

**Report CTF**

**CTF Hacksudo Thor Penetration Testing Report**

Date: 20/08/2024

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 | 20/08/2024 | 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 20th of August to 20th 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 | Hacksudo Thor | Machine Url: <https://www.vulnhub.com/entry/hacksudo-thor,733/>  Machine IP ( 192.168.120.118 ) |

### 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

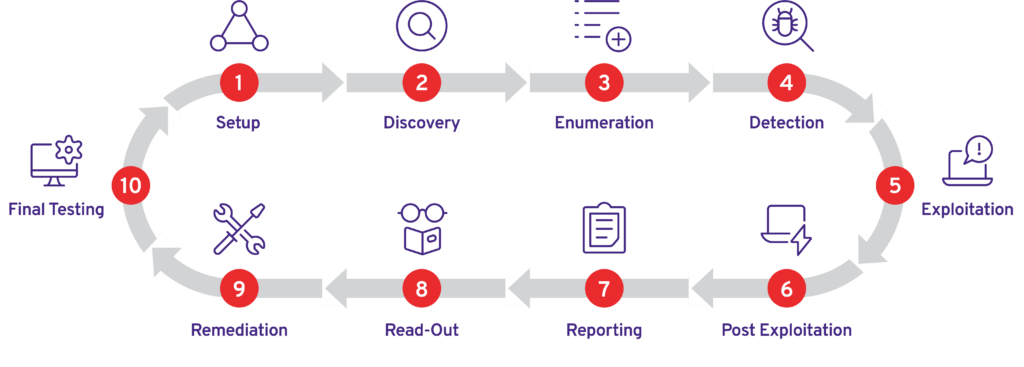
3] Enumeration

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

Table: Representing Severity Level

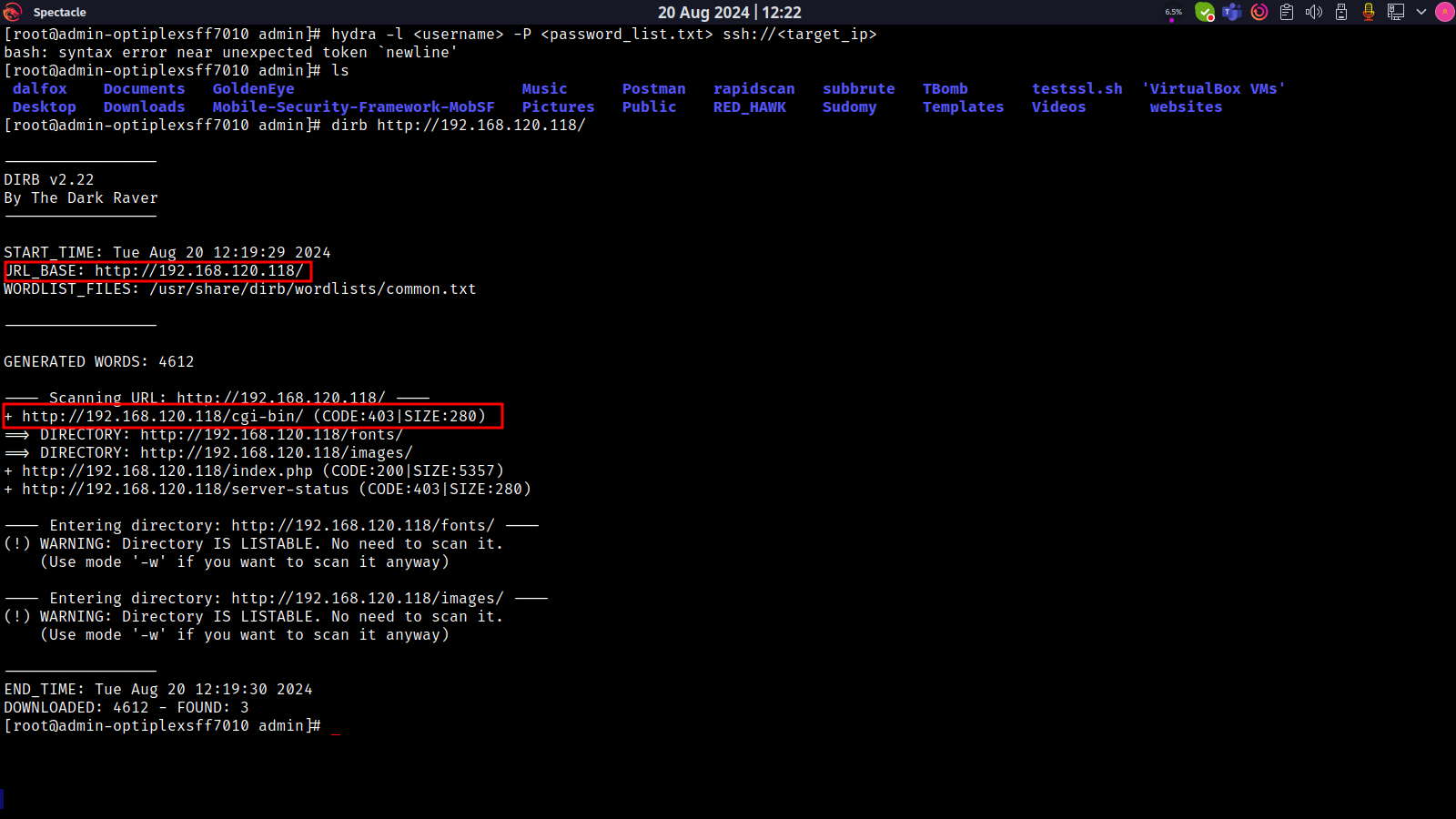
### Findings Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Observed Vulnerability** | **Risk Rating** | **Status** | **Comments** |
| 1. | Shellshock Vulnerability | Critical | Not Fixed | -- |
| 2. | Privilage Escalation | Critical | Not Fixed | -- |
| 3. | Dangerous Open Ports | High | Not Fixed | -- |
| 4. | SSH Brute Force | High | Not Fixed | -- |

