**Phishing Simulation**

**Table of contents**

[1. Lab Objective 3](#_Toc21709)

[2. Tools 3](#_Toc4553)

[3. Methodology 3](#_Toc32239)

[4. Simulation Steps 4](#_Toc26401)

[4.1. Pyphisher Simulation 4](#_Toc7627)

[4.2. GoPhish Simulation (Campaign) 5](#_Toc29052)

[5. Conclusion 9](#_Toc25615)

**List of Figures**

[Figure 4.1 Shows pyphisher tool 4](#_Toc23112)

[Figure 4.2 Shows phishing link to be sent to the victim 4](#_Toc28037)

[Figure 4.3 Shows gophisher sending profile (used Google mail ) 5](#_Toc18082)

[Figure 4.4 Shows gophisher Landing pages 5](#_Toc22527)

[Figure 4.5 Shows gophisher email template profile 6](#_Toc3641)

[Figure 4.6 Shows gophisher campaign page 6](#_Toc11083)

[Figure 4.7 Shows phishing mail successfully sent to the mail 7](#_Toc22724)

[Figure 4.8 Shows phishing link being opened in Windows VM(Victim VM) 7](#_Toc13946)

[Figure 4.9 Shows login credentials being captured in py-phisher 8](#_Toc20067)

[Figure 4.10 Shows OTP captured in py-phisher and redirection to genuine site 8](#_Toc2363)

**List of Tables**

[Table 2.1 Shows Tools 3](#_Toc10926)

# Lab Objective

* Simulate phishing attacks in a safe, isolated lab environment.
* Assess the ability of target VM to interact with phishing pages.
* Test the Windows unified monitoring scripts for detecting suspicious activity.
* Capture simulated credential attempts and log them in a structured format.
* Compare hands-on phishing (Py Phisher) and campaign-style phishing (Go-phish) techniques.

# Tools

| **Tool** | **Purpose / Use** |
| --- | --- |
| Py-phisher | Generate and host a harmless phishing page for lab VM |
| GoPhish | Create campaign-style phishing simulation (email + link) |
| Kali Linux | Attacker VM to host phishing simulations |
| Windows 10 | Target VM for interaction; |

