



CYART

[inquiry@cyart.io](mailto:inquiry@cyart.io)

[www.cyart.io](http://www.cyart.io)

---

## **Phishing Simulation**



---

## Table of contents

1. Lab Objective	3
2. Tools	3
3. Methodology	3
4. Simulation Steps	4
4.1. Pyphisher Simulation	4
4.2. GoPhish Simulation (Campaign)	5
5. Conclusion	9

## List of Figures

Figure 4.1 Shows pyphisher tool	4
Figure 4.2 Shows phishing link to be sent to the victim	4
Figure 4.3 Shows gophisher sending profile (used Google mail )	5
Figure 4.4 Shows gophisher Landing pages	5
Figure 4.5 Shows gophisher email template profile	6
Figure 4.6 Shows gophisher campaign page	6
Figure 4.7 Shows phishing mail successfully sent to the mail	7
Figure 4.8 Shows phishing link being opened in Windows VM(Victim VM)	7
Figure 4.9 Shows login credentials being captured in py-phisher	8
Figure 4.10 Shows OTP captured in py-phisher and redirection to genuine site	8

## List of Tables

Table 2.1 Shows Tools	3
-----------------------	---



## 1. 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.

## 2. 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*

## 3. 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

## 4. Simulation Steps

### 4.1. Py-phisher Simulation

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



```

kali@vbox: ~/PyPhisher
Session Actions Edit View Help

[?] PyPhisher [v2.1] [By KasRoudra]

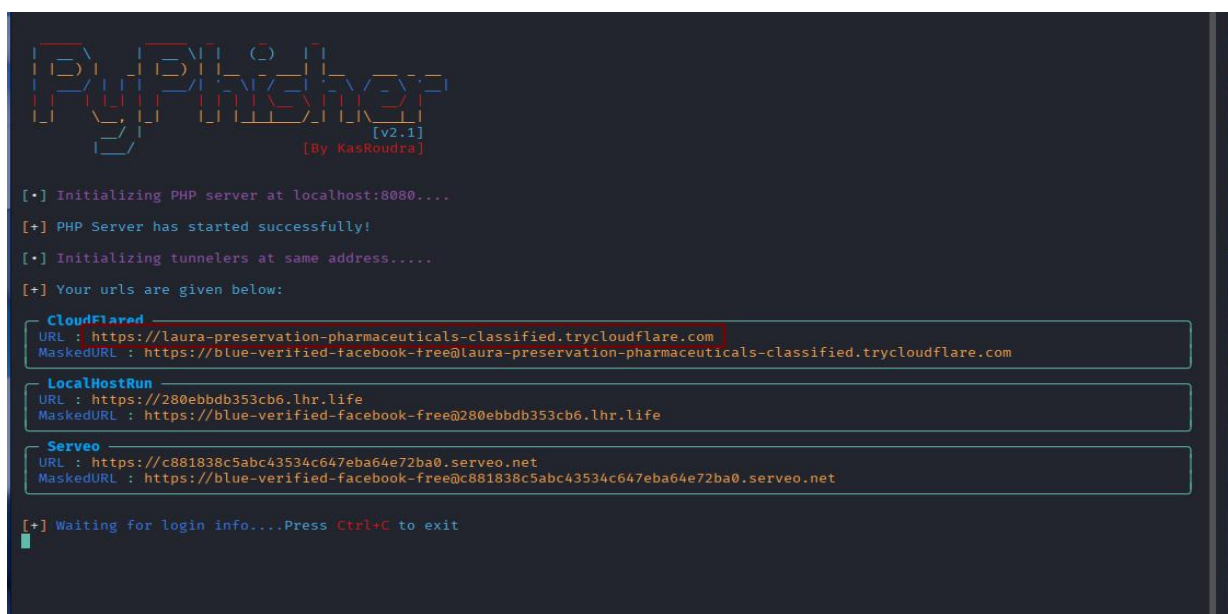
[01] Facebook Traditional [27] Reddit [53] Gitlab
[02] Facebook Voting [28] Adobe [54] Github
[03] Facebook Security [29] DevianArt [55] Apple
[04] Messenger [30] Badoo [56] iCloud
[05] Instagram Traditional [31] Clash Of Clans [57] Vimeo
[06] Insta Auto Followers [32] Ajio [58] Myspace
[07] Insta 1000 Followers [33] JioRouter [59] Venmo
[08] Insta Blue Verify [34] FreeFire [60] Cryptocurrency
[09] Gmail Old [35] Pubg [61] SnapChat2
[10] Gmail New [36] Telegram [62] Verizon
[11] Gmail Poll [37] Youtube [63] Wi-Fi
[12] Microsoft [38] Airtel [64] Discord
[13] Netflix [39] SocialClub [65] Roblox
[14] Paypal [40] Ola [66] UberEats
[15] Steam [41] Outlook [67] Zomato
[16] Twitter [42] Amazon [68] WhatsApp
[17] PlayStation [43] Origin [69] PayTM
[18] TikTok [44] DropBox [70] PhonePay
[19] Twitch [45] Yahoo [71] MobikWik
[20] Pinterest [46] WordPress [72] Hotstar
[21] Snapchat [47] Yandex [73] FlipCart
[22] LinkedIn [48] StackOverflow [74] Teachable
[23] Ebay [49] VK [75] Mail
[24] Quora [50] VK Poll [76] CryptoAir
[25] Protonmail [51] Xbox [77] Amino
[26] Spotify [52] Mediafire [78] Custom

[a] About [o] AddZip [s] Saved [x] More Tools [0] Exit

[?] Select one of the options > 01
  
