Brycen Fulscher

Kaleb Lingard

Group 5

IT 359 002

**GoPhish Project Report**

[**IT359 GoPhish Project Github Link**](https://github.com/bfulscher/IT359Project)

**GoPhish Introduction**

GoPhish is an open-source phishing simulation tool designed for security professionals and organizations to test and improve their phishing awareness. It allows them to create, send, and track phishing email campaigns, helping identify vulnerabilities in an organization. GoPhish enables the creation of customized phishing emails and landing pages that mimic real-world attacks. GoPhish also provides an easy-to-use dashboard to help show all the information we would need in one location. It provides detailed reports on user interactions, such as email openings, link clicks, and credential submissions. By simulating phishing attacks in a controlled environment, GoPhish helps organizations educate users, strengthen cybersecurity awareness, and implement better defenses against social engineering threats.

**Project Goals**

The goal of this project is to use GoPhish to create a sample phishing email campaign to help show a common method used for phishing and help bring awareness to how phishing is rolled out in the real world. The outcomes of this project include three major parts. First is the fake login page. It is a platform for users to type in their credentials and unintentionally give up information! Second, we will create phishing emails. This includes creating an email template and attack and victim accounts on MailHog. The third part involves the GoPhish analytics. We will go through the statistics and insight given by GoPhish to see what it has to offer as an ethical hacking tool.

**Project Scenario**

The overall scenario we will be creating is to be a company testing their employees on their phishing email awareness skills or simulating how a real threat actor may build out their phishing campaign. GoPhish is used by some major companies to help plan out their cybersecurity architecture. GoPhish will help show phishing vulnerabilities within their company. Our team will be keeping the perspectives of both a company testing their employees, and a malicious phishing attack approach, to understand how GoPhish is used.

**Project Installation**

GoPhish Install:

1. wget

[https://github.com/gophish/gophish/releases/latest/download/gophish-linux 64bit.zip](https://github.com/gophish/gophish/releases/latest/download/gophish-linux-64bit.zip)

1. sudo apt update && sudo apt install
2. unzip -y unzip gophish-linux-64.zip -d gophish
3. cd gophish
4. chmod +x gophish
5. sudo ./gophish

From here, the gophish command will give you a temporary password and a URL (127.0.0.1:3333) to access the Gophish admin tab from the admin tab is where we can build out our campaign (landing pages, email templates, mail accounts, etc.)