##### Table 2.1 Shows Tools

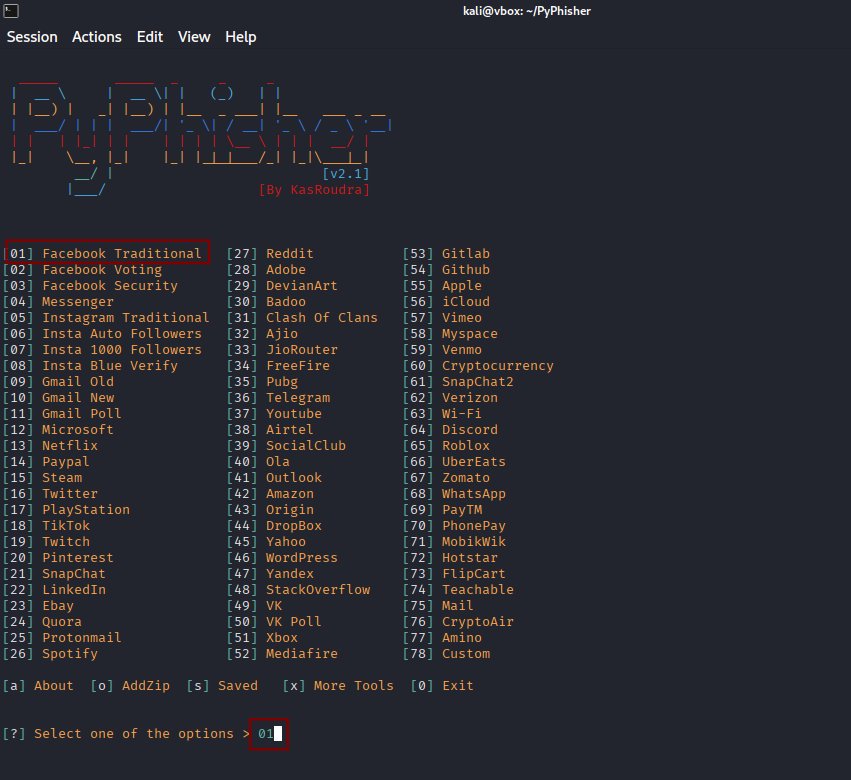
# Methodology

* Set up attacker and target VMs in a controlled lab environment.
* Attacker VM: Kali Linux (***IP: 192.168.1.43***)
* Target VM: Windows 10 (***IP: 192.168.1.53***)
* Configure Py-phisher to host a cloned login page and generate phishing links.
* Optionally configure Go-Phish campaigns for simulated email delivery within the lab VM network.
* Target VM interacts with phishing links

# **Simulation Steps**

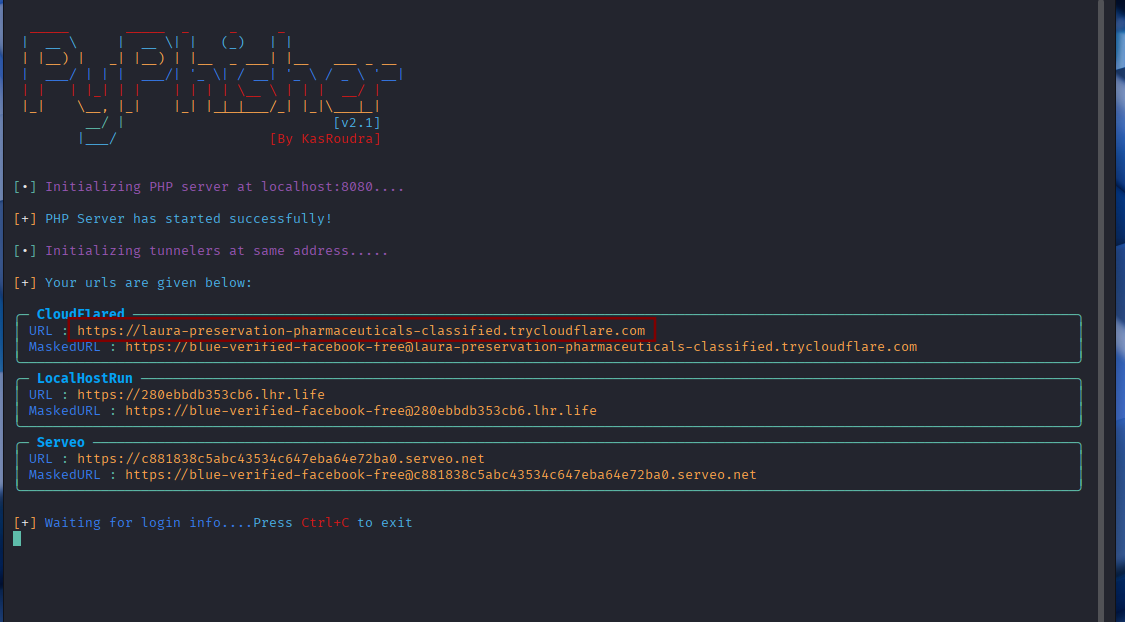
## Py-phisher Simulation

* Clone Py-phisher repository and launch the tool
* Select a login page template (e.g.,facebook).