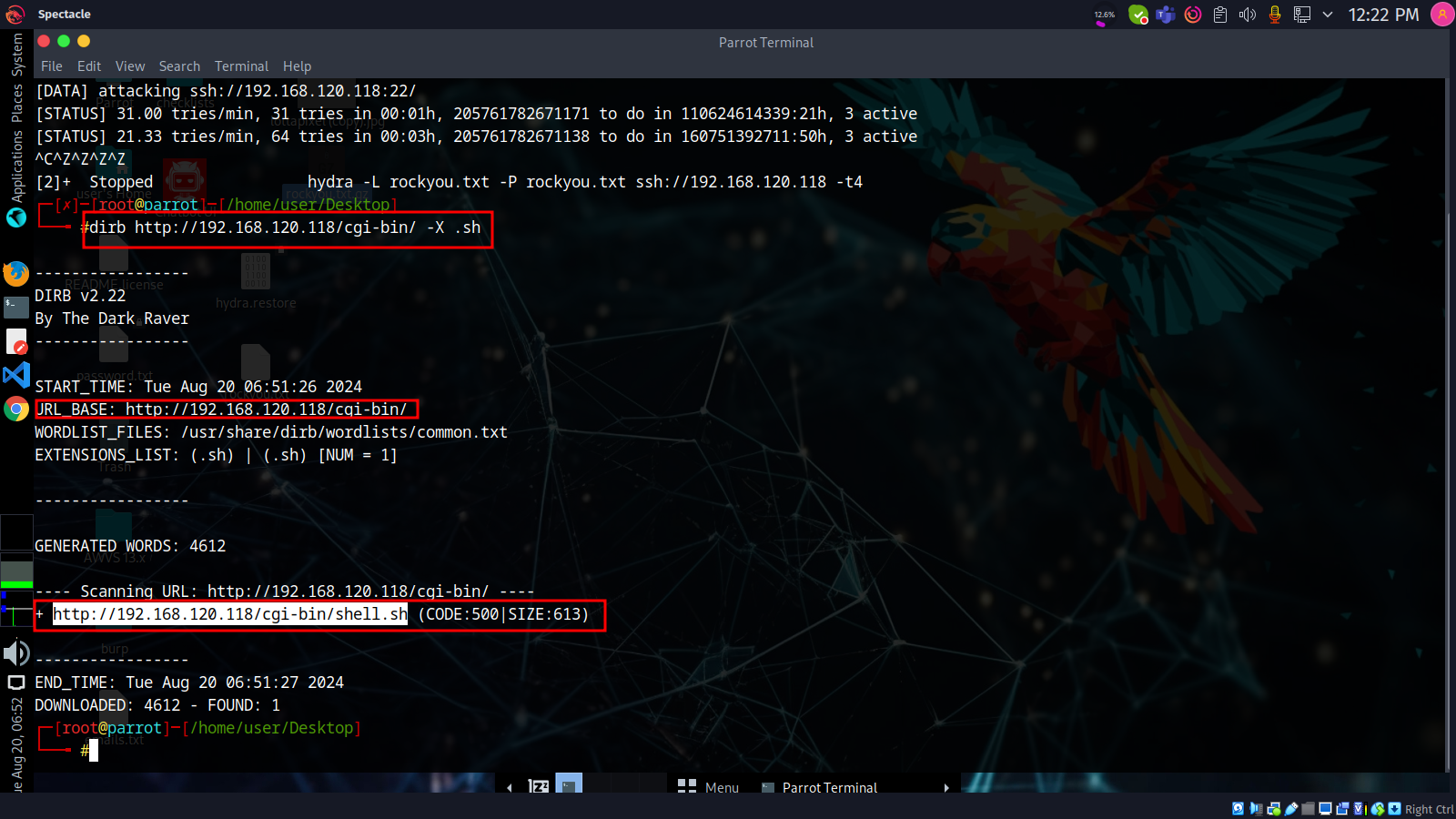
#### Shellshock Vulnerability

|  |  |
| --- | --- |
| **Vulnerability** | Shellshock Vulnerability |
| **Description** | Shellshock is a vulnerability in the Bash shell that allows an attacker to execute arbitrary shell commands by injecting malicious code into environment variables. The vulnerability is caused by a flaw in the way Bash processes environment variables, which can be exploited by an attacker to execute shell commands. |
| **Risk/Impact** | An attacker can exploit this vulnerability using the Shellshock vulnerability (CVE-2014-6271) to execute arbitrary shell commands on the server. |
