Practical 2: Phishing Simulation Using Gophish and Evilginx2

Aim

The aim of this practical is to study, install, and attempt the execution of phishing simulation tools **Gophish** and **Evilginx2** in a controlled laboratory environment, understand their working mechanisms, and analyze the practical challenges encountered during real-world phishing framework deployment.

Environment Setup

- Operating System: Kali Linux (Virtual Machine)
- User Privileges: Root / sudo
- Network Mode: NAT (Lab environment)
- Internet Connectivity: Enabled

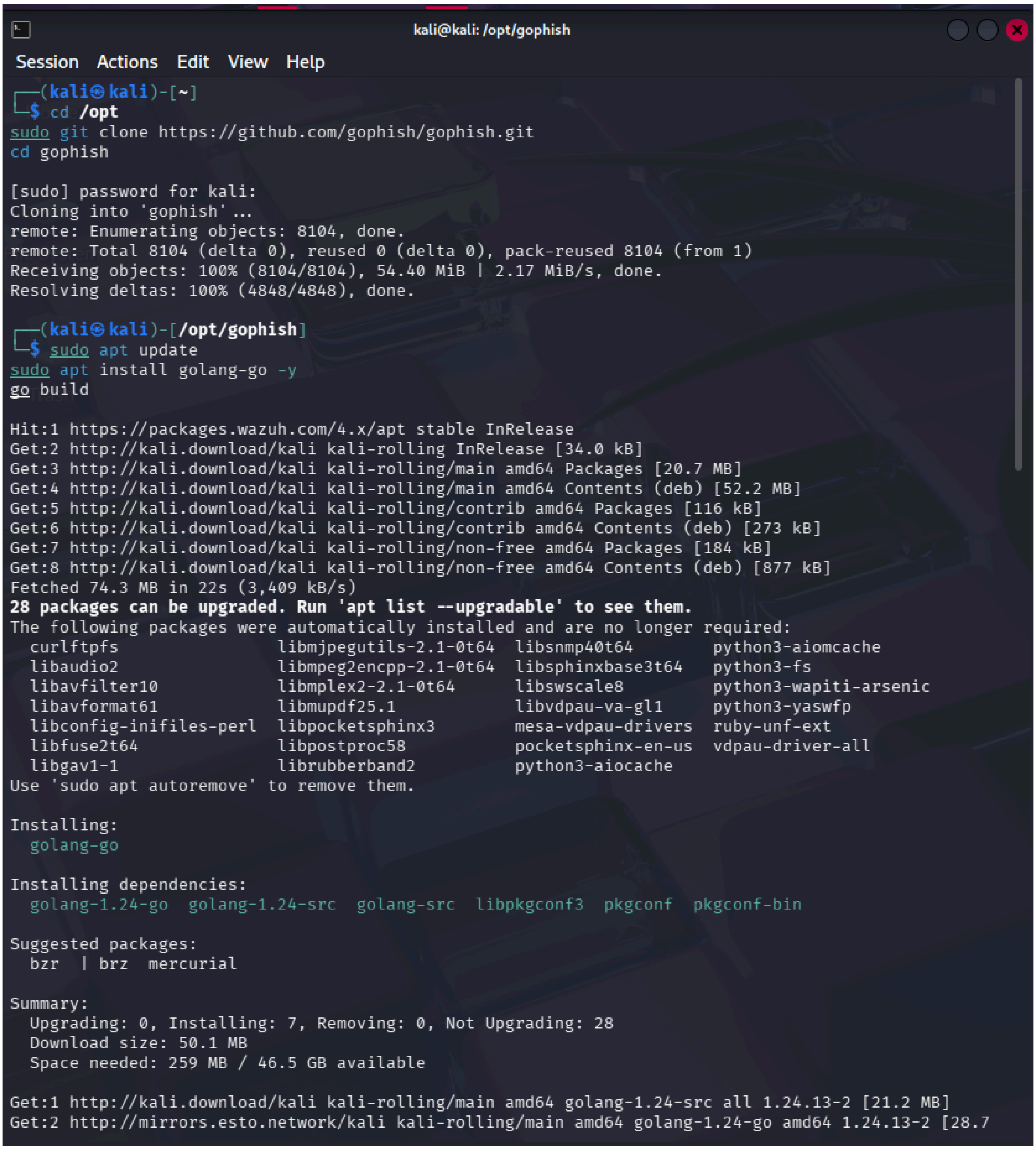
Tools Used

- Gophish (Open-source phishing simulation framework)
- Evilginx2 (Advanced phishing framework)
- Go Programming Language
- Git version control system