#### Figure 4.1 Shows py-phisher tool

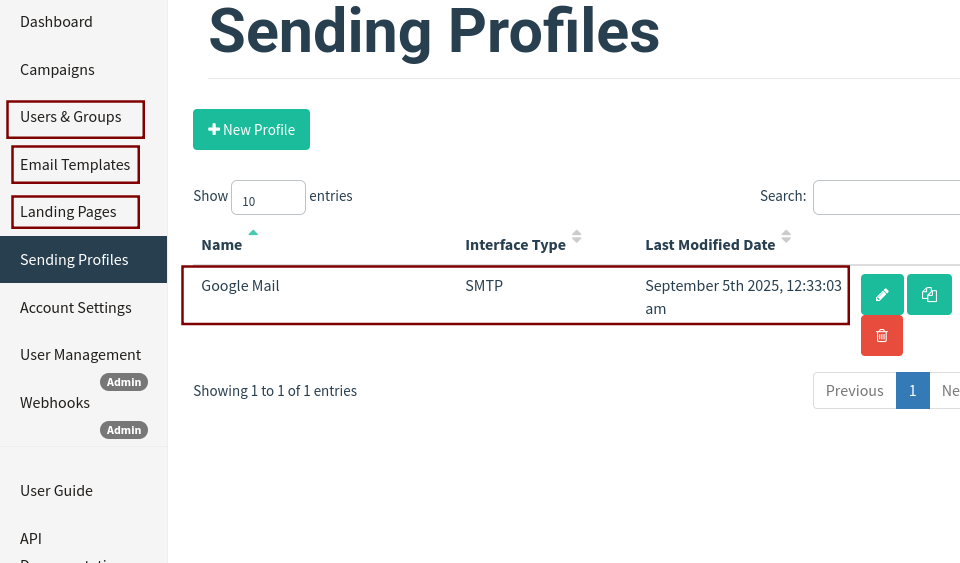
* Py-phisher generates a phishing link



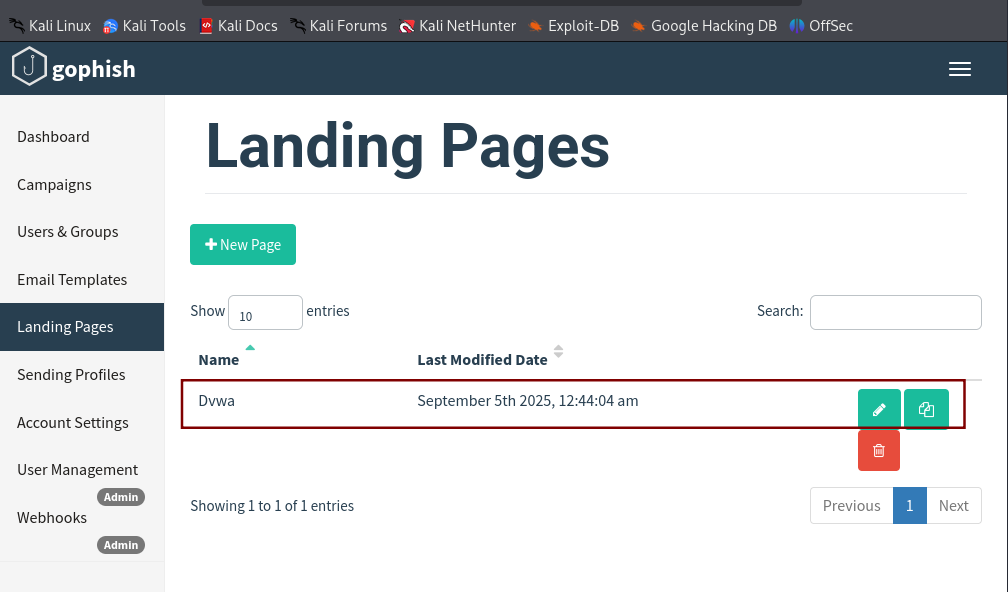
#### Figure 4.2 Shows phishing link to be sent to the victim