| **CVSS Score** | 9.5 Critical |
| **Path:** | 192.168.120.118 |
| **Remidiation / Solution** | Remidiation:  - Disable CGI scripts, such as shell.sh, if they are not necessary for the website's functionality.  - Use a Web Application Firewall (WAF)  - Use a secure shell, such as sh or ksh, instead of Bash. |
| **Refrence Url:** | <https://nvd.nist.gov/vuln/detail/cve-2014-6271> |

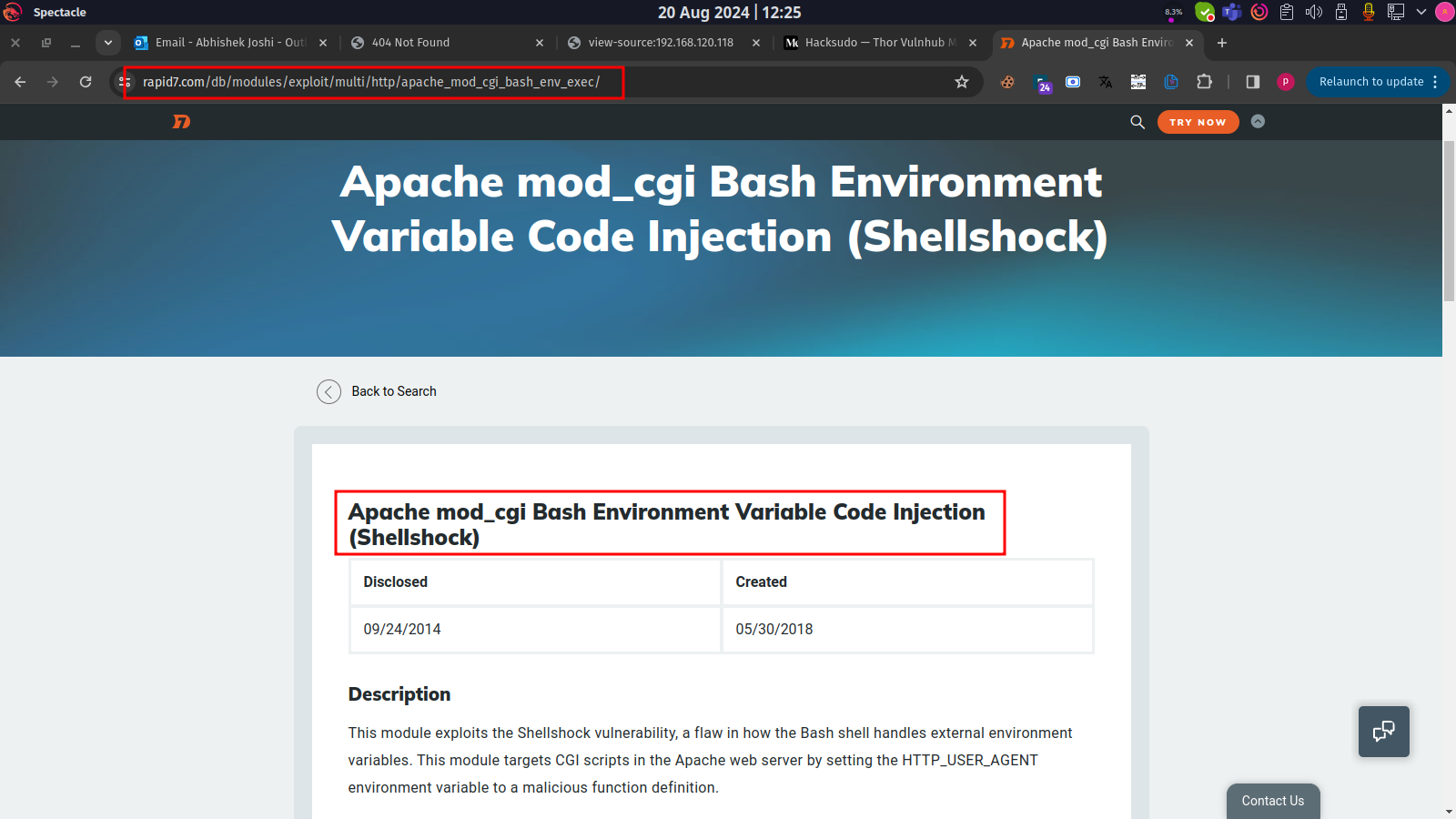
Below Screenshots shows that attacker found cgi-bin direcotry which shows access forbidden.