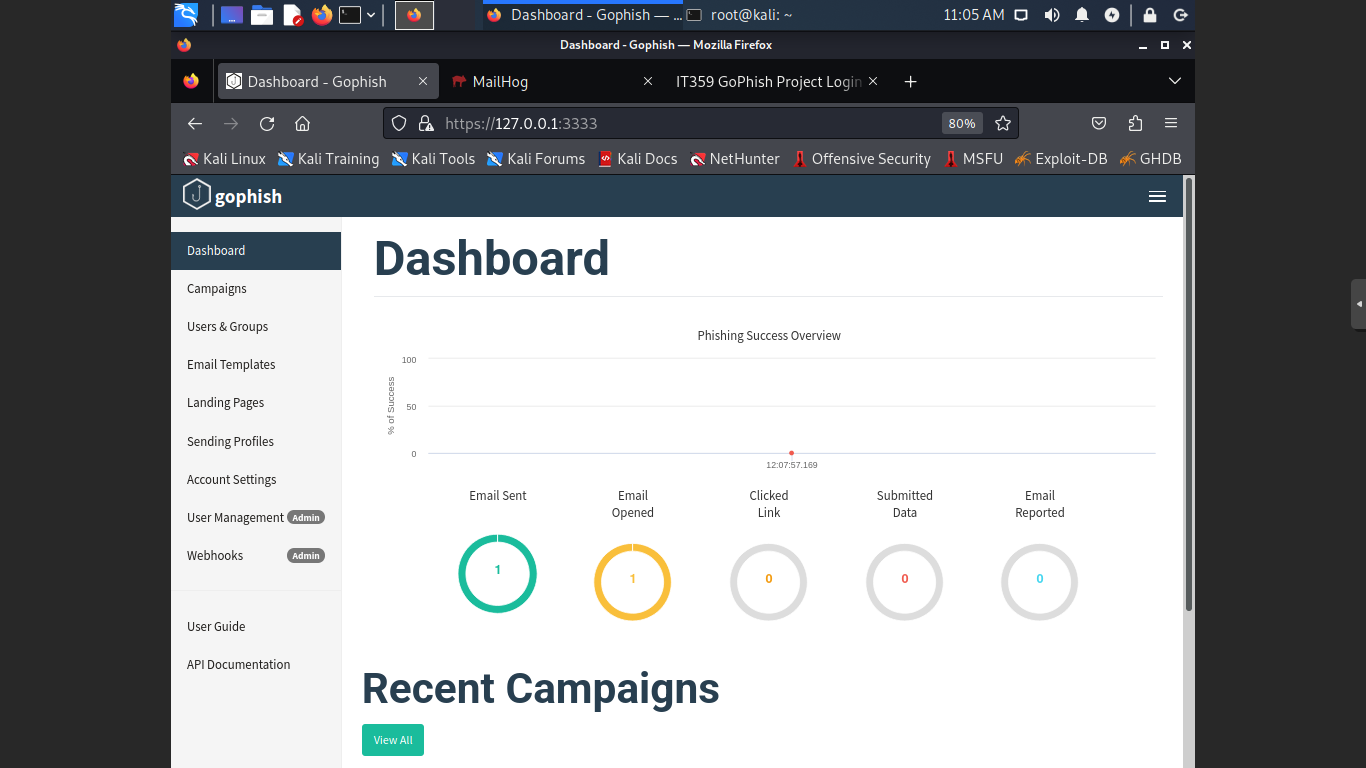
MailHog Install:

1. wget <https://github.com/mailhog/MailHog/releases/latest/download/v1.0.1/MailHog_linux_amd64>
2. chmod +x MailHog\_linux\_amd64
3. sudo mv MailHog\_linux\_amd64 /usr/local/bin/mailhog
4. mailhog

Mailhog will run indefinitely from the command line while you want the service open. When running the command, it will give you a URL (0.0.0.0:8025) access MailHog from.

