LAB 1 PART 1: INFORMATION GATHERING AND EXTRACTING UNENCRYPTED DATA

TASK 1: SNIFF LOGIN DETAILS FROM UNENCRYPTED HTTP

- Data transmitted through HTTP- vulnerable to interception, enabling attackers to eavesdrop on unencrypted packets + extract sensitive info
- Explore the extraction of data transmitted over unencrypted HTTP connections

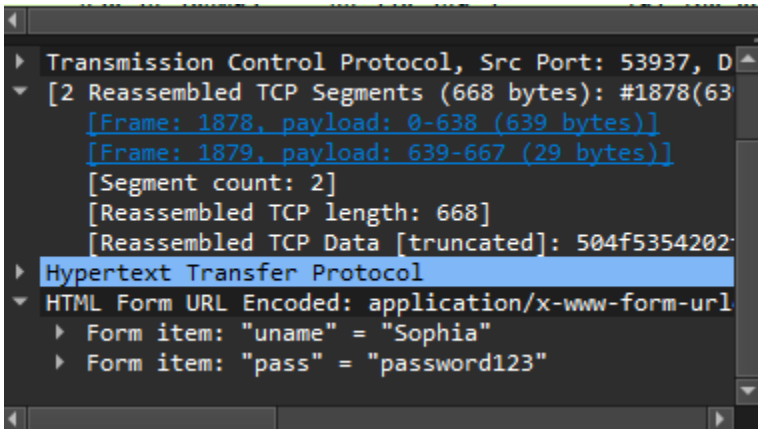
STEP 1

a. Open wireshark, ensure promiscuous mode is enabled on all interfaces
b. Start capturing files
c. Go to hΣp://testphp.vulnweb.com/login.php
d. Enter name on the user, enter “password123”

STEP 2

a. Utilise search bar to locate the packet with the login info (packet will be using HTTP protocol and will start with POST)
b. In wireshark, locate login details which are the username and the password used (write “frame contains “POST”” in search bar)

SCREENSHOT OF USERNAME + PASSWORD FROM WIRESHARK



- Filtered under HTTP, looked under HTML form URL encoded section

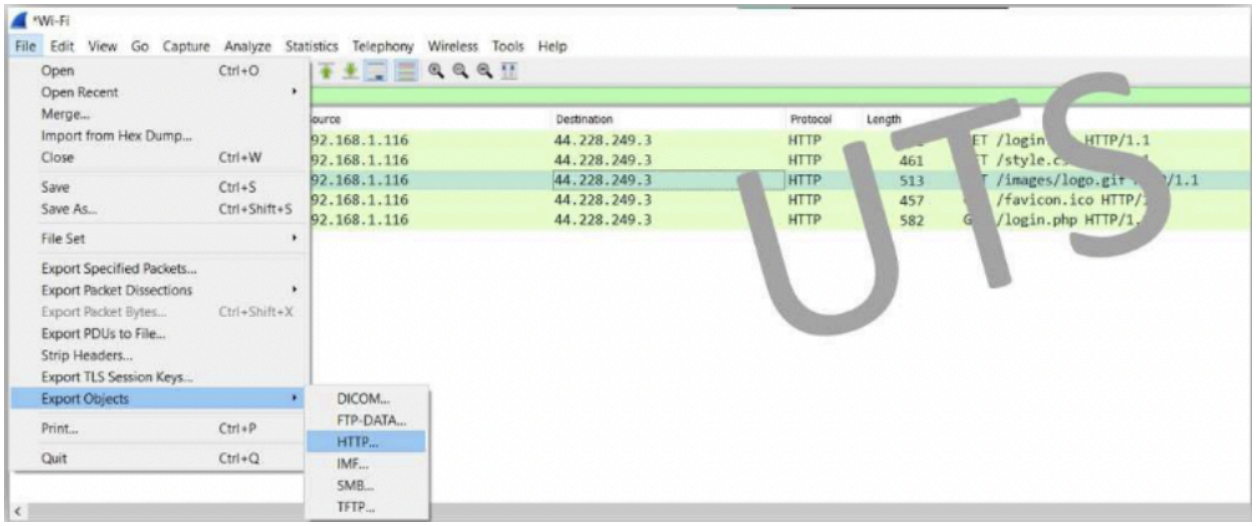
TASK 2: EXTRACT AN IMAGE FROM UNENCRYPTED HTTP PACKET

- Wireshark can extract images from unencrypted packets transmitted through websites that use HTTP protocol/unencrypted protocols