Part A: Gophish (Attempted – Execution Partially Successful)

Objective

To install Gophish, configure a phishing campaign, and observe campaign execution behavior.



```
┌──(kali㉿kali)-[~]
└─$ cd /opt
sudo git clone https://github.com/gophish/gophish.git
cd gophish

[sudo] password for kali:
Cloning into 'gophish'...
remote: Enumerating objects: 8104, done.
remote: Total 8104 (delta 0), reused 0 (delta 0), pack-reused 8104 (from 1)
Receiving objects: 100% (8104/8104), 54.40 MiB | 2.17 MiB/s, done.
Resolving deltas: 100% (4848/4848), done.

┌──(kali㉿kali)-[/opt/gophish]
└─$ sudo apt update
sudo apt install golang-go -y
go build

Hit:1 https://packages.wazuh.com/4.x/apt stable InRelease
Get:2 http://kali.download/kali kali-rolling InRelease [34.0 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Packages [20.7 MB]
Get:4 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [52.2 MB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Packages [116 kB]
Get:6 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [273 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Packages [184 kB]
Get:8 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [877 kB]
Fetched 74.3 MB in 22s (3,409 kB/s)
28 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
  curlftpfs                libmjpegutils-2.1-0t64  libsnmp40t64        python3-aiomcache
  libaudio2                libmpeg2encpp-2.1-0t64  libsphinxbase3t64   python3-fs
  libavfilter10            libmplex2-2.1-0t64      libswscale8         python3-wapiti-arsenic
  libavformat61            libmupdf25.1            libvdpau-va-gl1     python3-yaswfp
  libconfig-inifiles-perl  libpocketsphinx3        mesa-vdpau-drivers  ruby-unf-ext
  libfuse2t64              libpostproc58           pocketsphinx-en-us  vdpau-driver-all
  libgav1-1                librubberband2          python3-aiocache
Use 'sudo apt autoremove' to remove them.

Installing:
  golang-go

Installing dependencies:
  golang-1.24-go  golang-1.24-src  golang-src  libpkgconf3  pkgconf  pkgconf-bin

Suggested packages:
  bzr  | brz  mercurial

Summary:
  Upgrading: 0, Installing: 7, Removing: 0, Not Upgrading: 28
  Download size: 50.1 MB
  Space needed: 259 MB / 46.5 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 golang-1.24-src all 1.24.13-2 [21.2 MB]
Get:2 http://mirrors.esto.network/kali kali-rolling/main amd64 golang-1.24-go amd64 1.24.13-2 [28.7
```

Step 1: Cloning the Gophish Repository

```
cd /opt
sudo git clone https://github.com/gophish/gophish.git
cd gophish
```

The repository was cloned successfully into the `/opt` directory, confirming proper internet access and Git configuration.

Step 2: Installing Dependencies and Building Gophish

```
sudo apt update
sudo apt install golang-go -y
go build
```

Problem 1: VCS Stamping Error During Build

Error Observed

```
error obtaining VCS status: exit status 128
Use -buildvcs=false to disable VCS stamping.
```

Cause:
Recent Go versions enforce Version Control System (VCS) stamping during builds. Since the cloned repository did not contain expected metadata, the build failed.

Solution Applied

- `go build -buildvcs=false`

This disabled VCS stamping and allowed the build process to continue.

Problem 2: Permission Denied While Writing Binary

Error Observed

```
copying ... open gophish: permission denied
```

Cause:

The `/opt` directory requires root privileges to create executable files.