Below Screenshots shows that attacker found cgi-bin/shell.sh file which shows not accessible but present on system.



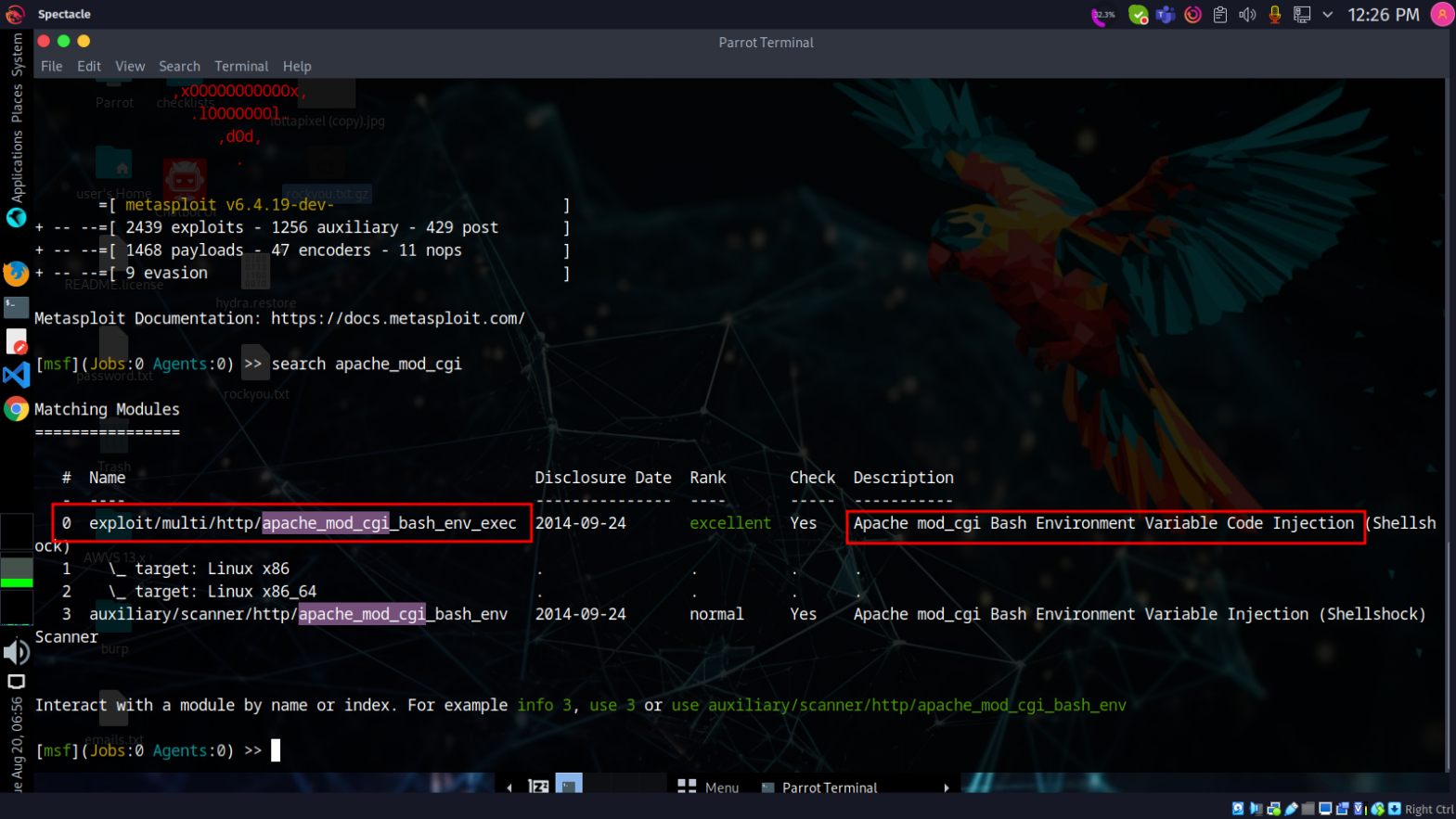
Below Screenshots shows that attacker found exploitation shellshock on online exploitation database.