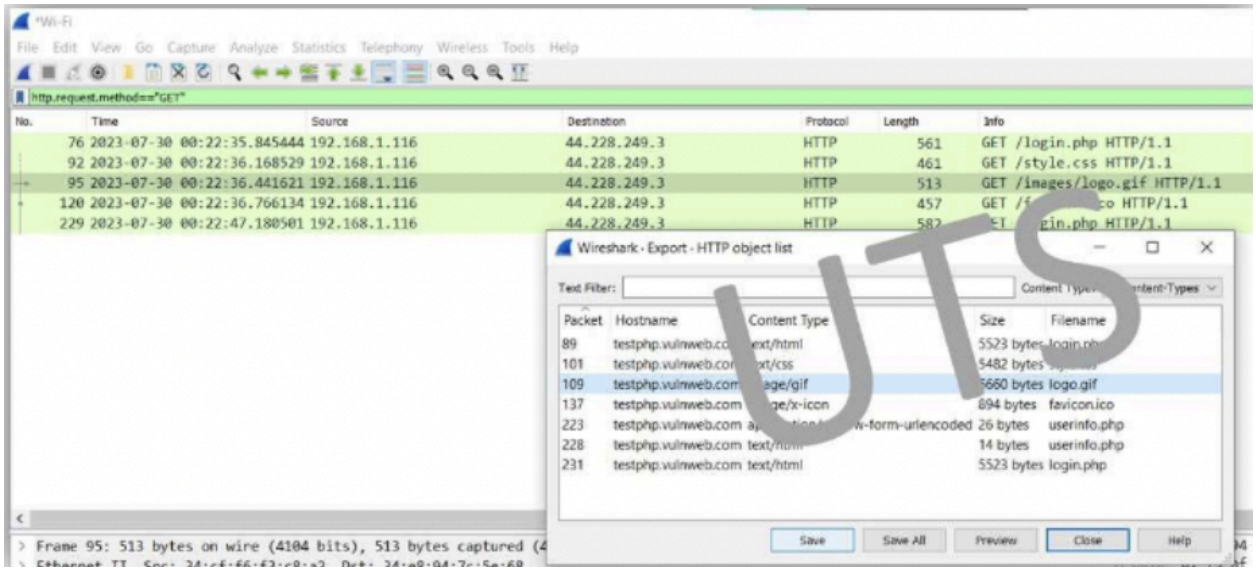
STEPS:

- Extract logo image of the same website^^
- Use the same packets captured from task 1

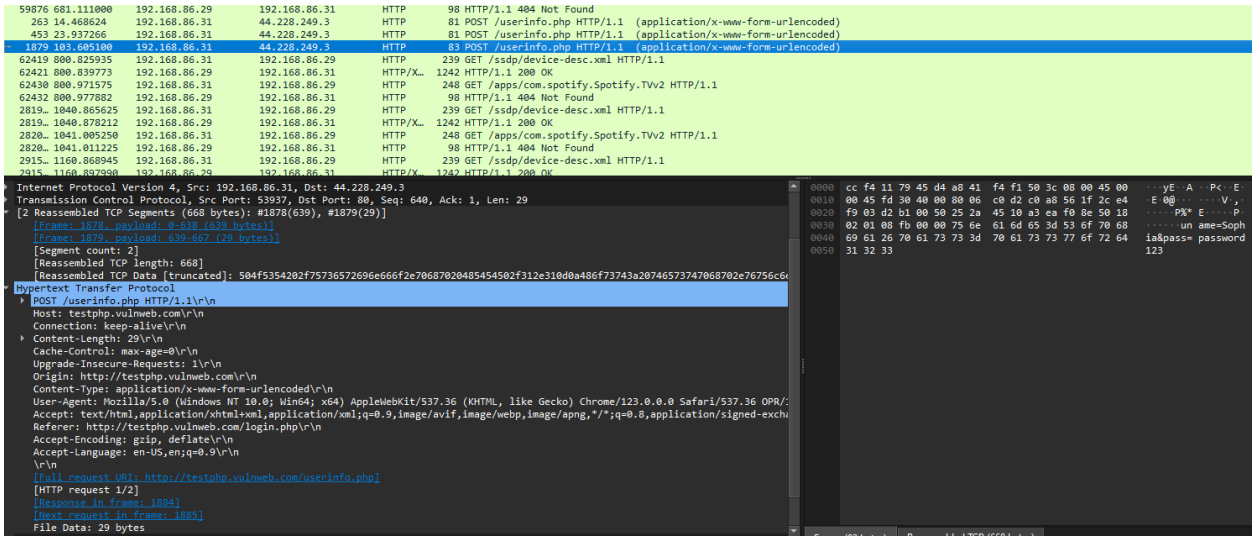
a. Locate the GET packet that contains the logo.gif raw data
b. Click on the packet with the image info and go to file a export objects a HTTP



