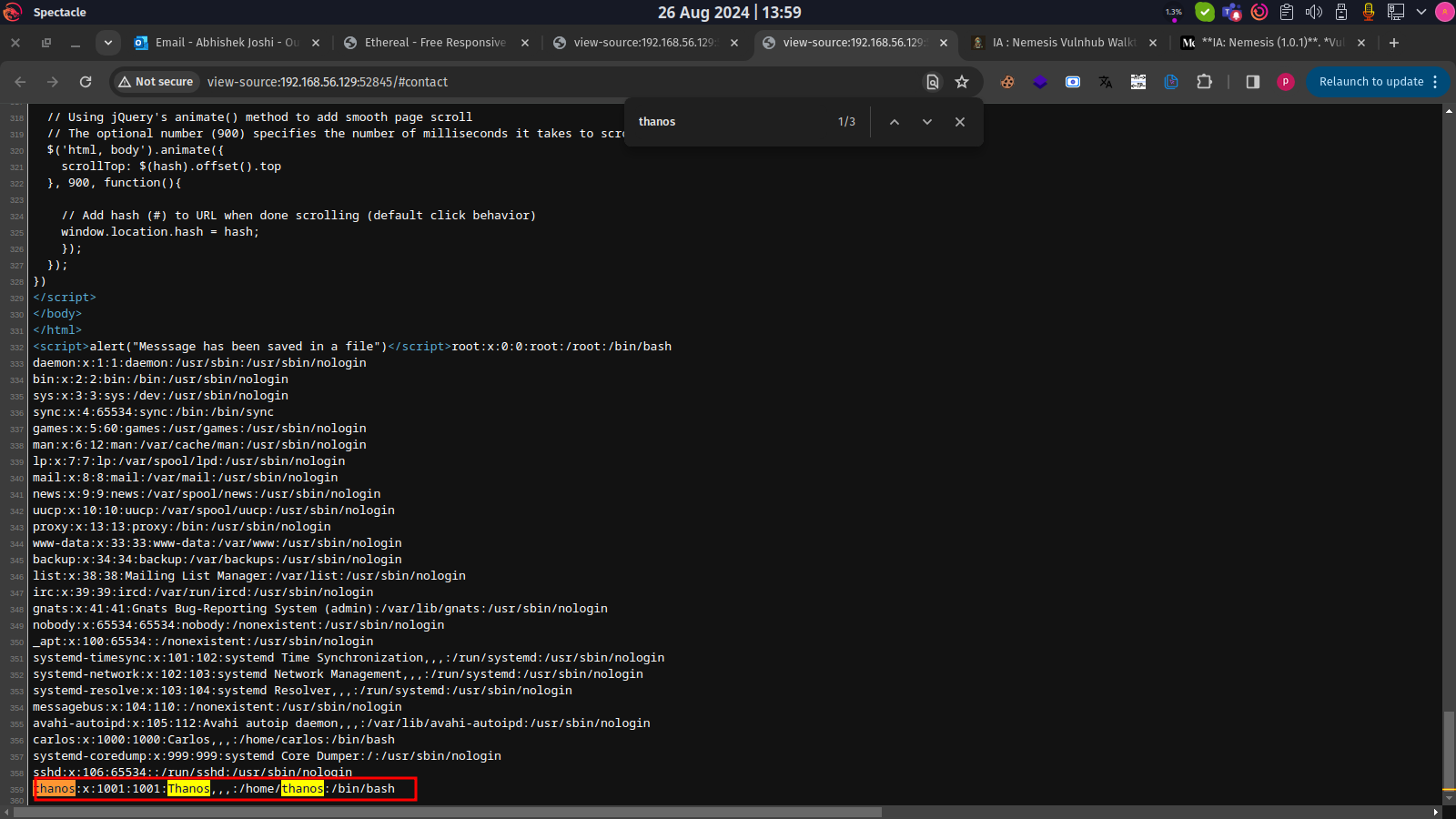
#### Unauthorized Access to SSH Private Key (Privilege Escalation via Command Injection)

|  |  |
| --- | --- |
| **Vulnerability** | Unauthorized Access to SSH Private Key (Privilege Escalation via Command Injection) |
| **Description** | During the exploitation, the attacker discovers an unsecured id\_rsa file, which contains the private SSH key for the system. The attacker then uses this key to access the system via SSH, bypassing traditional authentication mechanisms. |
| **Risk/Impact** | An attacker exploits a command injection vulnerability to gain unauthorized access to the system. |
| **CVSS Score** | 10.0 Critical |
| **Path:** | 192.168.56.129 |
| **Remidiation / Solution** | Remidiation:  - Secure SSH Keys: Ensure that SSH private keys are properly secured and not accessible to unauthorized users.  - Limit SSH Access: Restrict SSH access to only necessary users and systems.  - Monitor SSH Activity: Regularly monitor SSH activity for suspicious behavior.  - Input Validation to prevent command injection  - Implement Web Application Firewall |
| **Refrence Url:** | <https://attack.mitre.org/techniques/T1098/004/> |