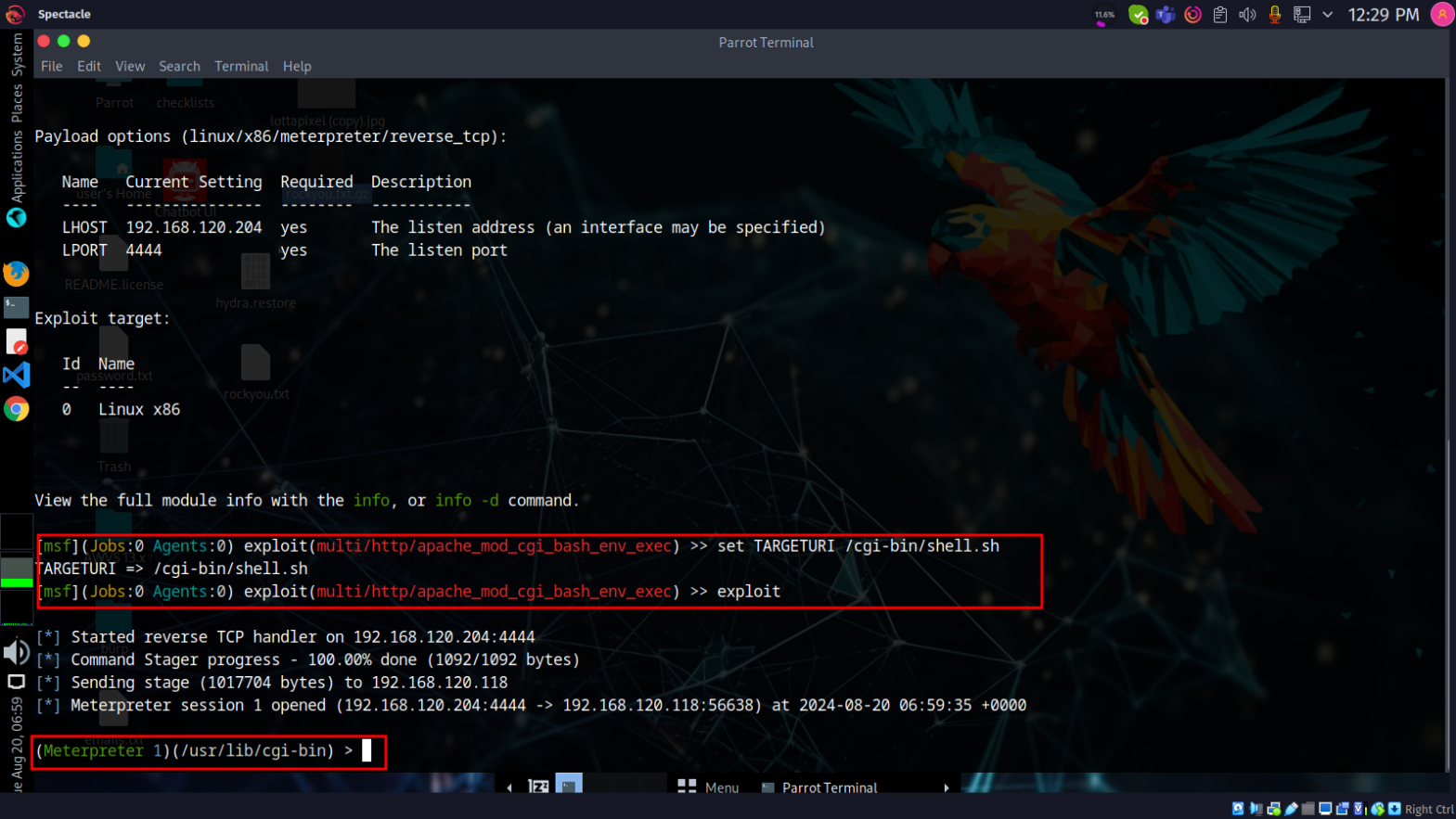
#### Privilage Escalation

|  |  |
| --- | --- |
| **Vulnerability** | Privilage Escalation |
| **Description** | The Shellshock vulnerability (CVE-2014-6271) is a critical vulnerability in the Bash shell that allows an attacker to execute arbitrary shell commands by injecting malicious code into environment variables. |
| **Risk/Impact** | An attacker can exploit this vulnerability to gain shell access to a vulnerable system. |
| **CVSS Score** | 9.5 Critical |
| **Path:** | 192.168.120.118 |
| **Remidiation / Solution** | Remidiation:  - Disable CGI scripts, such as shell.sh, if they are not necessary for the website's functionality.  - Use a Web Application Firewall (WAF)  - Use a secure shell, such as sh or ksh, instead of Bash. |
| **Refrence Url:** | <https://nvd.nist.gov/vuln/detail/cve-2014-6271> |