c. Download the object that contains the logo.gif data



SCREENSHOT OF THE IMAGE LOGO.GIF IN THE HTTP OBJECT LIST



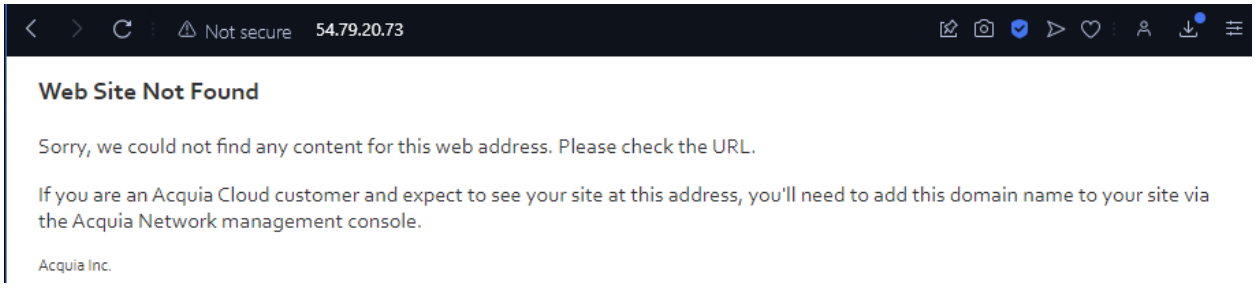
- Tried clicking on the logo on the web page, but got redirected to another site

TASK 3: INFORMATION GATHERING

- Passive information gathering: can be used in a direct attack/reinforce other attacks- risks the security of an organisation
- Active information gathering: collects current data (location, ISP, network constraints)- investigates current state of target
- Zenmap: network discovery, security auditing- useful for network inventory, managing service upgrade schedules and monitoring host or service uptime
- Netcraft: internet services company that provides internet security services (antifraud, anti-phishing services, application testing, code reviews, automated penetration testing)

QUESTIONS (using zenmap/netcraft to scan www.uts.edu.au)
https://sitereport.netcraft.com/?url=http://uts.edu.au

1. IP address: 54.79.20.73
2. Type IP address in browser, explain findings:



- Website can't be found: since the web server doesn't know the domain name, and it can't be specified by only putting the IP address

3. IP owner: Amazon Corporate Services Pty Ltd
4. Server's operating system: Linux
5. Type of web server being used: Apache

6. Server-side scripting technology: PHP
7. Email for the domain admin of the website (phishing attack): unavailable due to security reasons?
8. Reverse DNS for the website: ec2-54-79-20-73.ap-southeast-2.compute.amazonaws.com (uts.edu.au)
9. The domain registrar: audns.net.au
10. Nameserver organisation: whois.audns.net.au
11. Company hosting the website: Amazon - Asia Pacific (Sydney) Datacenter
12. Location of the hosting company: Sydney, NSW

LAB 1 PART 2: PASSWORD CRACKING USING JOHN THE RIPPER AND SQL INJECTION

- John the Ripper: auto detects the encryption on the hashed data, compares it against a large plain-text file containing popular passwords, hashing each password then stopping when it matches

