# Simulation avec CamPhish

## 1. Contexte et objectif

CamPhish est un outil open source de démonstration de techniques d'ingénierie sociale combinant plusieurs vecteurs d'attaque : hameçonnage par lien piégé, demande d'accès à la caméra du navigateur, et collecte de données de géolocalisation.

## 2. Installation

CamPhish est cloné depuis son dépôt GitHub officiel et exécuté directement via un script bash, sans installation de paquet supplémentaire :

```
git clone https://github.com/techchipnet/CamPhish
cd CamPhish
bash camphish.sh
```

L'outil nécessite PHP (pour le serveur local) et un accès Internet pour établir le tunnel de redirection. Aucune dépendance supplémentaire n'est requise sur une installation Debian standard.

## 3. Configuration de la campagne

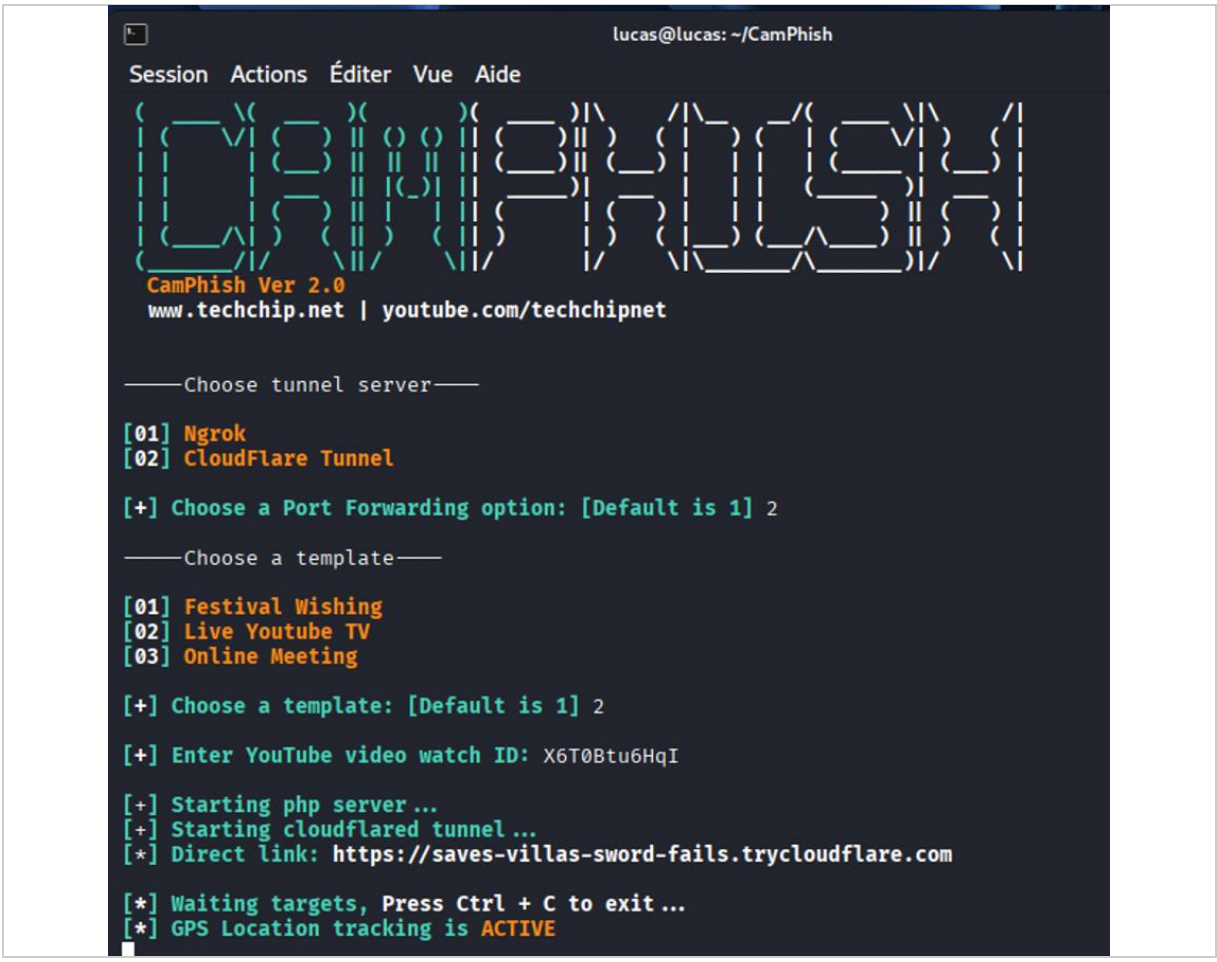
### 3.1 Choix du tunnel et du template

Au lancement, CamPhish propose deux options de tunneling pour exposer la page piégée sur Internet : Ngrok ou Cloudflare Tunnel. Cloudflare Tunnel (option 02) a été retenu car il ne nécessite pas de compte ni de token d'authentification, contrairement à Ngrok.

L'outil propose ensuite trois templates de pages leurres, destinées à inciter la cible à accepter l'accès à sa caméra :

- Festival Wishing : fausse page de vœux festifs
- Live Youtube TV : fausse page de vidéo YouTube en direct (template utilisé ici)
- Online Meeting : fausse page d'invitation à une réunion en ligne

Le template "Live Youtube TV" (option 02) a été sélectionné. L'outil demande ensuite l'identifiant d'une vidéo YouTube réelle, affichée en arrière-plan pour renforcer la crédibilité de la page.