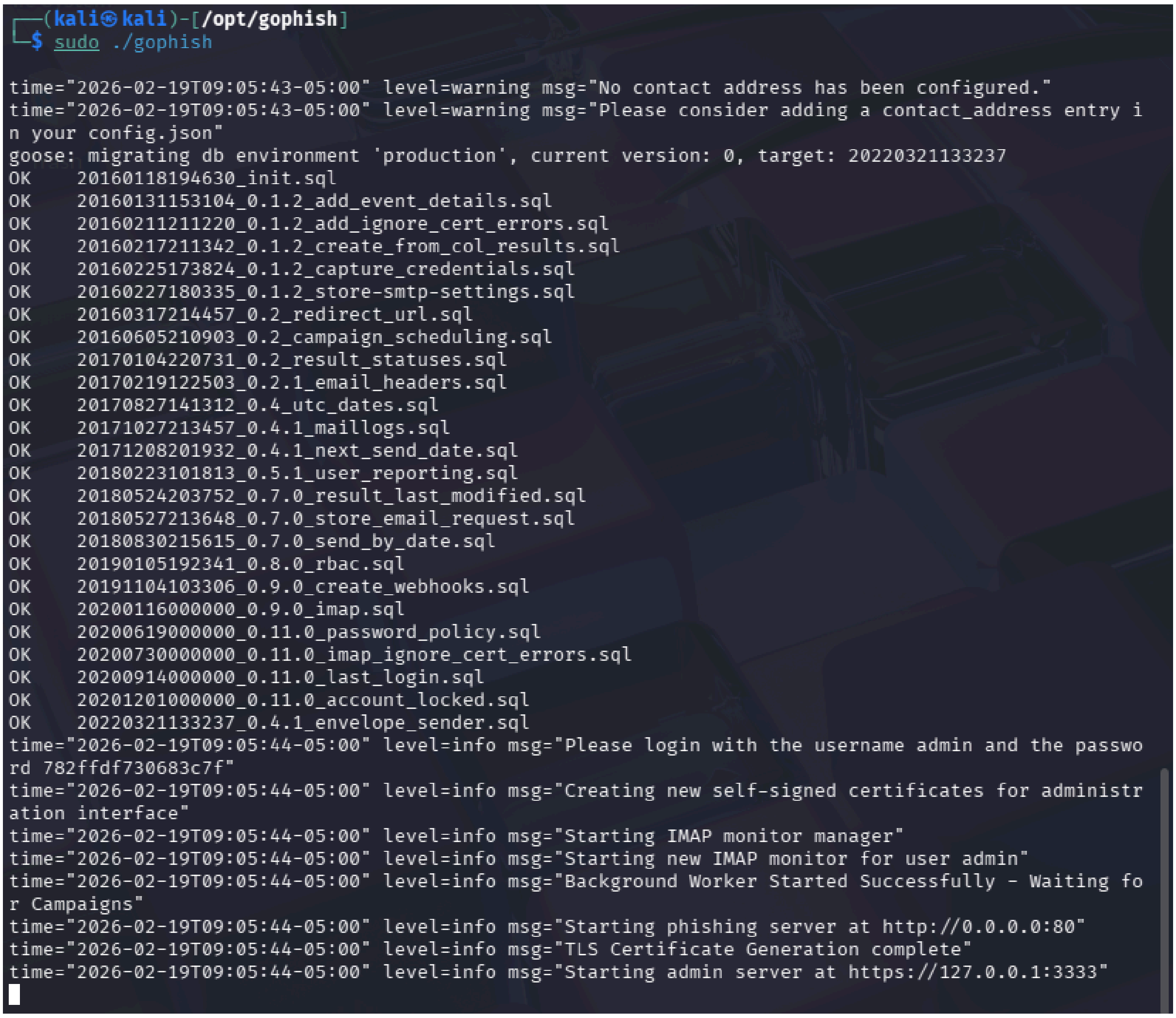
Solution Applied

```
sudo go build -buildvcs=false
```

After executing the command with elevated privileges, the Gophish binary was successfully generated.