TASK 4: USING JOHN THE RIPPER

STEP 1

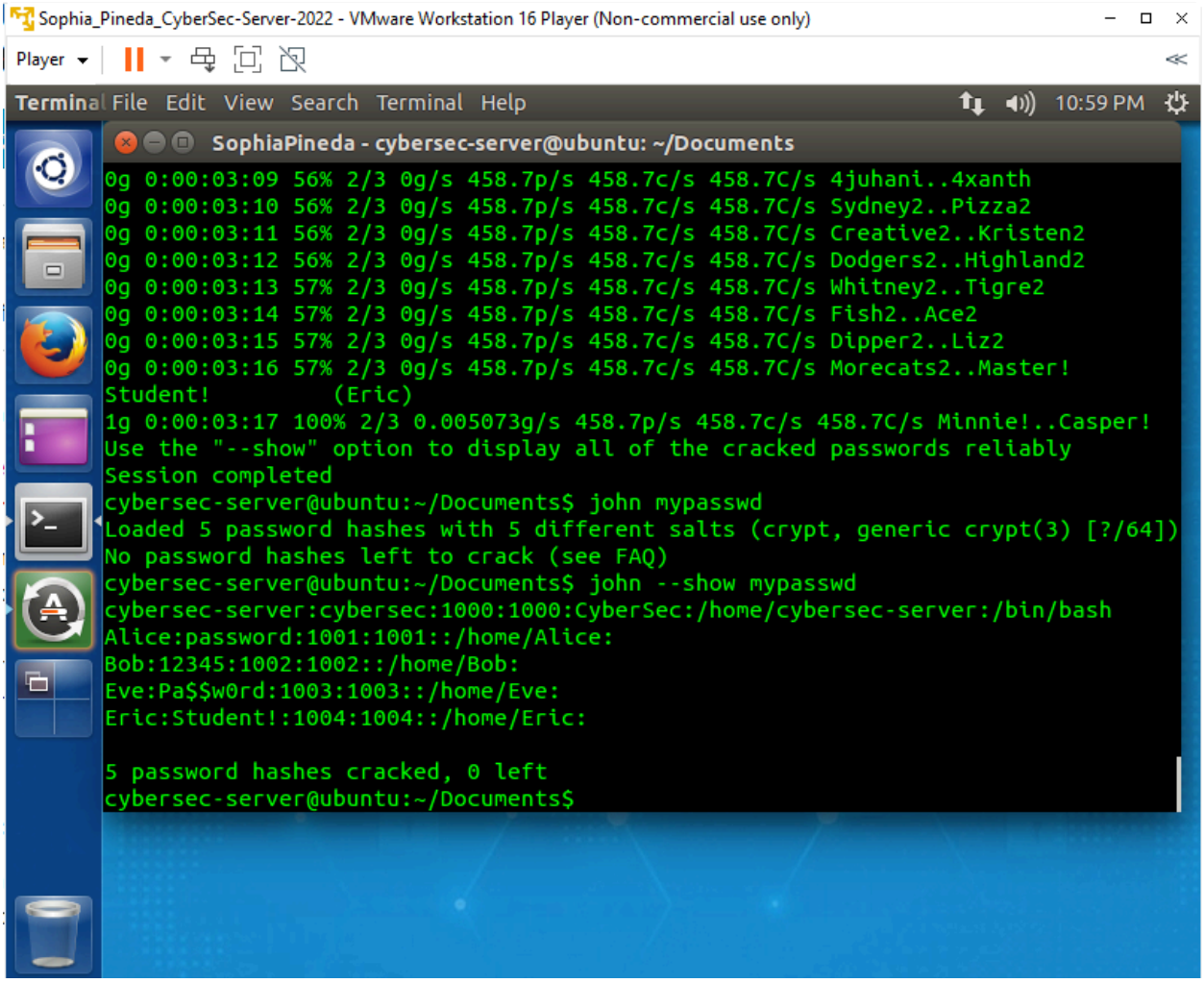
a. Change to documents directory
b. Check contents of “mypasswd” file

STEP 2- RECOVERING PASSWORDS

a. Type john –show mypasswd
b. Enter john mypasswd to enter crack mode
c. Use control c to abort the program
d. Type in john –show mypasswd again to check the file
e. Type in john mypasswd –wordlist=”password.lst” to retrieve password hashes

PASSWORD FOR USER ERIC

- Student!
- The password cracking process involves decrypting hashed data and comparing it against a text file that contains popular passwords. To create a secure password is to ensure that the attacker would have difficulty deciphering the hashed data.



SQL INJECTION

- Vulnerability in SQL query exploited (stems from interaction between user and application)
    - User inputting additional data
- Web application lacks proper input validation