**GoPhish Functionality**

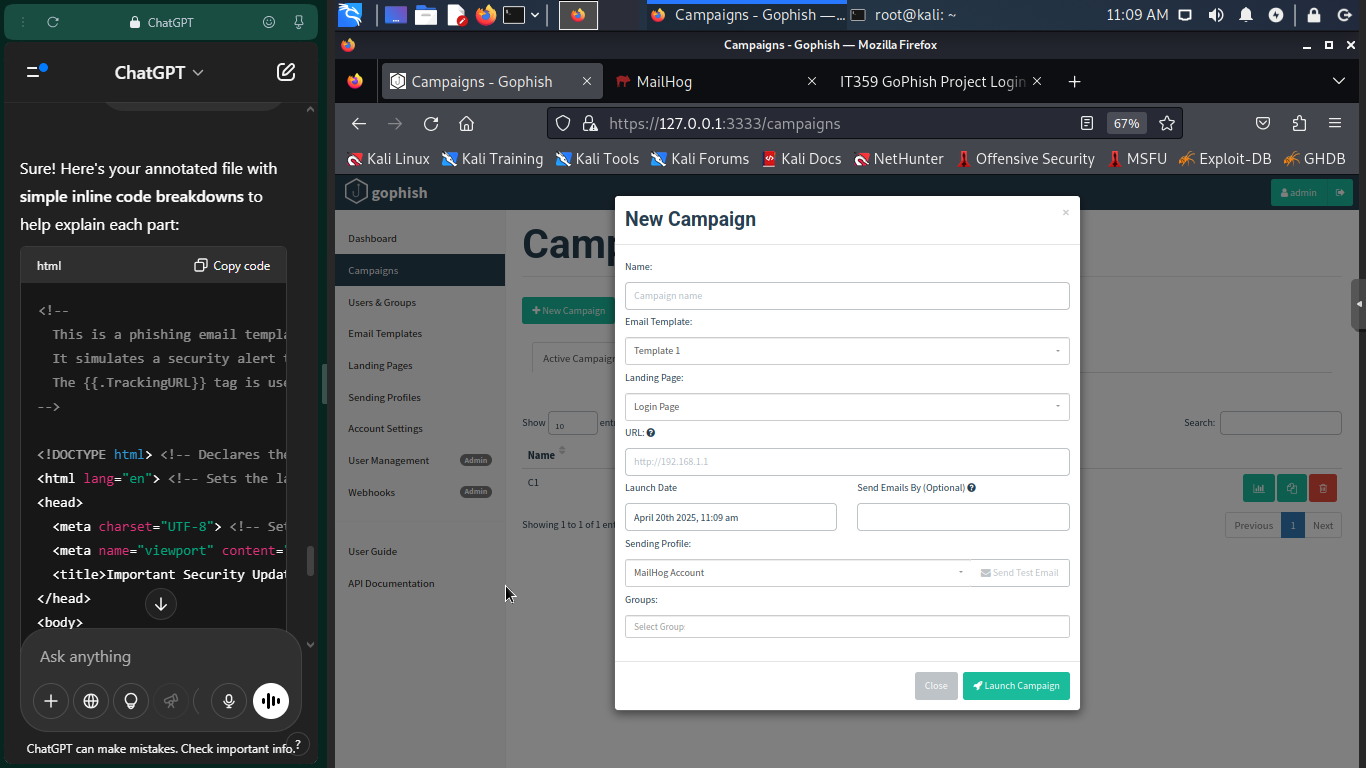
GoPhish provides a web-based interface to design, launch, and monitor the phishing campaigns, along with showing us statistics about the users' interactions with the campaigns. Below are the main features and tabs of the GoPhish interface, and this is what we navigated through when building out our campaign. There is also a screenshot showing what GoPhish looks like.



Core Features of GoPhish

* Dashboard: Offers a real-time overview of campaigns, email status, and captured credentials.
* Email Templates: Allows creation of custom phishing emails with tracking features like {{.TrackingURL}} to detect when emails are opened. We created emailtemplate.html for our campaign.
* Landing Pages: Lets users create or clone login pages that mimic real sites to capture credentials. We chose to build our own webpage (login.html) to help avoid confusion of a real phishing attack.
* Users & Groups: Manages target recipients by importing CSVs with names and emails.
* Sending Profiles: Configures SMTP settings to send emails from a legitimate-looking source. Here we added our MailHog sending account.
* Campaigns: Launches and manages phishing simulations, tracks opens, clicks, and submitted data. It can also be scheduled to help with automation.