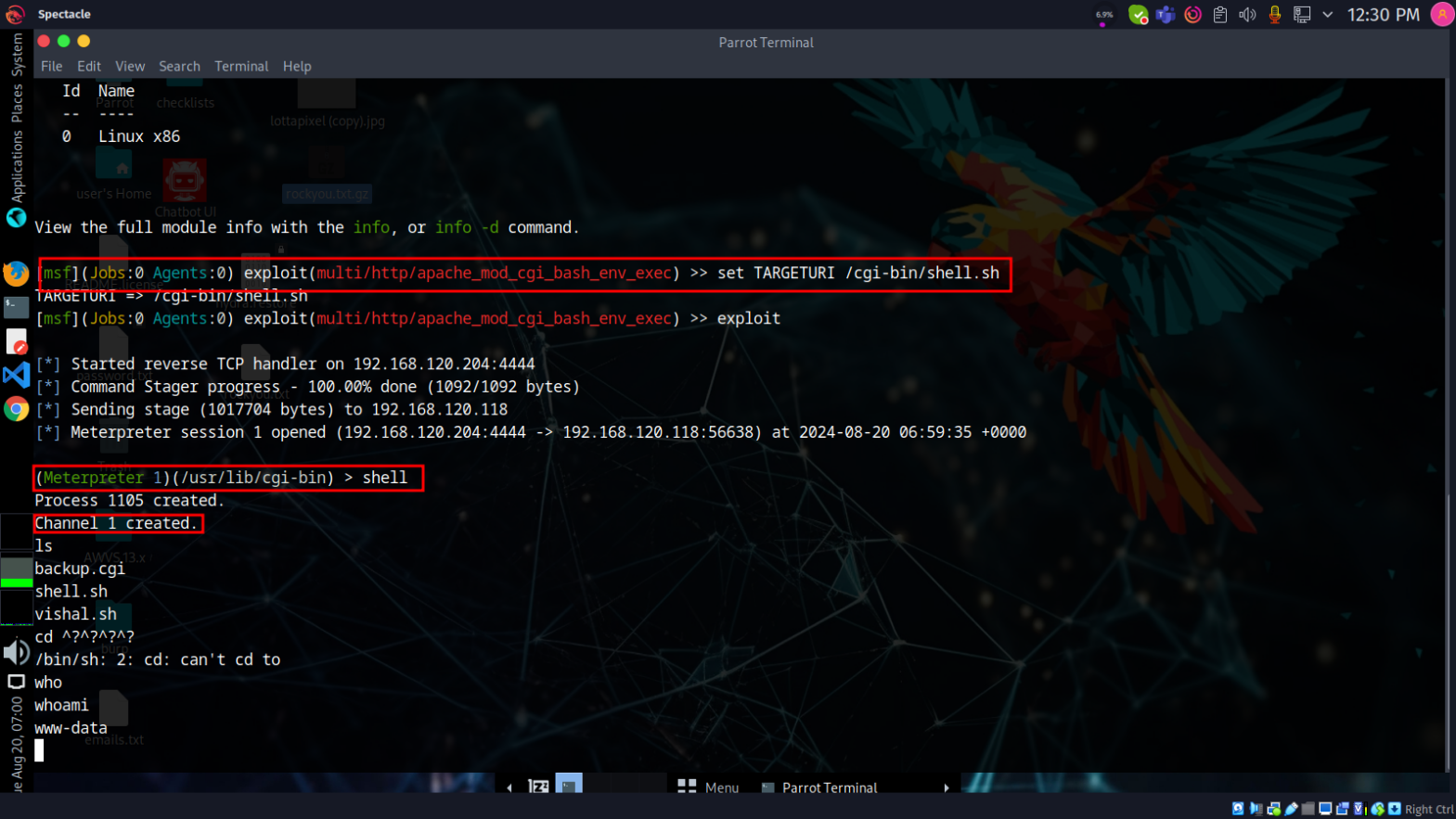
Below Screenshots shows that attacker use metasploit to exploit shellshock vulnerability.



Below Screenshots shows that attacker successfully exploit shellshock vulnerability.



Below Screenshots shows that attacker can gain Shell acess which means attacker now takeover whole system.