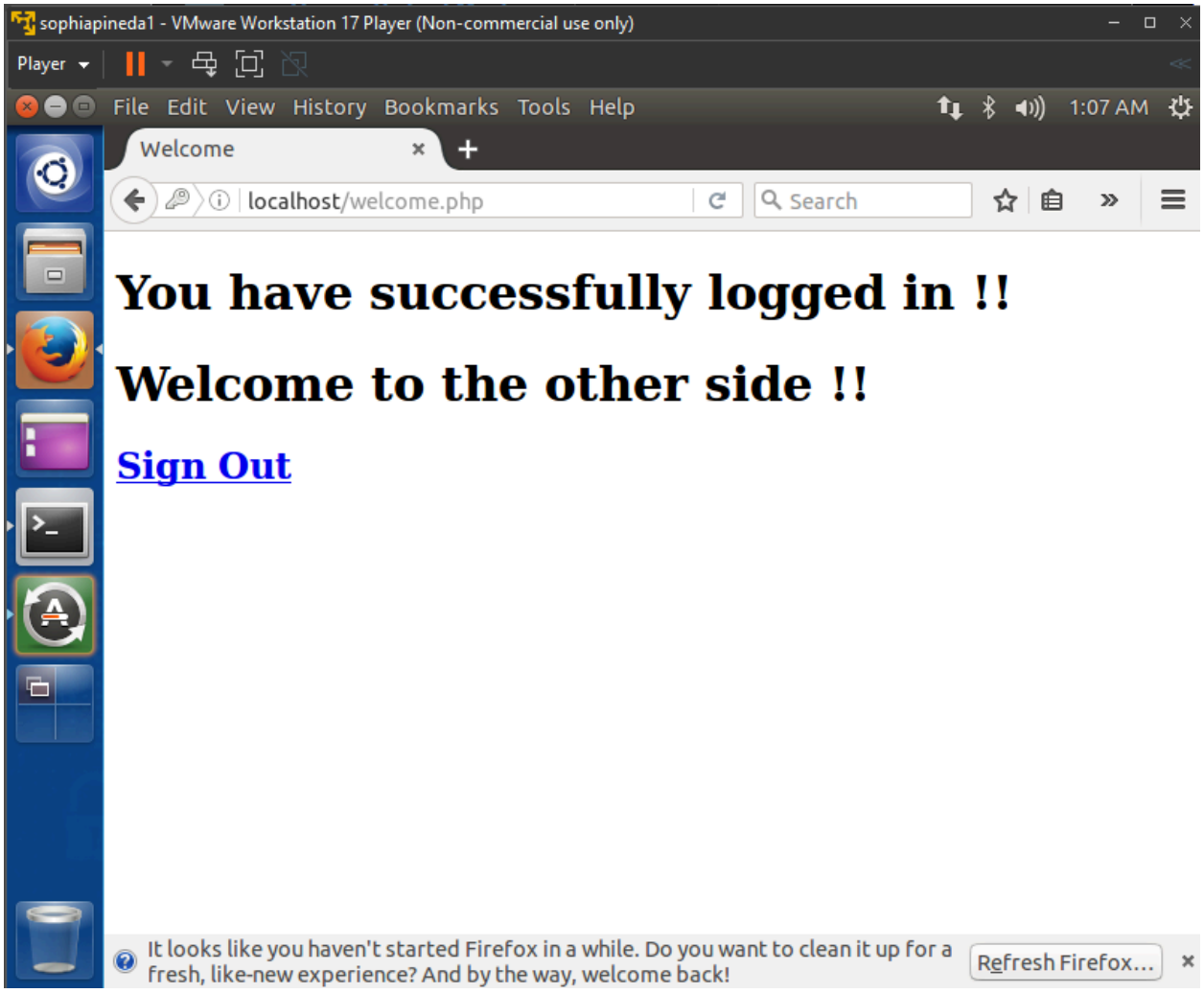
TASK 5: SQL INJECTION

STEPS:

a. Open localhost
b. Type in ‘ in the user
c. Type in nano /var/www/sqlinjection/index.php

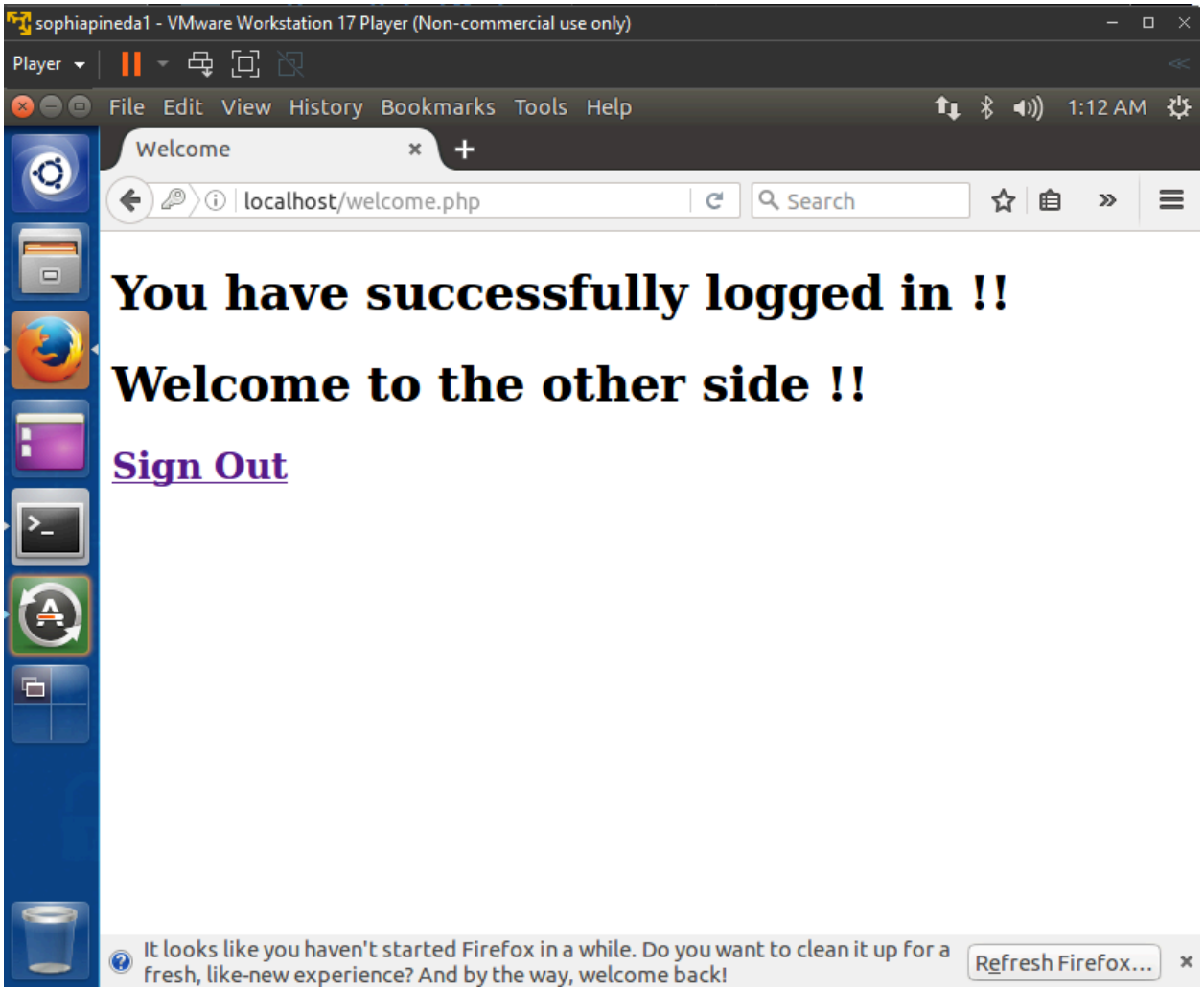
QUESTIONS https://www.invicti.com/blog/web-security/sql-injection-cheat-sheet/