## **Go-phish Simulation (Campaign)**

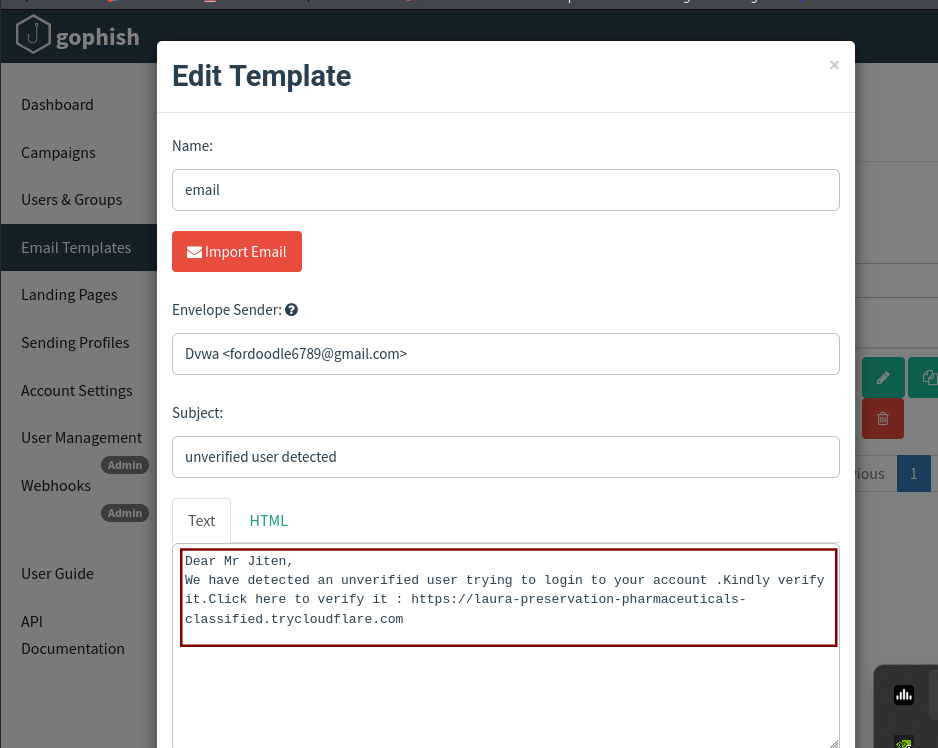
* After noting down the link provided by py-phisher ,send the link to target VM (Windows VM) through Go-phish
* Access admin interface of go-phish at : <https://127.0.0.1:3333>
* Start making profiles for sending profiles,landing pages,email templates,users and groups and finally start the campaign.