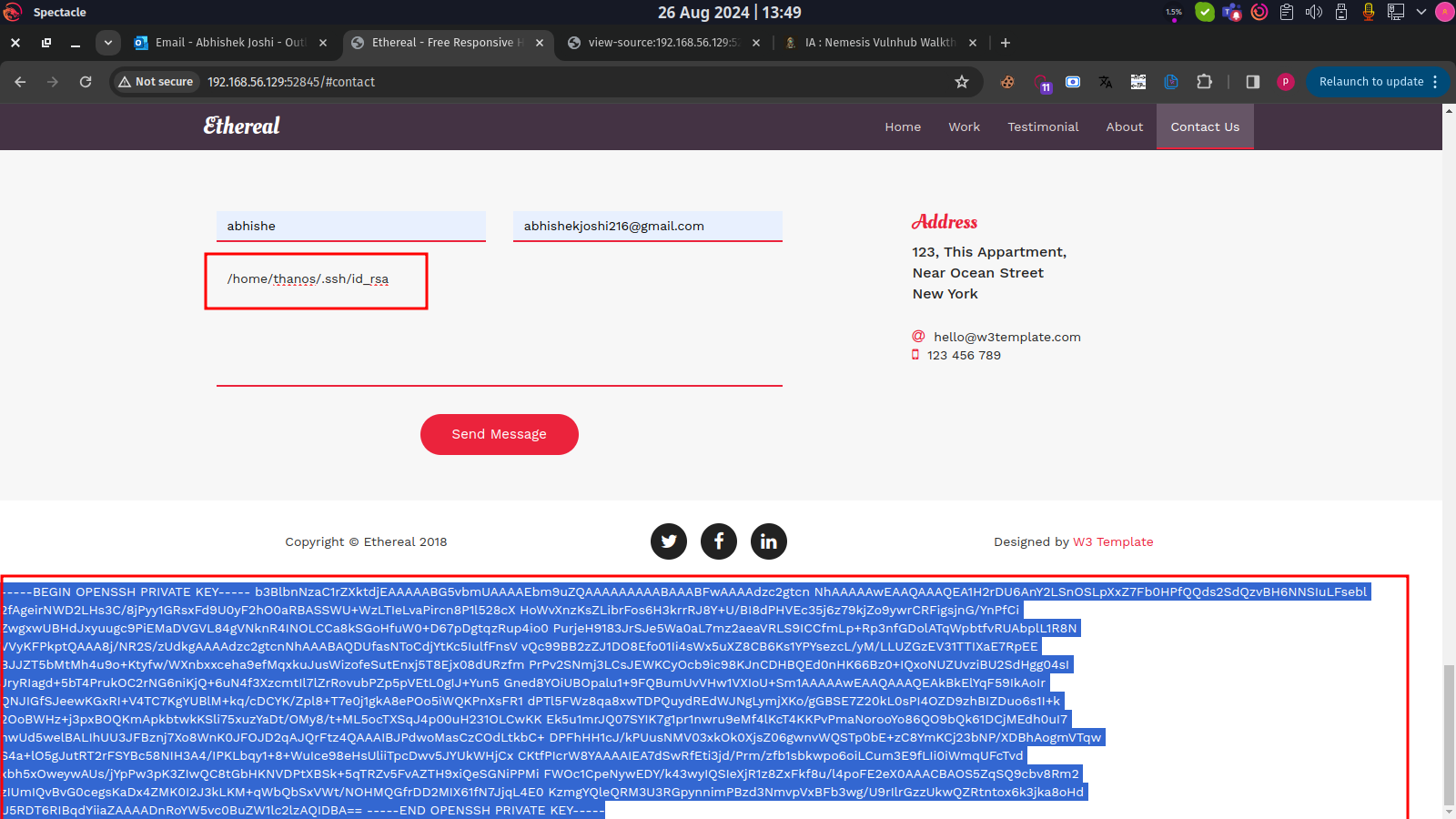
Below Screenshots shows that attacker found SSH Port on diffrent port number “52846”.