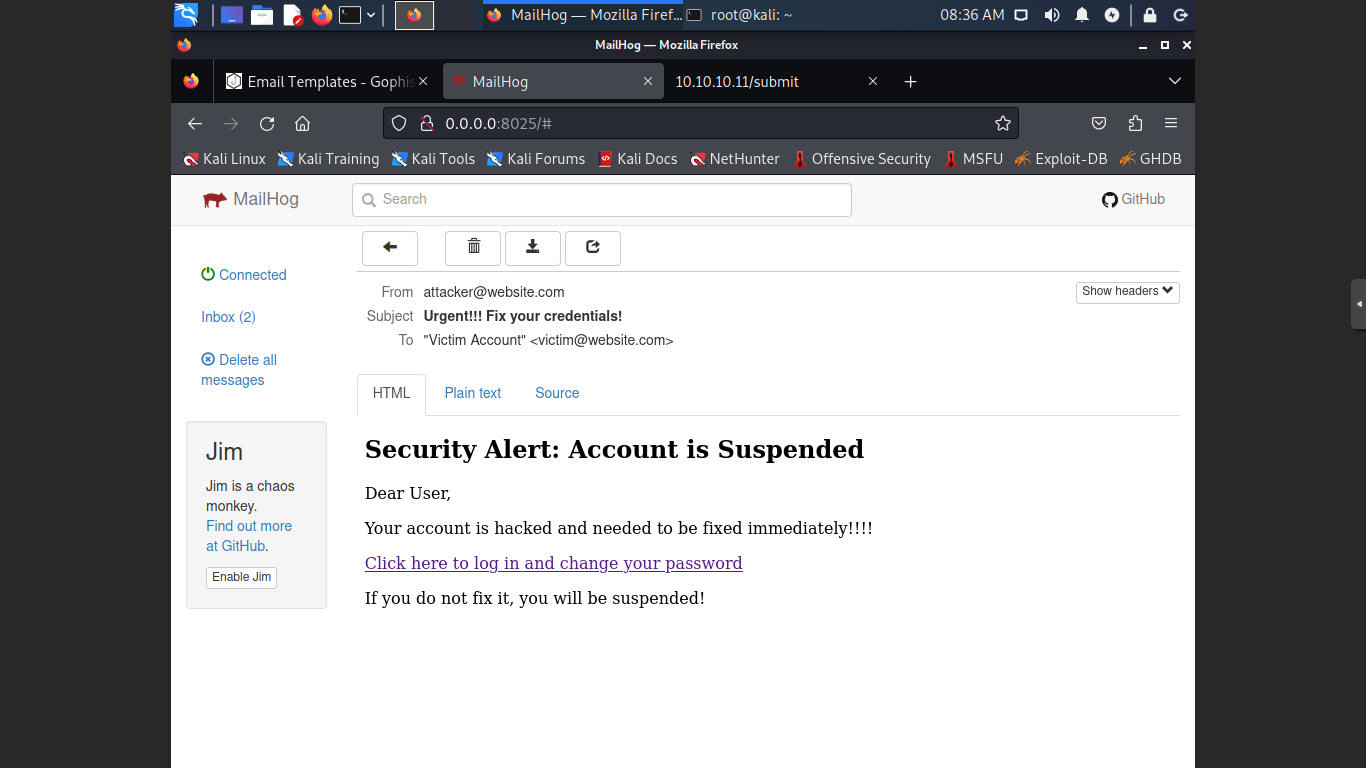
When creating a campaign, we can click New Campaign and put our phishing test together. Below is a screenshot of what it looks like to build a GoPhish campaign.