```

Figure 4.1 Shows py-phisher tool

- Py-phisher generates a phishing link



```

[?] PyPhisher [v2.1] [By KasRoudra]

[*] Initializing PHP server at localhost:8080....
[+] PHP Server has started successfully!
[*] Initializing tunnelers at same address.....
[+] Your urls are given below:

CloudFlare -
URL : https://laura-preservation-pharmaceuticals-classified.trycloudflare.com
MaskedURL : https://blue-verified-facebook-free@laura-preservation-pharmaceuticals-classified.trycloudflare.com

LocalHostRun
URL : https://280ebdb353cb6.lhr.life
MaskedURL : https://blue-verified-facebook-free@280ebdb353cb6.lhr.life

Serveo -
URL : https://c881838c5abc43534c647eba64e72ba0.serveo.net
MaskedURL : https://blue-verified-facebook-free@c881838c5abc43534c647eba64e72ba0.serveo.net

[+] Waiting for login info....Press Ctrl+C to exit
  
```

Figure 4.2 Shows phishing link to be sent to the victim



## 4.2. 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.

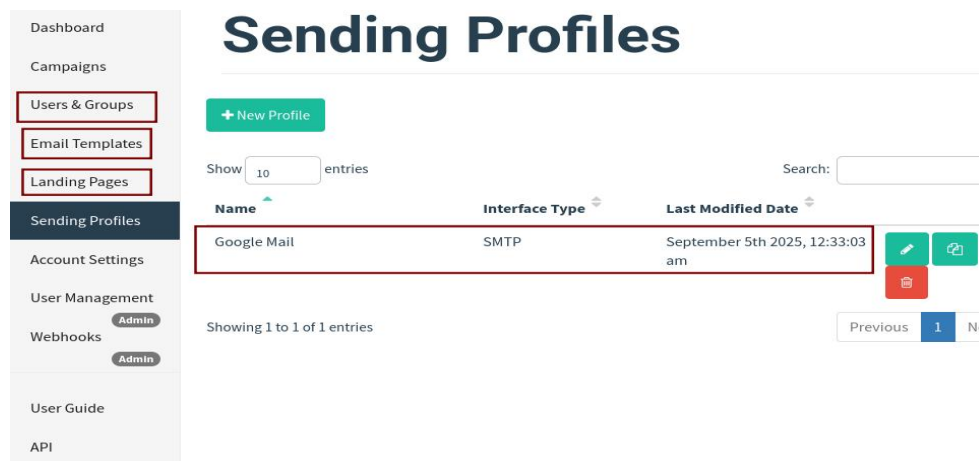


Figure 4.3 Shows go-phisher sending profile (used Google mail )