Step 3: Launching Gophish

- `sudo ./gophish`



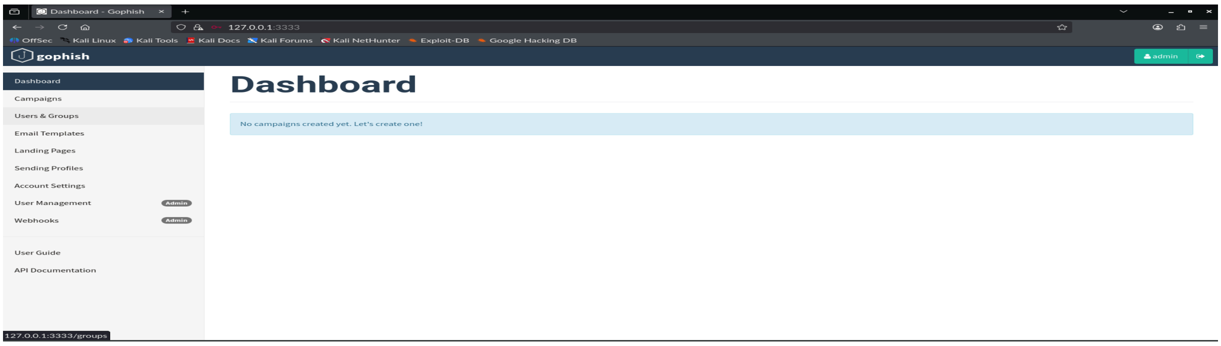
```
┌──(kali㉿kali)-[/opt/gophish]
└─$ sudo ./gophish

time="2026-02-19T09:05:43-05:00" level=warning msg="No contact address has been configured."
time="2026-02-19T09:05:43-05:00" level=warning msg="Please consider adding a contact_address entry i
n your config.json"
goose: migrating db environment 'production', current version: 0, target: 20220321133237
OK    20160118194630_init.sql
OK    20160131153104_0.1.2_add_event_details.sql
OK    20160211211220_0.1.2_add_ignore_cert_errors.sql
OK    20160217211342_0.1.2_create_from_col_results.sql
OK    20160225173824_0.1.2_capture_credentials.sql
OK    20160227180335_0.1.2_store-smtp-settings.sql
OK    20160317214457_0.2_redirect_url.sql
OK    20160605210903_0.2_campaign_scheduling.sql
OK    20170104220731_0.2_result_statuses.sql
OK    20170219122503_0.2.1_email_headers.sql
OK    20170827141312_0.4_utc_dates.sql
OK    20171027213457_0.4.1_maillogs.sql
OK    20171208201932_0.4.1_next_send_date.sql
OK    20180223101813_0.5.1_user_reporting.sql
OK    20180524203752_0.7.0_result_last_modified.sql
OK    20180527213648_0.7.0_store_email_request.sql
OK    20180830215615_0.7.0_send_by_date.sql
OK    20190105192341_0.8.0_rbac.sql
OK    20191104103306_0.9.0_create_webhooks.sql
OK    20200116000000_0.9.0_imap.sql
OK    20200619000000_0.11.0_password_policy.sql
OK    20200730000000_0.11.0_imap_ignore_cert_errors.sql
OK    20200914000000_0.11.0_last_login.sql
OK    20201201000000_0.11.0_account_locked.sql
OK    20220321133237_0.4.1_envelope_sender.sql
time="2026-02-19T09:05:44-05:00" level=info msg="Please login with the username admin and the passwo
rd 782ffdf730683c7f"
time="2026-02-19T09:05:44-05:00" level=info msg="Creating new self-signed certificates for administr
ation interface"
time="2026-02-19T09:05:44-05:00" level=info msg="Starting IMAP monitor manager"
time="2026-02-19T09:05:44-05:00" level=info msg="Starting new IMAP monitor for user admin"
time="2026-02-19T09:05:44-05:00" level=info msg="Background Worker Started Successfully - Waiting fo
r Campaigns"
time="2026-02-19T09:05:44-05:00" level=info msg="Starting phishing server at http://0.0.0.0:80"
time="2026-02-19T09:05:44-05:00" level=info msg="TLS Certificate Generation complete"
time="2026-02-19T09:05:44-05:00" level=info msg="Starting admin server at https://127.0.0.1:3333"
```

Gophish started successfully, and the web interface became accessible at:

- Admin Panel: `https://127.0.0.1:3333`

- Phishing Server: `http://127.0.0.1:80`



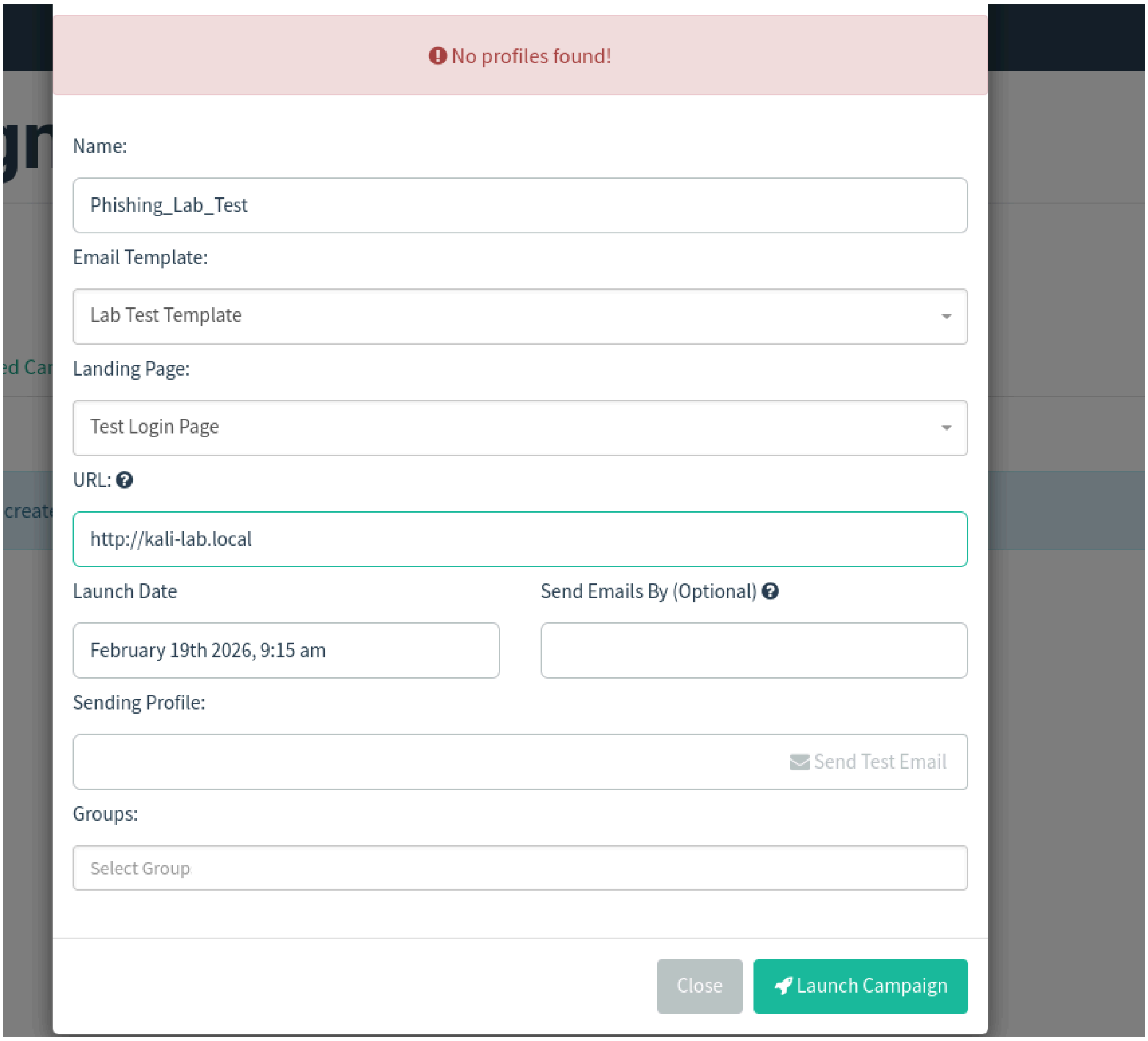
Dashboard

No campaigns created yet. Let's create one!

Step 4: Campaign Configuration Attempt

The following configurations were performed using the Gophish dashboard:

- Created an email template
- Created a landing page
- Added a test target user (`testuser@lab.local`)
- Created a campaign named **Phishing_Lab_Test**



No profiles found!

Name: Phishing_Lab_Test

Email Template: Lab Test Template

Landing Page: Test Login Page

URL: http://kali-lab.local

Launch Date: February 19th 2026, 9:15 am

Send Emails By (Optional):

Sending Profile:

Groups: Select Group

Problem 3: “No Profiles Found” Error

Observed Message

- `No profiles found!`

Cause:
Gophish requires a **Sending Profile (SMTP configuration)** to dispatch phishing emails. No SMTP profile was configured at the time of campaign launch.