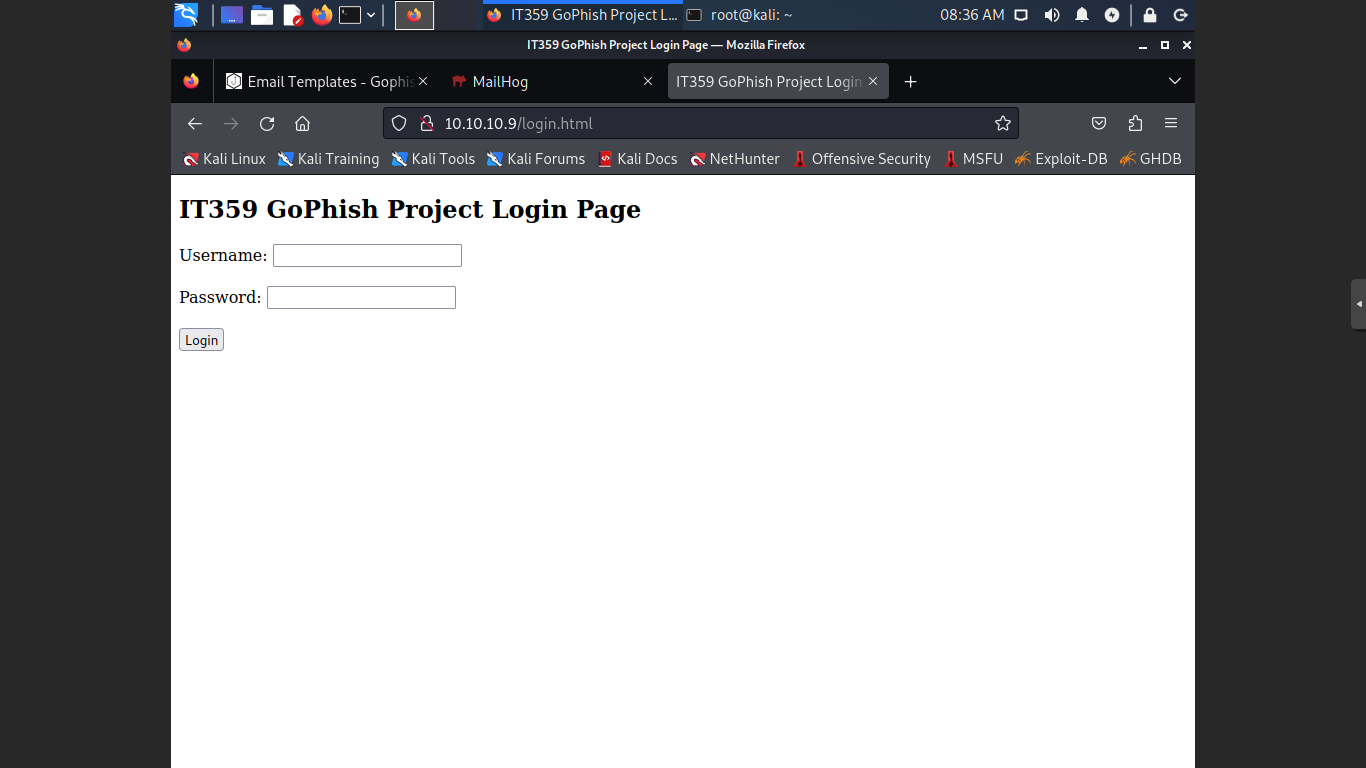


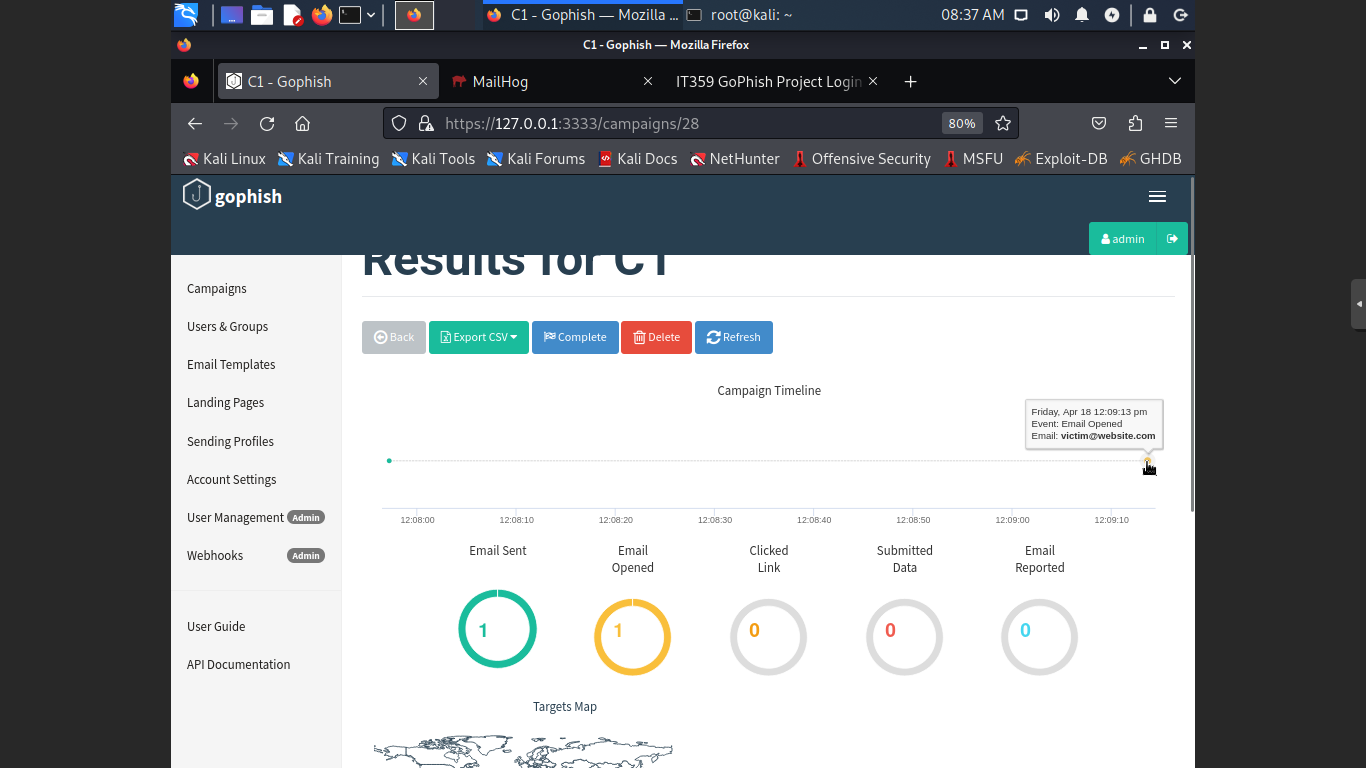
To launch a campaign, we just select from the drop boxes what we want to use. We chose the name to be C1, the email template we created and added to GoPhish, the Login page we made for the project, the URL, which is the URL from which we are listening from (Kali http://10.10.10.11), we can select a launch date, and finally we choose our sending account and to which groups to send this phishing email to. Then you press Launch Campaign, and the emails are sent out.