## 3.2 Génération du lien piégé

Une fois la configuration validée, CamPhish démarre un serveur PHP local et établit automatiquement un tunnel Cloudflare. Un lien public en *.trycloudflare.com est généré et affiché dans le terminal. Ce lien est celui à partager avec la cible (ou à ouvrir soi-même dans le cadre d'un test).

L'outil se met ensuite en attente d'une connexion, affichant le message "Waiting targets, Press Ctrl+C to exit" ainsi que la confirmation que le tracking GPS est actif.

# 4. Mécanisme de l'attaque et données collectées

## 4.1 Côté victime : demande de permissions

Lorsque la cible ouvre le lien dans son navigateur, elle voit s'afficher la vidéo YouTube choisie. Le navigateur déclenche simultanément une demande d'autorisation d'accès à la caméra et à la localisation, présentée comme une simple popup système. Si la cible accepte, les données sont immédiatement transmises au serveur CamPhish.




Ce scénario illustre un principe fondamental de sensibilisation : une popup de permission navigateur peut provenir de n'importe quel site, y compris un site malveillant, et l'accepter sans vérifier l'URL source expose l'utilisateur à des risques réels.

## 4.2 Données reçues par l'attaquant

Dès que la cible interagit avec la page, CamPhish collecte et affiche dans le terminal :

- L'adresse IP de la cible (IPv4 et IPv6 si disponible), enregistrée dès l'ouverture du lien
- Les coordonnées GPS (latitude, longitude, précision en mètres) et un lien Google Maps direct, si la permission de localisation est accordée
- Des photos prises par la caméra frontale ou arrière de l'appareil, stockées localement dans le dossier CamPhish, si la permission caméra est accordée

```
[+] Target opened the link!
[+] IP: 35.196.132.85

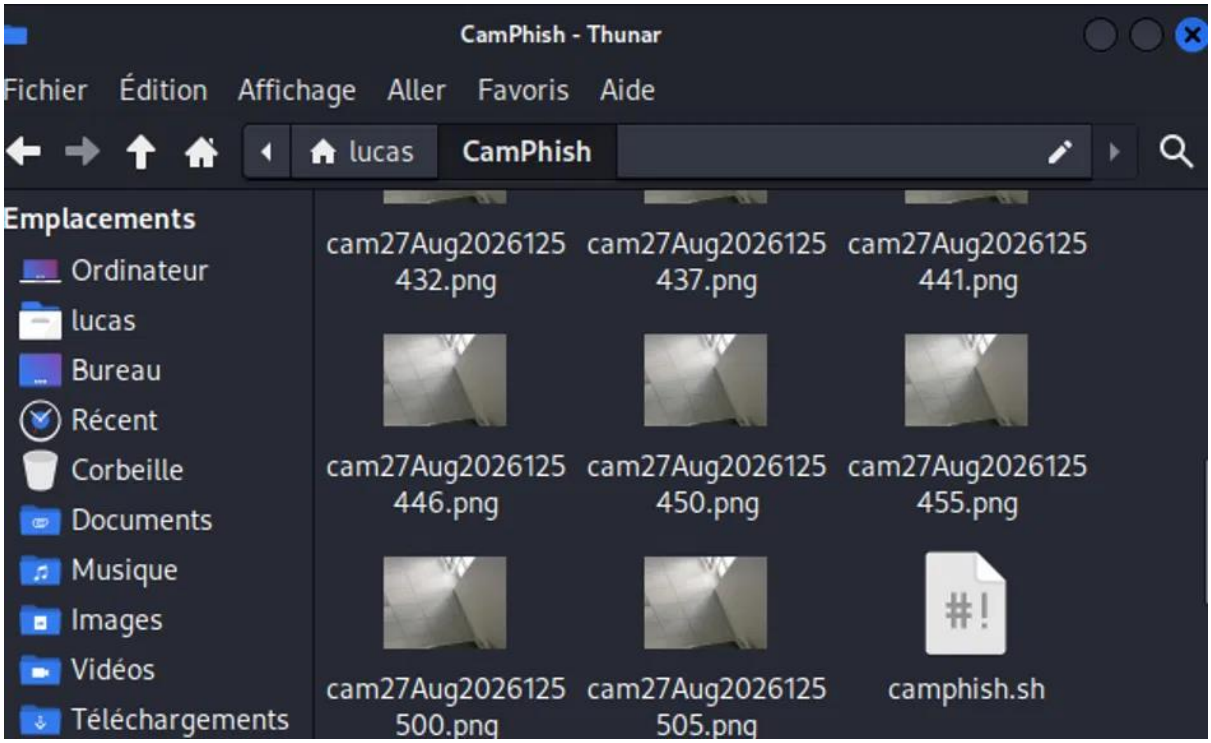
[+] Target opened the link!
[+] IP: 2a02:8440:850e:377:28bd:d37:add9:5a9f

[+] Location data received!
[+] Current location data:
Latitude: 48.690073910182136
Longitude: 6.187217854632751
Accuracy: 14.24595459057527 meters
Google Maps: https://www.google.com/maps/place/48.690073910182136,6.187217854632751
Date: 27Aug2026125025

[!] No location file found

[+] Location data received!
[!] No location file found

[+] Cam file received!
```



## 6. Conclusion

CamPhish illustre de manière concrète comment une attaque combinant ingénierie sociale, fausse interface et tunneling peut collecter des données sensibles (position GPS, image) à partir d'un simple lien cliqué par la cible. Cette démonstration, réalisée sur un appareil de test personnel dans un environnement de laboratoire, s'inscrit dans une démarche de sensibilisation aux risques liés aux permissions accordées dans le navigateur.