#### Figure 4.3 Shows go-phisher sending profile (used Google mail )

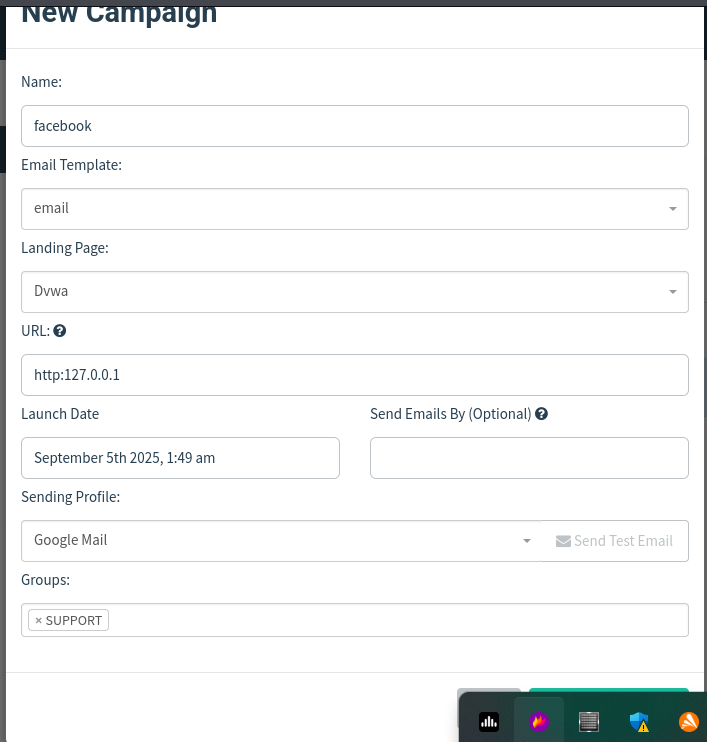


#### Figure 4.4 Shows go-phisher Landing pages



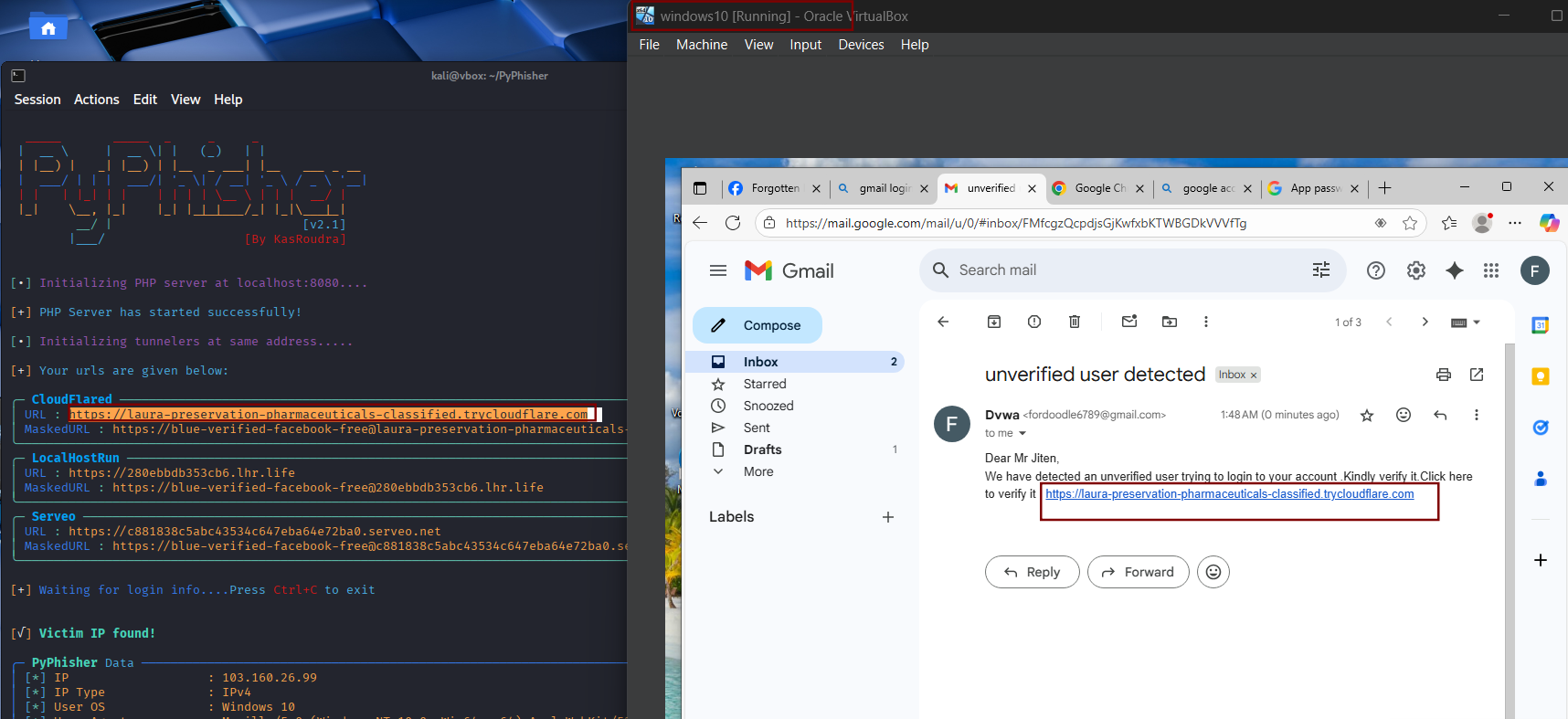
#### Figure 4.5 Shows go-phisher email template profile