1. Login user “123456789” password as an SQL command to gain unauthorised access



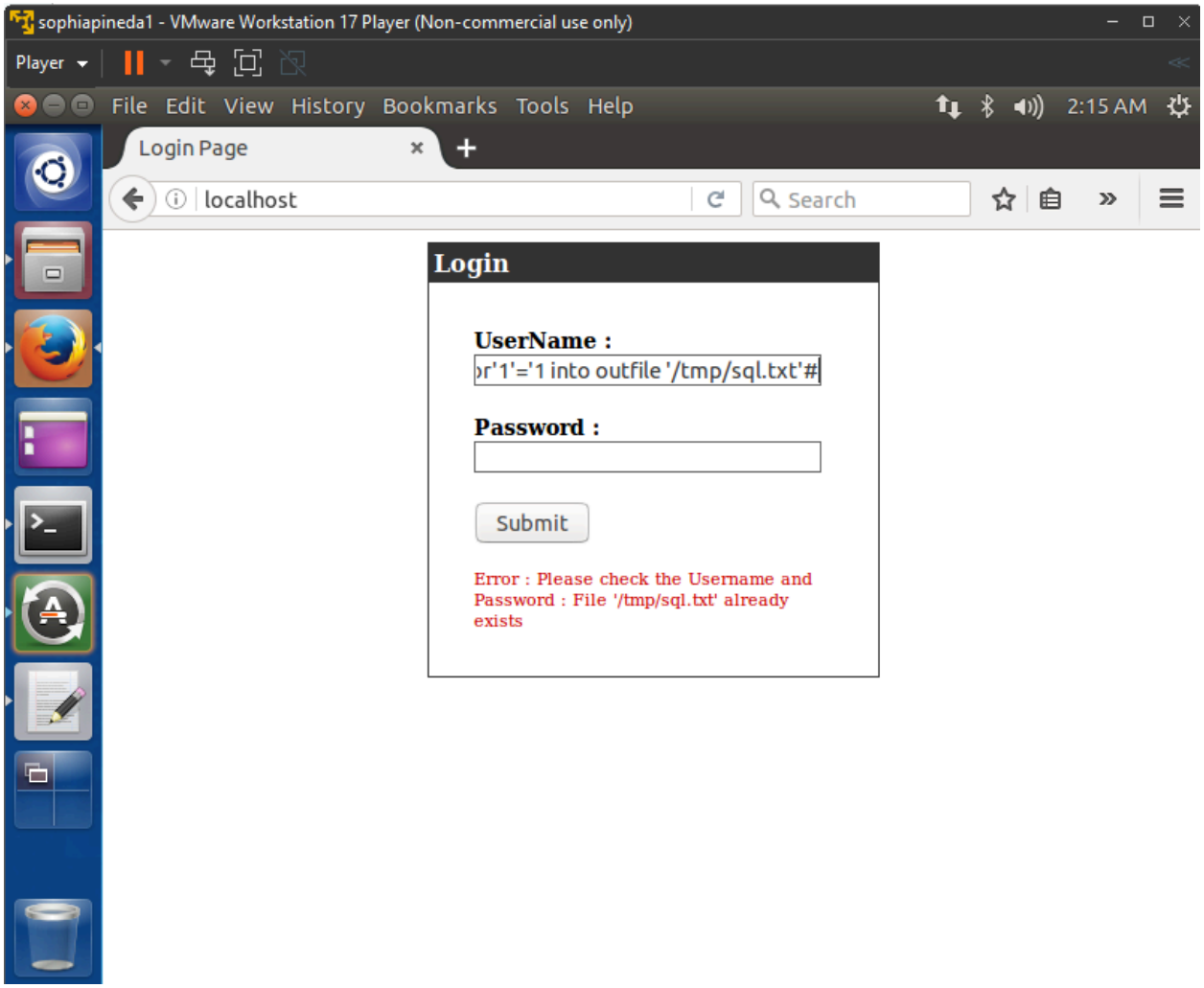
- Typing in ‘or’ 1’=’1 in the password field- means that values are always true, bypassing the validation of the password field, enabling access