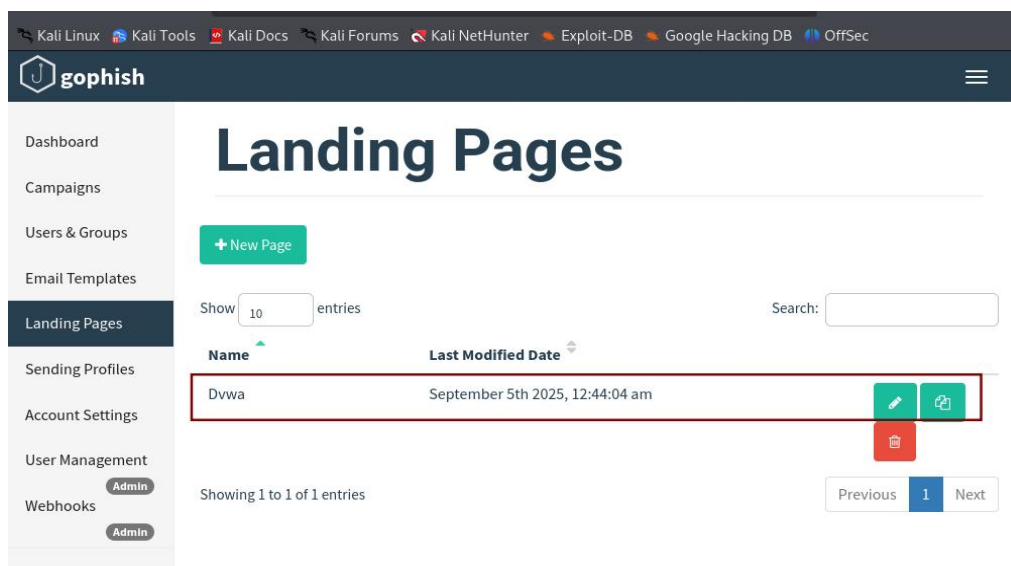


Figure 4.4 Shows go-phisher Landing pages



**go-phisher**

Dashboard  
Campaigns  
Users & Groups  
Email Templates  
Landing Pages  
Sending Profiles  
Account Settings  
User Management  
Webhooks  
User Guide  
API  
Documentation

### Edit Template

Name:

[Import Email](#)

Envelope Sender:

Subject:

Text **HTML**

Dear Mr Jiten,  
We have detected an unverified user trying to login to your account .Kindly verify it.Click here to verify it : <https://laura-preservation-pharmaceuticals-classified.trycloudflare.com>

Figure 4.5 Shows go-phisher email template profile

- Created a phishing campaign with target VM

### New Campaign

Name:

Email Template:

Landing Page:

URL:

Launch Date:

Send Emails By (Optional):

Sending Profile:  [Send Test Email](#)

Groups:

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

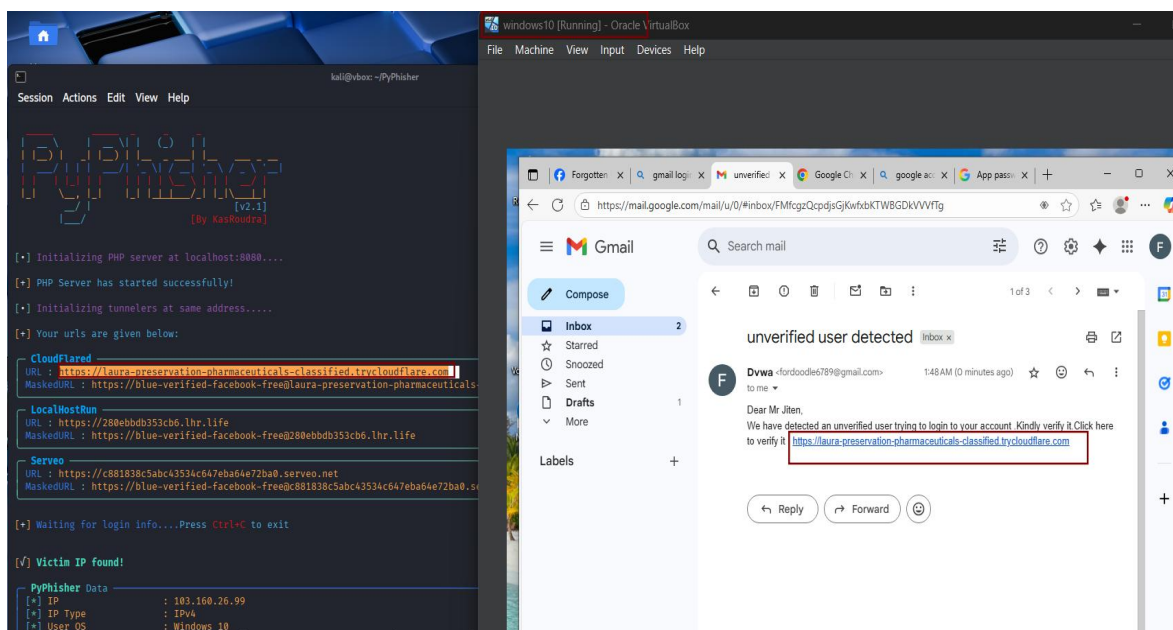


Figure 4.7 Shows phishing mail successfully sent to the mail

- Target VM opens the link (harmless).

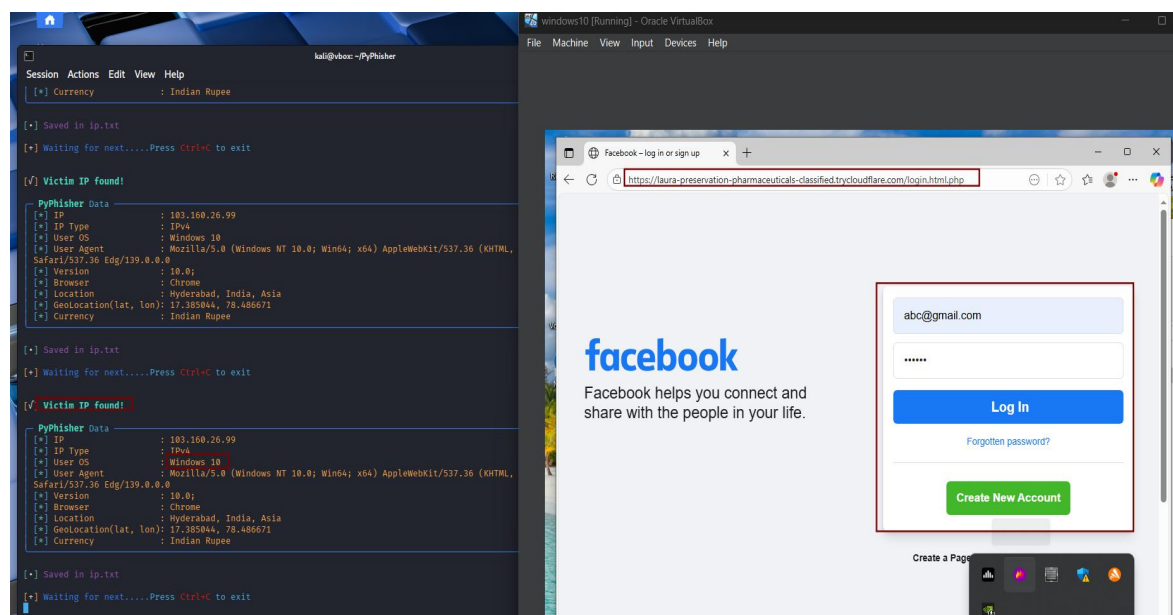
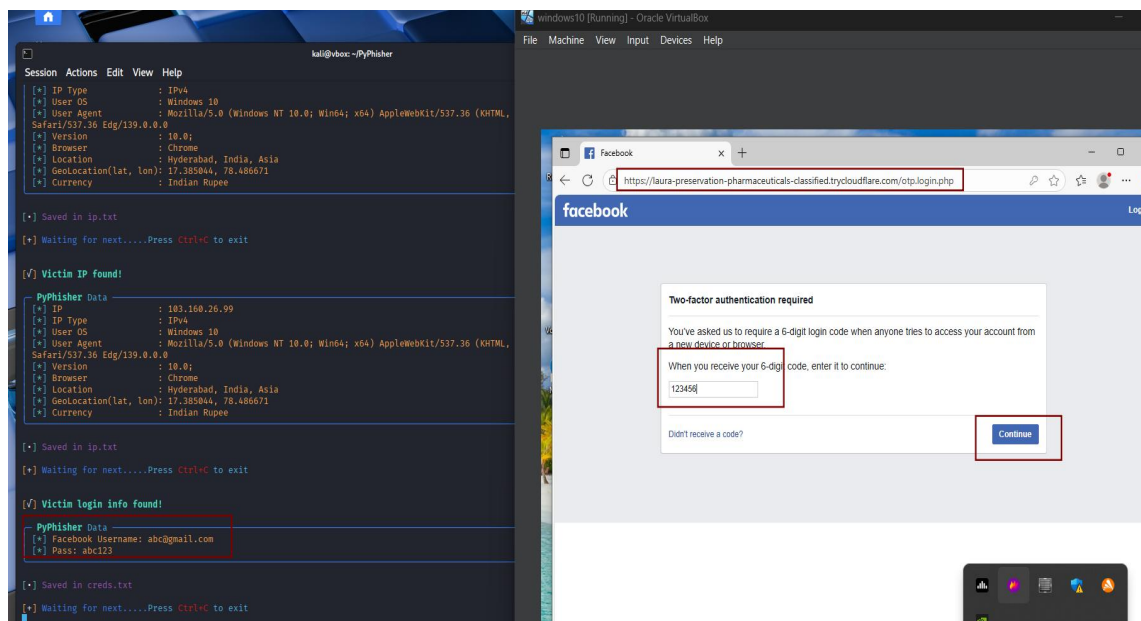