* Created a phishing campaign with target VM



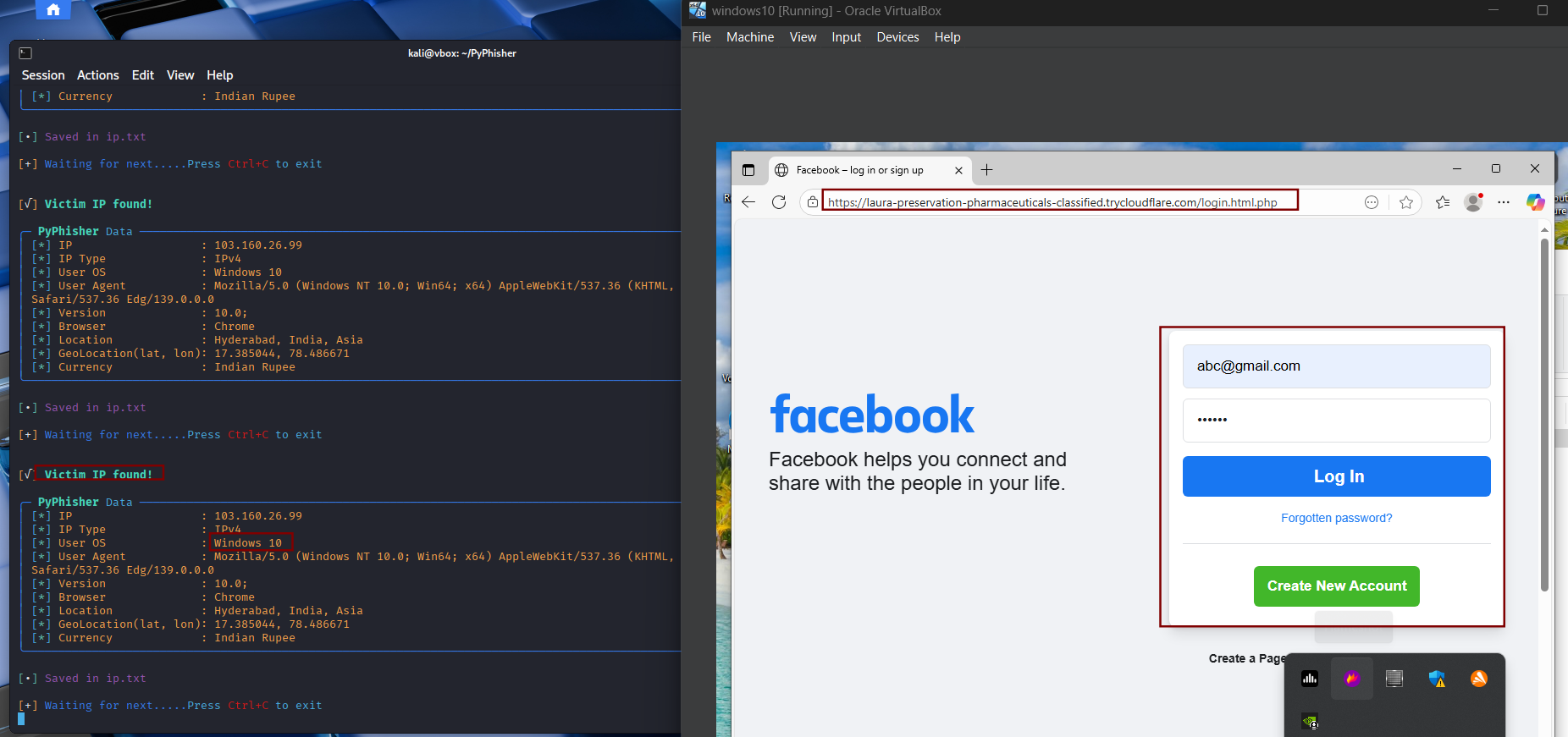
#### Figure 4.6 Shows go-phisher campaign page

* Once the campaign starts, at a given time it starts sending messages to the provided gmail as shown below