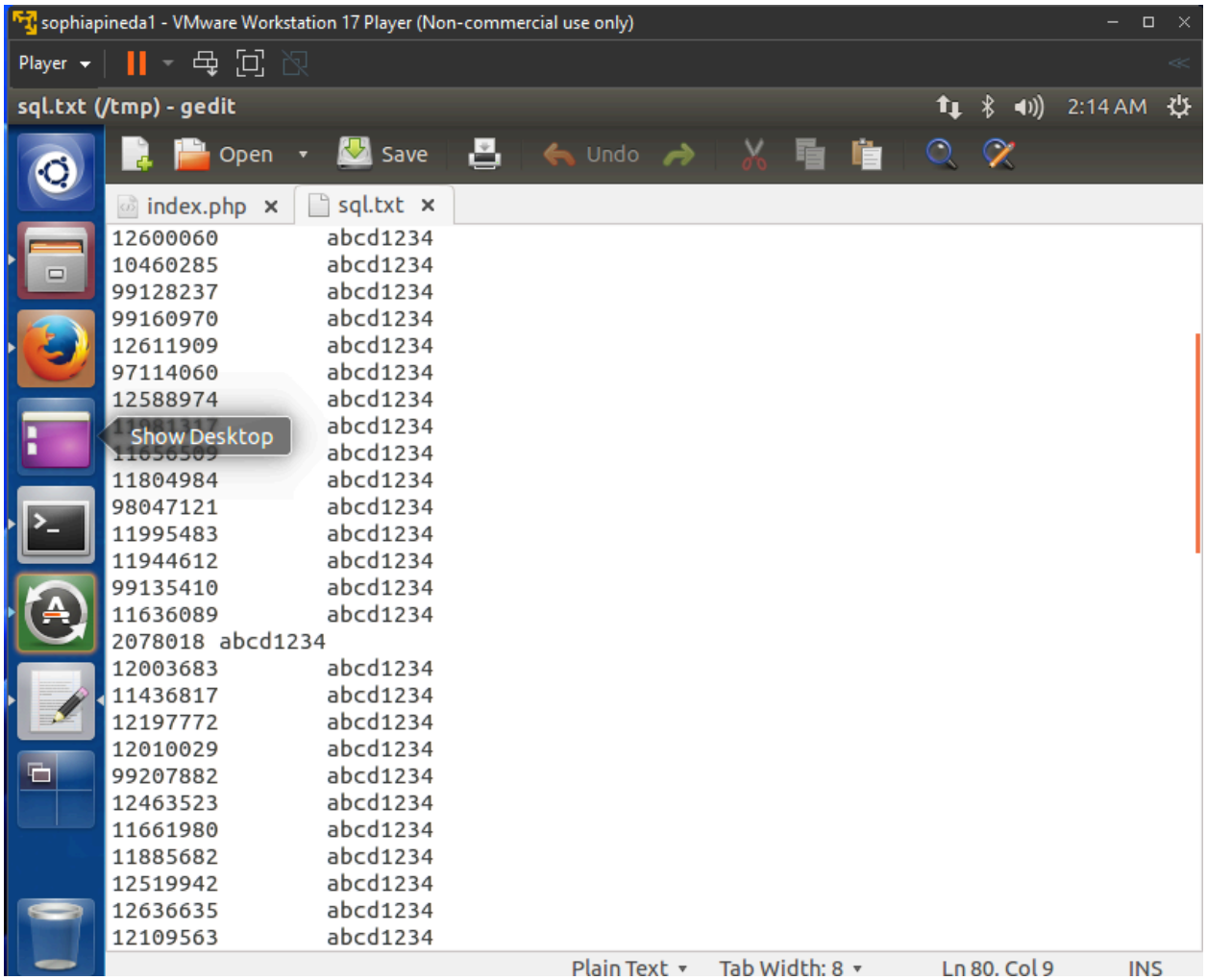
2. Login with both username and password as SQL commands



- Typing in ‘or’ 1’=’1 in both fields (password + user)- means the values are always true, bypassing the validation

