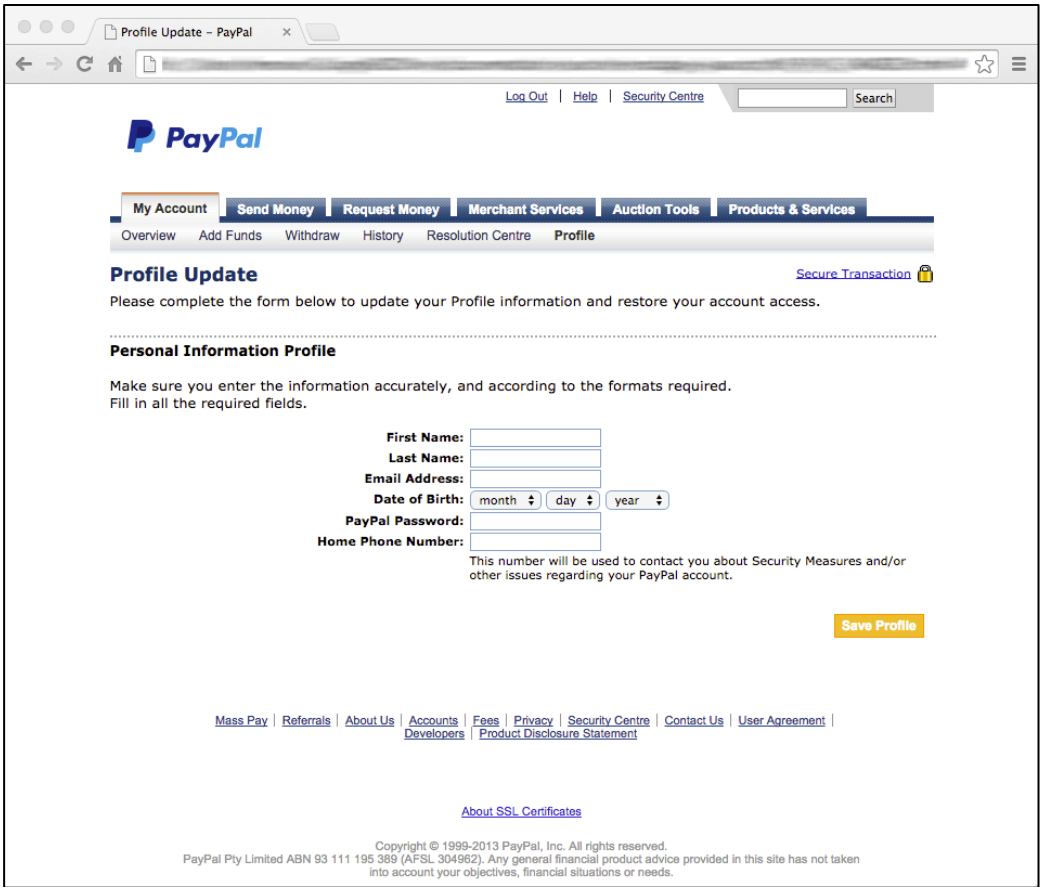
Licence L2 STS Mention SPI Parcours Informatique
Unité 174EN007
Sécurité Informatique

TP1
« Dans la peau d'un cybercriminel »

Ce premier TP porte sur l'étude d'un exemple de *phishing* constitué d'une page HTML reçu par mail. Une autre manière de procéder est de donner un lien web dans le corps du mail vers un faux site web.



Dans l'exemple étudié, le scénario est le suivant :

1. Le cybercriminel envoie un mail constitué d'une page HTML.
2. Le destinataire réceptionne le mail et le visualise dans le lecteur de mail capable d'afficher une page HTML.
3. Le destinataire remplit les champs demandés (des informations confidentielles).
4. Après validation de la page, la requête est traitée par une page PHP installée frauduleusement sur un serveur compromis (accès en écriture pour le cybercriminel).
5. Le script PHP sauvegarde localement sur le serveur compromis les informations confidentielles.
6. Le cybercriminel n'a plus qu'à récupérer régulièrement les informations stockées sur le serveur compromis et s'en servir…

Téléchargez puis dézippez le fichier *exemple_phishing.zip* (cf. UMTICE).

Question n°1

Quand vous ouvrez le fichier *NE_PAS_OUVRIR.html* simultanément dans un navigateur internet (Chrome, FireFox, Internet Explorer …) et dans un éditeur de texte voyez-vous une relation entre les deux contenus visualisés dans le navigateur et dans l'éditeur ?

Selon vous, pourquoi les deux contenus ne semblent pas être en relation ?

Remarque pour l'ouverture dans l'éditeur texte :
Le fichier est encodé en ISO Latin 1 avec deux caractères de fin de ligne (CRLF).

Question n°2

Donnez la liste des informations confidentielles obtenues par la page du navigateur.

Donnez l'adresse de la page PHP qui gère la requête finale (réception des informations confidentielles).

Comment réagit la page PHP qui gère la requête finale (quelles informations sont affichées par le navigateur) ? D'après vous, pourquoi le site réagit de cette manière ?

Question n°3

Etudiez le fichier *NE_PAS_OUVRIR.html* ouvert dans l'éditeur de texte.

Aides :
- Le code JavaScript est masqué (offusqué) de plusieurs manières possibles
- Rappel du code ASCII en hexadécimal : table ASCII sur wikipedia.org
- Aide mémoire JavaScript : http://www.xul.fr/ecmascript/aide-memoire-javascript.php
- Aide mémoire syntaxe JavaScript : https://fr.wikipedia.org/wiki/Syntaxe_JavaScript

Extrayiez la fonction *bf9r* et insérez-là dans un nouveau fichier nommé *test.html* avec la structure suivante :

```
<!DOCTYPE HTML>

<script type="text/javascript">

function bf9r(texte) {
        …
}

bf9r("<h1>texte à regarder</h1>")

</script>
```

- Que donne l'exécution avec le texte "<h1>texte à regarder</h1>" ?
- Que donne l'exécution en appliquant *bf9r* sur le résultat de la question précédente ?
  Remarque : le caractère " doit être protégé avec \ pour bien définir l'argument de *bf9r*.

- Qu'en déduisez-vous pour le fonctionnement de la fonction *bf9r* : est-ce une fonction à double sens (un résultat précédent appliqué comme argument permet de retrouver l'argument initial) ou à sens unique ?
- Compte tenu de l'objectif du hacker et de l'utilisation de la fonction *bf9r* dans le fichier HTML, qu'en déduisez-vous pour la fonction *bf9r* : est-ce une fonction de codage, de décodage ou les deux ?
- Donnez au moins trois manières dont l'obfuscation du code a été faite.
- Proposez une ré-indentation du code plus lisible.
- Insérer des commentaires explicatifs pour chaque ligne de code.
- Convertir chaque ligne de code en son équivalent non offusqué.
- Vérifiez que la fonction fonctionne normalement.
- Expliquez en quelques lignes le principe de fonctionnement de la fonction *bf9r*.

Mettez le code source final du fichier *test.html* dans le compte-rendu.

Question subsidiaire

Trouvez la fonction faisant le travail complémentaire de la fonction *bf9r* que vous appellerez *r9fb*.
Testez cette fonction avec la chaîne : "/9$ue'sO%Zzl'xO)!ll{U"
Quel est le résultat (code HTML) ?

Mettez le code source proposé de la fonction *r9fb* dans le compte-rendu.

Travail à rendre

Dans un compte-rendu au format PDF, vous répondrez-aux questions et incorporerez les codes sources demandés. Les codes sources seront dans la police Arial 10 points.