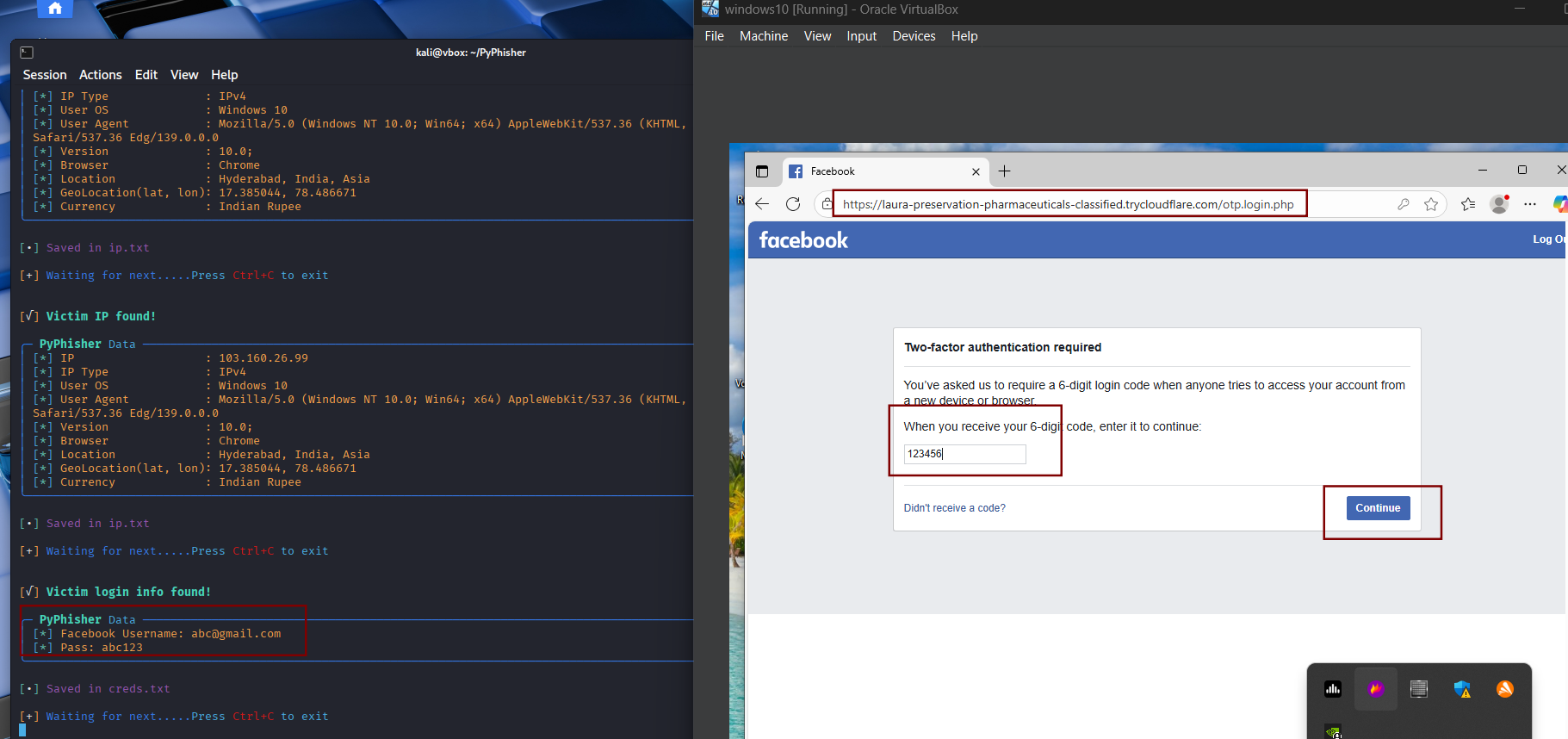
#### Figure 4.7 Shows phishing mail successfully sent to the mail

* Target VM opens the link (harmless).