Action Taken:
An SMTP profile setup was attempted, but no valid mail server credentials were available in the lab environment.

Problem 4: Campaign Status Showing “Error”

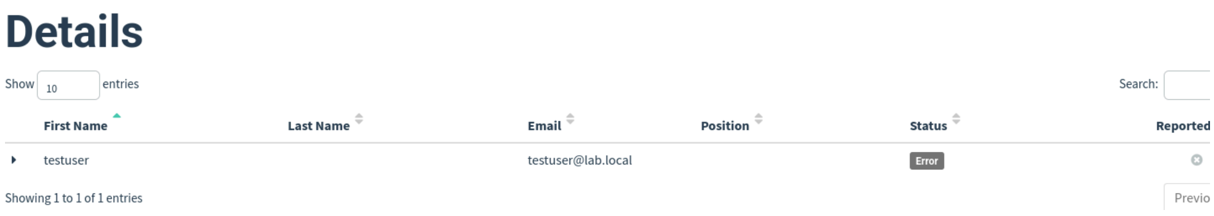
Observed Behavior

- Campaign launched
- Target user status showed **Error**
- No email delivery occurred

Cause:

- Absence of a working SMTP relay
- Local lab environment without external mail server integration

Outcome:
The phishing campaign configuration process was successfully demonstrated, but **email delivery could not be completed**, which reflects realistic infrastructure limitations during phishing simulations.



Details

Show 10 entries | Search:

	First Name	Last Name	Email	Position	Status	Reported
▸	testuser		testuser@lab.local		Error	⊗

Showing 1 to 1 of 1 entries

Previo

Part B: Evilginx2

Objective

To install Evilginx2 and verify its operational readiness in a lab environment.