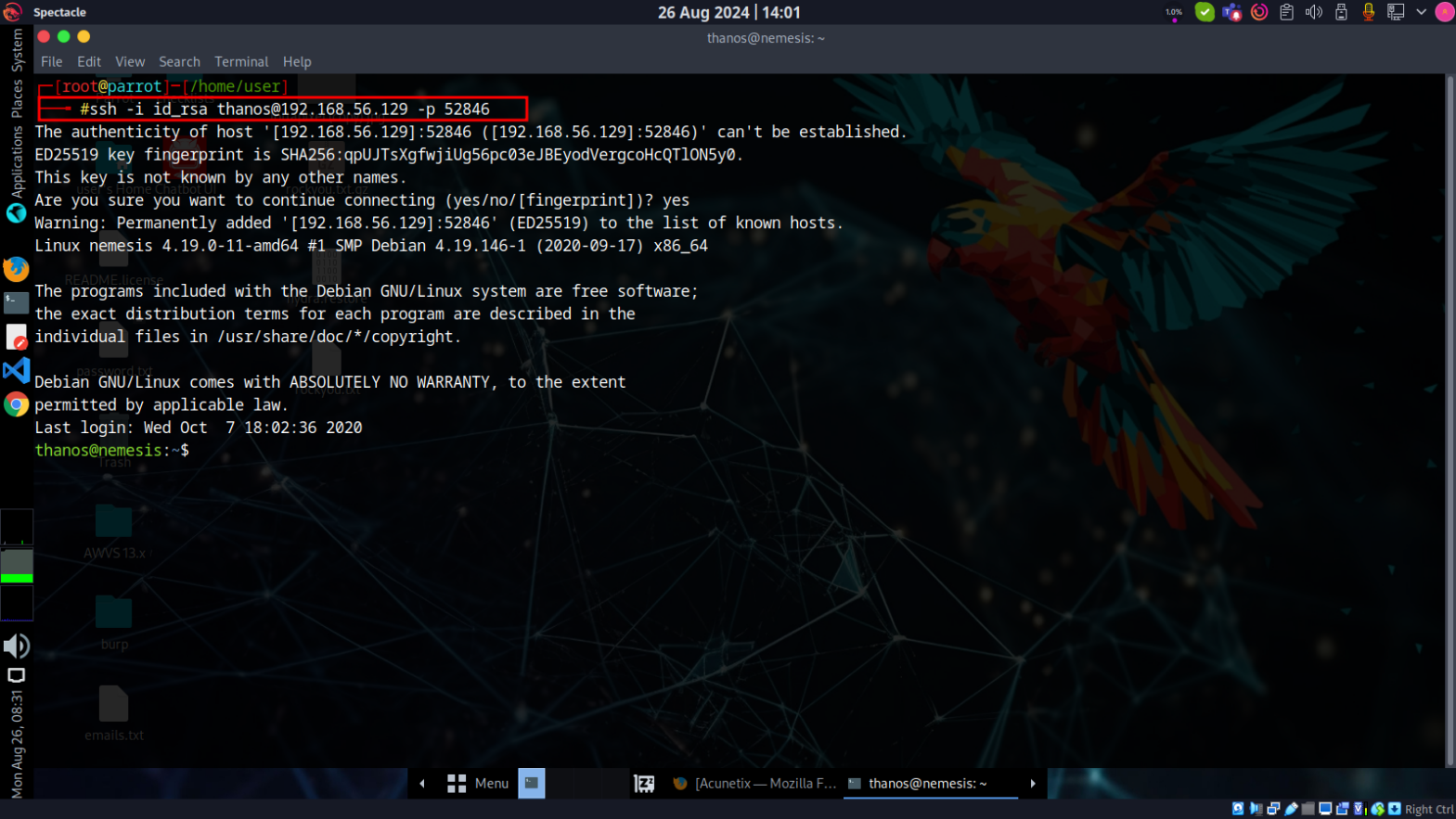
Below Screenshots shows that attacker found command injection is vulnerable so they try to gather usernames using “/etc/passwd” command.



Below Screenshots shows that attacker can find id\_rsa for access SSH using command injection. And attacker copy SSH key and save on id\_rsa file



Below Screenshots shows that attacker use id\_rsa file and succussfully able to access system.



**END OF REPORT**