Figure 4.8 Shows phishing link being opened in Windows VM (Victim VM)



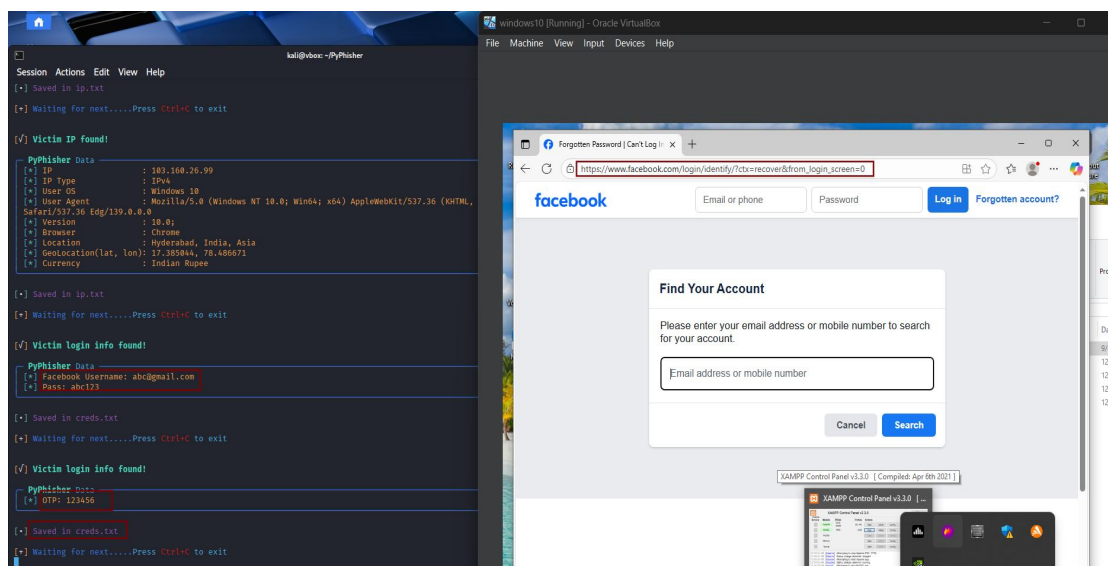


- Now target starts typing their email and password ,followed by OTP which is seamlessly captured in py-phisher as **gmail: [abc@gmail.com](mailto:abc@gmail.com) and password as abc123 and are saved in creds.txt** ,as shown below *Figure 4.9 and 4.10*



*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*





## 5. 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.