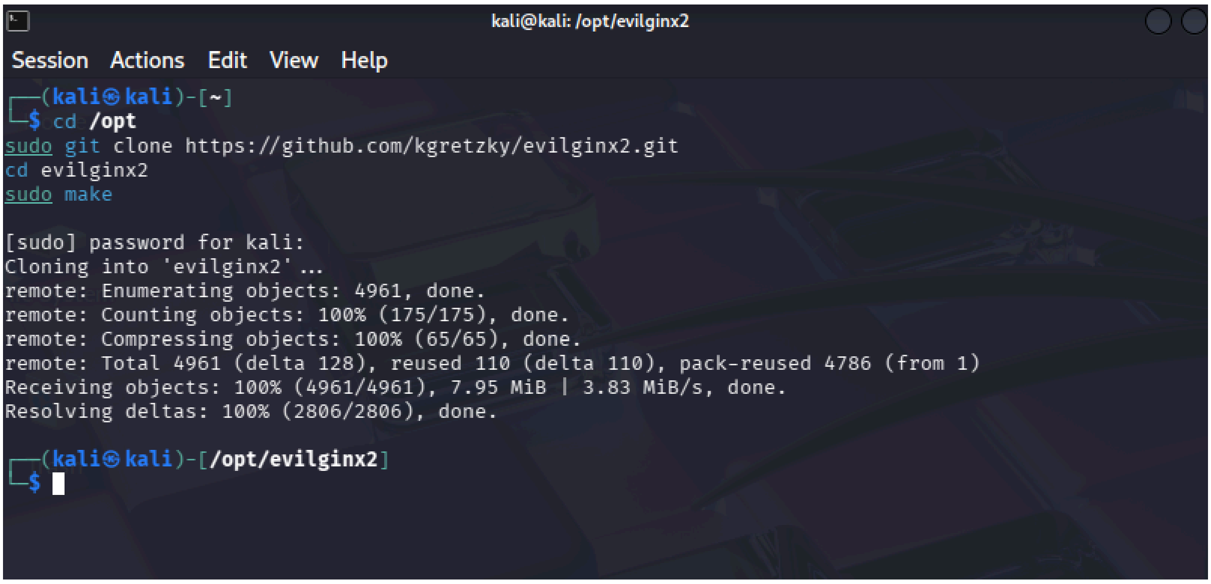
Step 1: Installation and Compilation

□ `cd /opt`

□ `sudo git clone https://github.com/kgretzky/evilginx2.git`

□ `cd evilginx2`

□ `sudo make`



```
┌──(kali㉿kali)-[~]
└─$ cd /opt
sudo git clone https://github.com/kgretzky/evilginx2.git
cd evilginx2
sudo make

[sudo] password for kali:
Cloning into 'evilginx2'...
remote: Enumerating objects: 4961, done.
remote: Counting objects: 100% (175/175), done.
remote: Compressing objects: 100% (65/65), done.
remote: Total 4961 (delta 128), reused 110 (delta 110), pack-reused 4786 (from 1)
Receiving objects: 100% (4961/4961), 7.95 MiB | 3.83 MiB/s, done.
Resolving deltas: 100% (2806/2806), done.

┌──(kali㉿kali)-[/opt/evilginx2]
└─$ 
```

The tool compiled successfully without build errors.

Problem 5: Binary Execution Path Confusion

Initial Attempts

- `sudo ./bin/evilginx`
- `sudo ./opt/evilginx`

Error Observed

- `command not found`

Cause:

Evilginx2 places the compiled binary in the **project root directory**, not in `/bin` or `/opt`.