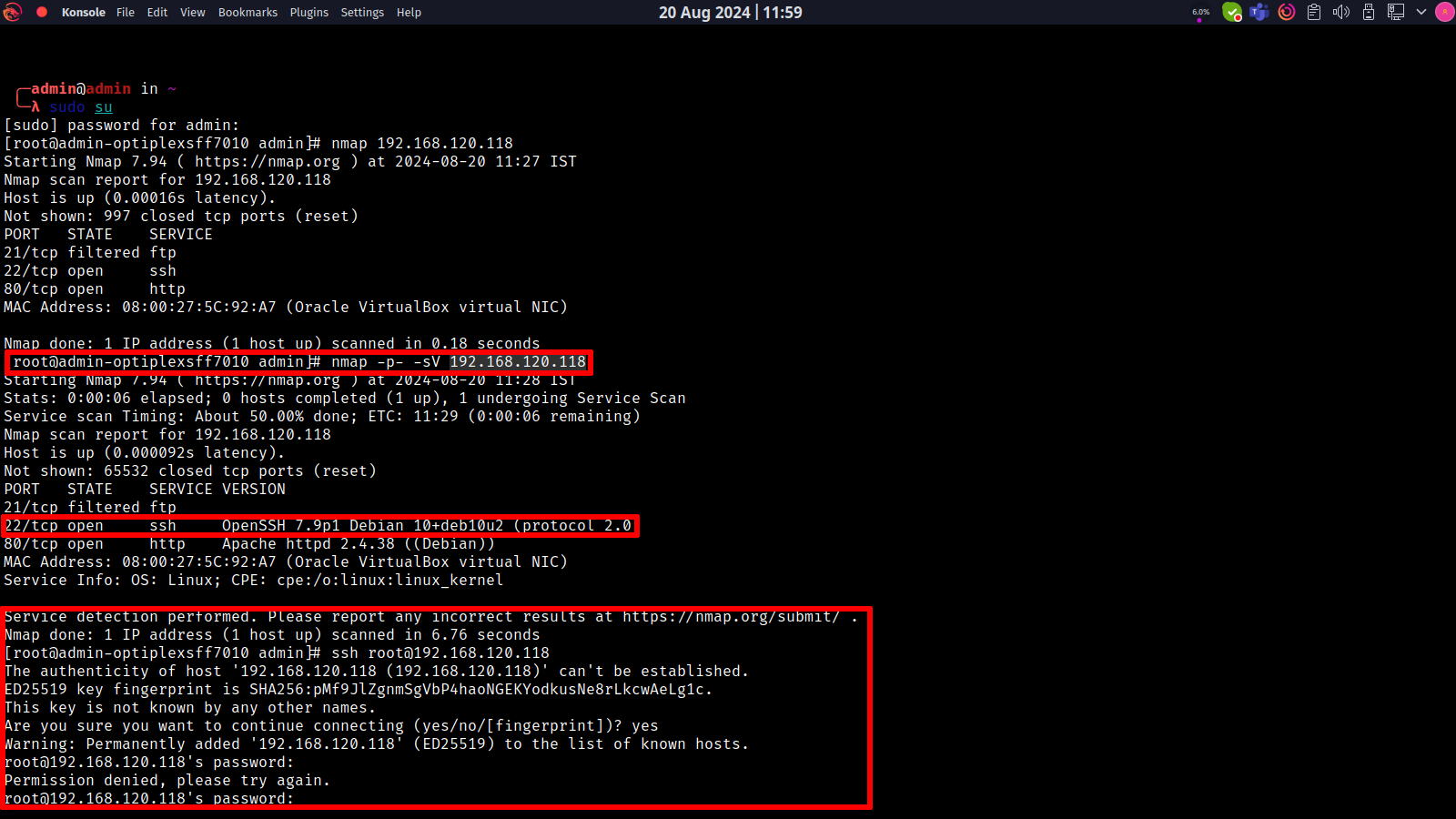
#### Dangerous Open Ports

|  |  |
| --- | --- |
| **Vulnerability** | Dangerous Open Ports |
| **Description** | Open Port SSH (Secure Shell) is a protocol used for secure remote access to a system. However, if the SSH port (typically port 22) is left open and not properly secured, it can provide an entry point for attackers to exploit vulnerabilities in the SSH server or its configuration. |
| **Risk/Impact** | An attacker able to communicate SSH protocol and try to brute force or gain access of system. |
| **CVSS Score** | 7.5 High |
| **Path:** | 192.168.120.118 |
| **Remidiation / Solution** | Remidiation:  - Use a secure SSH protocol version (e.g., SSH-2).  - Disable password authentication and use public key authentication instead.  - Limit the number of authentication attempts.  - Set a secure umask (e.g., 077) to restrict file permissions. |
| **Refrence Url:** | <https://www.upguard.com/blog/open-port> |

26a, The Downs +

Below Screenshots shows that machine have sensitive open port “SSH” which is communicable.