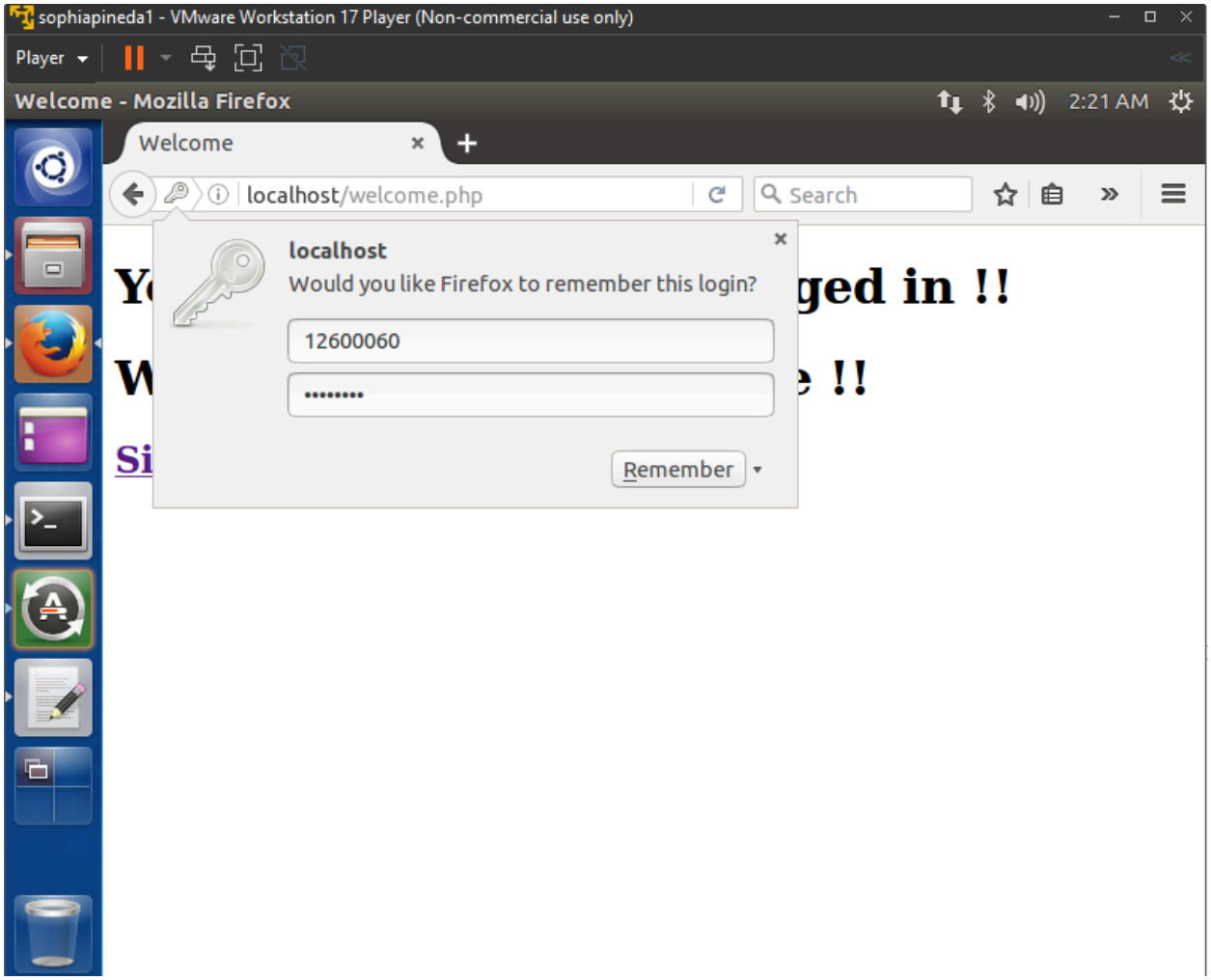
3. Find table details containing all the usernames and passwords through SQL injection


sophiapineda1 - VMware Workstation 17 Player (Non-commercial use only)
Player
File Edit View History Bookmarks Tools Help
2:15 AM
Login Page
localhost
Search
Login
UserName :
or'1'='1 into outfile '/tmp/sql.txt'#
Password :
Submit
Error : Please check the Username and Password : File '/tmp/sql.txt' already exists



- Open terminal, type in gedit /var/www/sqlinjection/index.php
- Typing in ‘or’ 1’=’1 into outfile ‘/tmp/sql.txt’# in the user, while leaving the password empty
- Going in files and opening the text to see the usernames and passwords

4. Login into a specific user account by extracting the username and password from the table



- Left hand column: password
- Right hand column: user

SUGGESTED DEFENCE (MITIGATION)

- Add input validation: clearly defining the data requirements
- Parameterized queries: separates user input from the query
- Developer of the web page must sanitise all input: removing any unsafe characters

‘or’1’=’1’ into outfile ‘/tmp/sql.txt’#

Cat /tmp/sql.txt