Solution Applied

- `sudo ./evilginx`

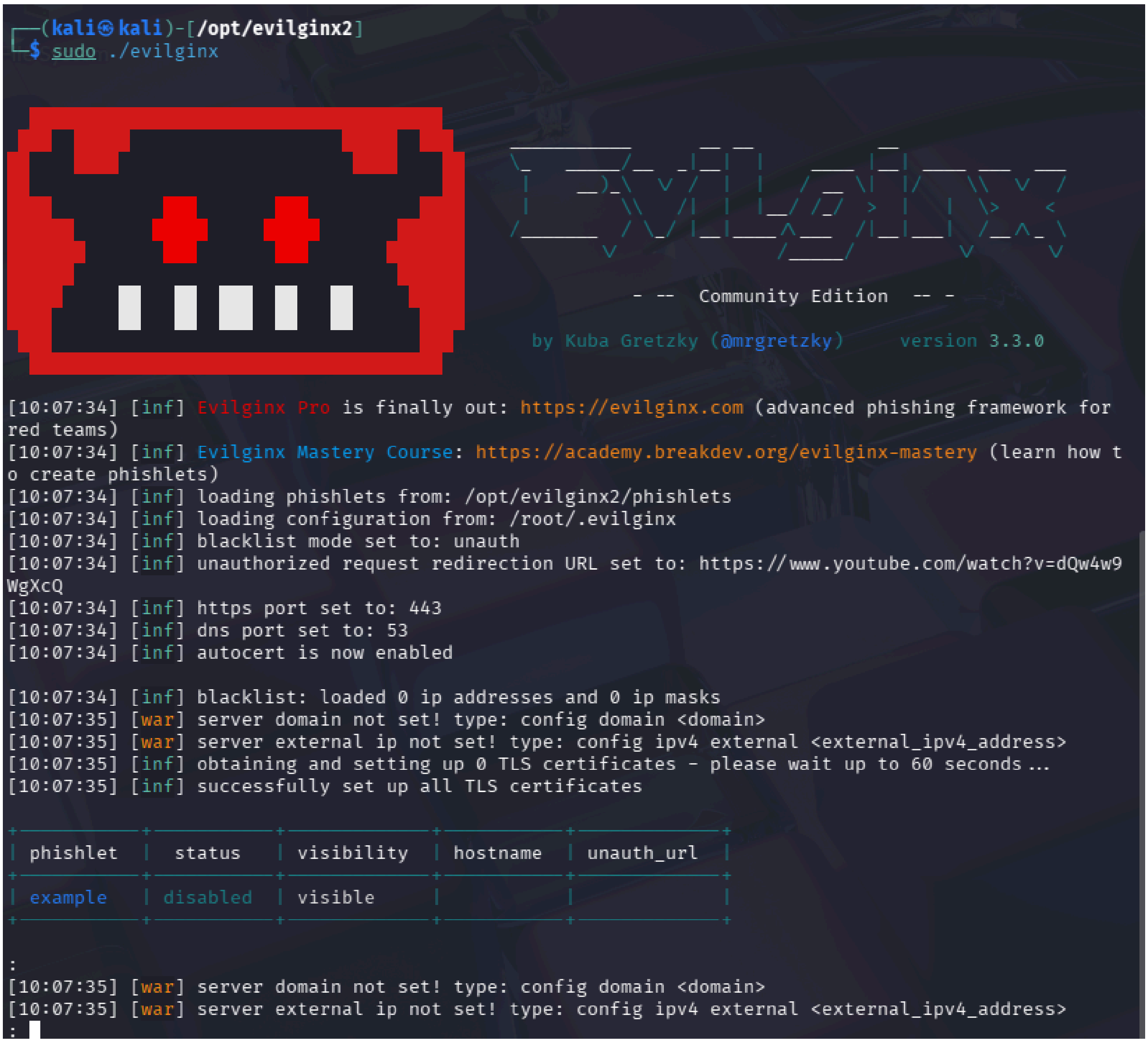
Successful Execution

After executing the correct command:

- Evilginx2 launched successfully
- TLS certificates were automatically generated
- Phishlets were loaded correctly
- HTTPS (443) and DNS (53) services were initialized

Warnings Observed (Non-blocking)

- `server domain not set`
- `server external ip not set`



```
┌──(kali㉿kali)-[/opt/evilginx2]
└─$ sudo ./evilginx

                                        ___________      .__.__            .__
                                        \_   _____/__  _|__|  |    ____   |__| ____ ___  ___
                                         |    __)_\  \/ /  |  |   / ___\  |  |/    \\  \/  /
                                         |        \\   /|  |  |__/ /_/  > |  |   |  \>    <
                                        /_______  / \_/ |__|____/\___  /  |__|___|  /__/\_ \
                                                \/              /_____/           \/      \/

                                                    - --  Community Edition  -- -

                                          by Kuba Gretzky (@mrgretzky)     version 3.3.0

[10:07:34] [inf] Evilginx Pro is finally out: https://evilginx.com (advanced phishing framework for
red teams)
[10:07:34] [inf] Evilginx Mastery Course: https://academy.breakdev.org/evilginx-mastery (learn how t
o create phishlets)
[10:07:34] [inf] loading phishlets from: /opt/evilginx2/phishlets
[10:07:34] [inf] loading configuration from: /root/.evilginx
[10:07:34] [inf] blacklist mode set to: unauth
[10:07:34] [inf] unauthorized request redirection URL set to: https://www.youtube.com/watch?v=dQw4w9
WgXcQ
[10:07:34] [inf] https port set to: 443
[10:07:34] [inf] dns port set to: 53
[10:07:34] [inf] autocert is now enabled

[10:07:34] [inf] blacklist: loaded 0 ip addresses and 0 ip masks
[10:07:35] [war] server domain not set! type: config domain <domain>
[10:07:35] [war] server external ip not set! type: config ipv4 external <external_ipv4_address>
[10:07:35] [inf] obtaining and setting up 0 TLS certificates - please wait up to 60 seconds...
[10:07:35] [inf] successfully set up all TLS certificates

+-----------+-----------+-------------+-----------+-------------+
| phishlet  |  status   | visibility  | hostname  | unauth_url  |
+-----------+-----------+-------------+-----------+-------------+
| example   | disabled  | visible     |           |             |
+-----------+-----------+-------------+-----------+-------------+

:
[10:07:35] [war] server domain not set! type: config domain <domain>
[10:07:35] [war] server external ip not set! type: config ipv4 external <external_ipv4_address>
:
```

Explanation:

These configurations are required for live phishing campaigns. Since this focused on tool deployment and validation, these warnings did not prevent execution.

Conclusion

In this practical, phishing simulation tools **Gophish** and **Evilginx2** were installed and analyzed in a controlled lab environment. Gophish was successfully built and configured up to the campaign creation stage; however, the absence of a valid SMTP infrastructure prevented successful email delivery. This highlights a critical real-world dependency of phishing frameworks on external services such as mail servers.

Evilginx2, on the other hand, was successfully installed and executed, demonstrating its readiness for advanced phishing simulations. Minor configuration warnings were observed but did not affect tool functionality within the scope of this lab.

Overall, this practical provided valuable insight into the operational complexity of phishing tools, emphasizing that successful phishing simulations require not only software tools but also proper supporting infrastructure, network configuration, and external service integration.