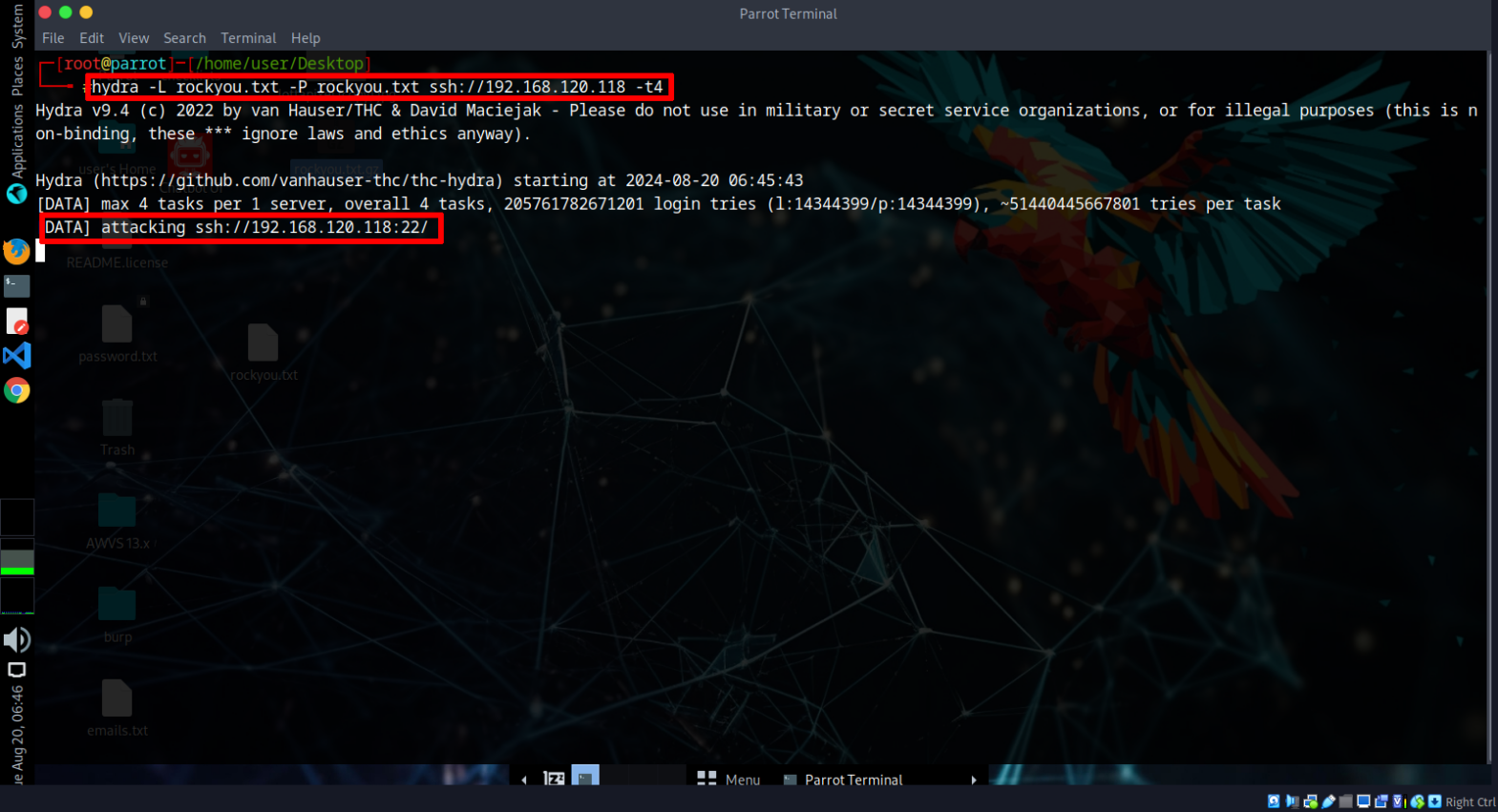
4 (0)161 233 0100



#### SSH Brute Force

|  |  |
| --- | --- |
| **Vulnerability** | SSH Brute Force |
| **Description** | SSH Brute Force is a type of attack where an attacker attempts to guess the password of a valid user account on a Secure Shell (SSH) server by systematically trying a large number of passwords. This attack can be performed manually or using automated tools, such as password cracking software. |
| **Risk/Impact** | An attacker can gain access to the system, potentially leading to data breaches, malware infections, or lateral movement within the network by using brute forcing attack on SSH protocol. |
| **CVSS Score** | 7.0 High |
| **Path:** | 192.168.120.118 |
| **Remidiation / Solution** | Remidiation:  - Implement strong passwords.  - Limit login attempts such as max 5 login attempts and once login attempts reach user should be blocked.  - Monitor SSH logs.  - Implement Firewall and blacklist unauthorized IPs. |
| **Refrence Url:** | <https://nvd.nist.gov/vuln/detail/CVE-2020-1616> |

Below Screenshots shows that attacker able to perform brute forcing on SSH and trying to gain access.



**END OF REPORT**