**Project Results**

In the end, our project allowed us to build out and schedule a phishing email campaign with a fake login page and a phishing email. We were also able to track when the phishing email victims receive and open their phishing email. Our ultimate goal was to track credentials of the phishing victim, but we were not able to achieve that. Our phishing email login page was functional and allowed for successful logins based on our php and MySQL backends. The email template we made was also successful. It was not meant to be actually convincing, but it does fill its role and does act as a foundation for us to work with. Finally, we used MailHog to view SMTP traffic and our testing emails from a realistic view. Below, we will show some screenshots of our project results:

This is the view of the phishing email from MailHog. We used a fake email for the sender and receiver, and using our email template, it appears as shown above, and not as plain text. The messages try to create a fake sense of urgency and authority.

This is the login and landing page. It is very simple, and it just meant to act as a platform for phishing email to track credentials. In a real scenario, it would likely mimic a real website like Amazon, Microsoft, or Google to gather credentials, but we did not want to make a convincing login page.



This is the Results tab for our campaign named C1. It gives us many different stats including links clicked, submitted data, and if the email was reported or not. We can see that we did log that the email was sent and that the email was opened. GoPhish also tracks location on a map of where the email was interacted with and the timeline of the campaigns.

**Project Reflection**

When looking back on the project, while not achieving our desired outcomes, we did learn a lot about GoPhish and phishing attacks. GoPhish is a great platform and makes the deployment of a phishing email campaign incredibly easy. The UI and application end of GoPhish is easy to use and made our job straightforward in terms of phishing deployment. The problem when building a phishing campaign is in web development and social engineering. The web development was the most difficult as managing the basic functionality of a login website and then trying to combine it with a GoPhish tracking system, was something that we were not experienced or prepared for. As a small, unexperienced group, the web development was difficult, but outside of the GoPhish implementation, the landing page and its login functionality is something we are proud of. While not immediately convincing or usable for a real phishing test, we do understand the tool and feel confident in building out a real phishing email test, if our future careers or employers called for it. The other side of phishing is social engineering. For us, we did not want to develop a convincing email, and there was a careful balance when creating our email templates. We enjoyed the activity and scenario we built, but ultimately did not want to harm anyone. It did allow us to see into the side of an attacker though, which was an insightful view into the penetration testing landscape.