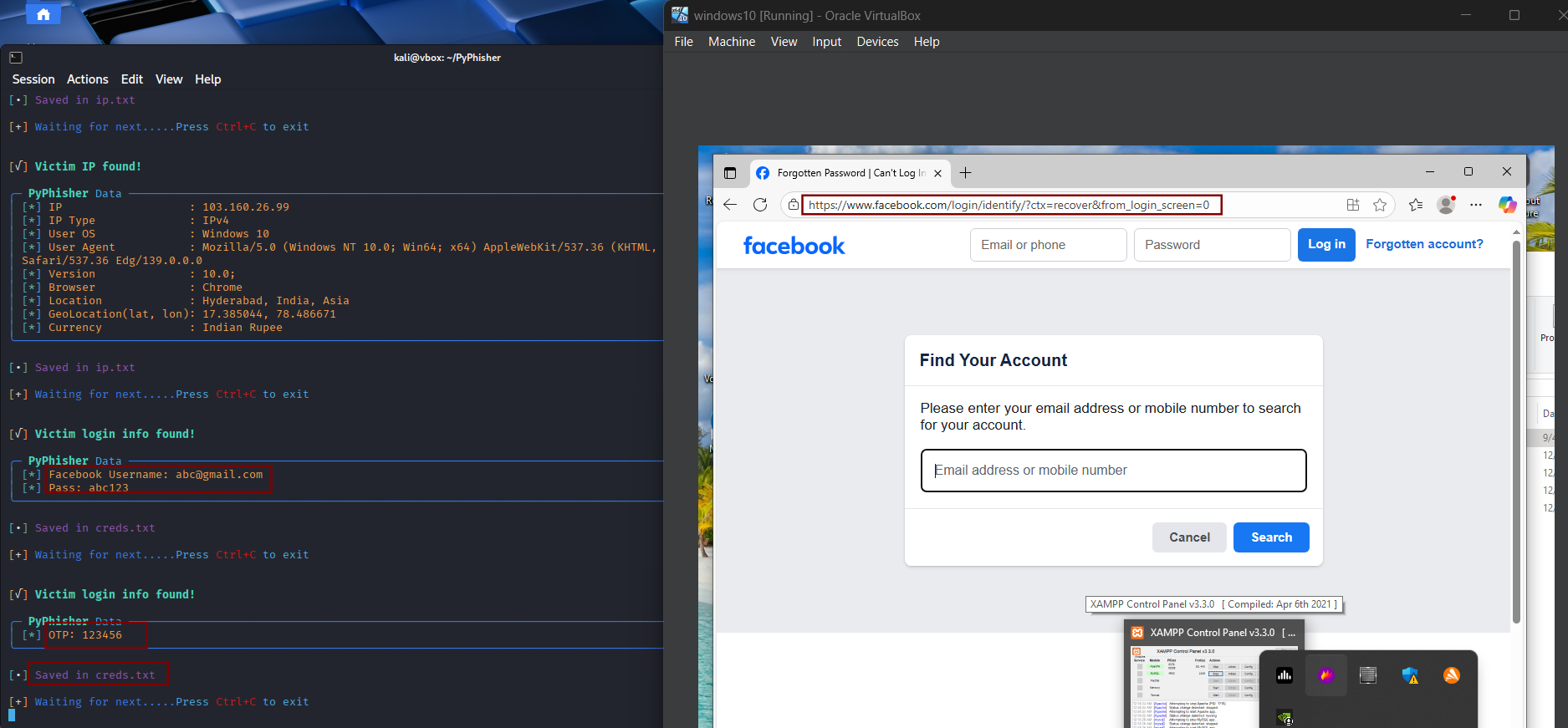
#### Figure 4.8 Shows phishing link being opened in Windows VM(Victim VM)

* Now target starts typing their email and password ,followed by OTP which is seamlessly captured in py-phisher as ***gmail: [abc@gmail.com](mailto:abc@gmail.com) and password as abc123 and are saved in creds.txt*** ,as shown below *Figure 4.9 and 4.10*



#### Figure 4.9 Shows login credentials being captured in py-phisher

* Now after the OTP is captured ,the user is then redirected to the genuine website where, he is again prompted to login.



#### Figure 4.10 Shows OTP captured in py-phisher and redirection to genuine site

# **Conclusion**

* Simulation using Py-phisher demonstrated hands-on phishing page creation and interaction.
* Go-phish campaign-style simulation showed email-based attacks in a lab-controlled network.
* No real credentials or external targets were used.