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

TP1 - Correction
« 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.

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 ?

Normalement le fichier devrait contenir du code HTML correspondant à la page affichée dans le navigateur web.

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

Du code JavaScript génère dynamiquement le contenu HTML.

Question n°2

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

Nom, prénom, email, date de naissance, mot de passe PayPal, numéro de téléphone, adresse complète, nom de jeune fille, numéro de carte bancaire, date d'expiration, numéro de vérification de la carte de crédit

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

http://www.maquinasitalianinha.com.br/imagens/PT/new.php

Comment réagir 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 ?

La page n'est plus disponible. Le page frauduleuse a été trouvée et supprimée.

Question n°3

- Que donne l'exécution avec le texte "<h1>Texte à regarder</h1>" ?

La chaîne suivante : "/9$uE'[wiZào'NH(BH{JjAC}_" (en UTF-8)
La chaîne suivante : "/9$uE'[wiZÃ ©UYTo_T" (en ISO Latin 1)

- Que donne l'exécution en appliquant *bf9r* sur le résultat de la question précédente ?

***La chaîne suivante :* "kp['^^&Rgàl{GBPbJ fq9_L\"** *(en UTF-8)*

- Expliquez les différences entre le résultat attendu et le résultat obtenu.

***On retrouve les caractères accentués mais pas les autres caractères car les caractères accentués n'existent pas dans le tableau* dguw.**

- Qu'en déduisez-vous pour la fonction *bf9r* : est-ce une fonction de codage, de décodage ou les deux ?

C'est une fonction de décodage uniquement.

- Donnez au moins trois manières dont l'offuscation du code a été faite.

(a) La fonction bf9r décode du caractère codé auparavant.
(b) Utilisation de code hexadécimal.
(c) Ne pas utilisez directement de nom de fonction
(d) Utilisez des noms de variables non signifiants de 4 caractères arbitraires
(e) « Compresser » le code : pas d'indentation ; pas de mise en forme en général

- 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 toujours normalement.
- Expliquez en quelques lignes le principe de fonctionnement de la fonction *bf9r*.

Chaque caractère est codé selon qu'il existe dans le tableau ou pas. S'il n'existe pas dans le tableau, il n'est pas codé (*sortie += cc*). Par contre, s'il existe dans le tableau, il est transformé en autre caractère se trouvant à l'index calculé par* index = index – (i+1)%taille.** ***Le tableau semble classé aléatoirement.

```
<!DOCTYPE HTML>
<script type="text/javascript">
function decode(chaine) {
	var i,
	cc,
	alphabet = "{R@?YNJ^_BiDU\'[|0$Ee1x6TH\trOsu8CP>5Im-%gvQd)24©o(Zp.9\rhq7yz
S&KWa+nV~wL#,]A\"jtI\n!\\cFf=:XMGb;3*}k/<",
	index,
	taille = alphabet.length,
	sortie = "";

	for(i=0; i<chaine.length; i++) {			// On parcours la chaine cryptée
		cc = chaine.charAt(i);				// Le caractère courant
		index = alphabet.indexOf(cc);			// L'index du caractère courant dans le tableau alphabet
		if (index > -1) {				// Si l'index est positif (le caractère est dans le tableau)
			index -= (i+1) % taille;	// Nouvelle index dans le tableau en faisant attention à ne pas déborder
			if (index < 0) {			// Si le nouvelle index est en dehors du tableau
				index += taille;	// Ajout à l'index la taille du tableau pour être sûr d'être dedans
			}
			sortie += alphabet.charAt(index);		// Ajout du nouveau caractère décodé
		}
		else {						// Sinon l'index est négatif
			sortie += cc;				// Ajout du caractère courant sans décodage
		}
```

```
	}

	document.write(sortie); this.sortie = null;		// Ecriture dans le document
}

decode("<html>code à décoder </html>");
//decode("/9AP8O\"g>B(à©sén8H{J%\"]89i_N");

</script>
```

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.

```
function code(chaine) {
	var i,
	cc,
	alphabet = "{R@?YNJ^_BiDU\'[|0$Ee1x6TH\trOsu8CP>5lm-%gvQd)24©o(Zp.9\rhq7yz
S&KWa+nV~wL#,]A\"jtl\n!\\cFf=:XMGb;3*}k/<",
	index,
	taille = alphabet.length,
	sortie = "";

	for(i=0; i<chaine.length; i++) {			// On parcours la chaine cryptée
		cc = chaine.charAt(i);			// Le caractère courant
		index = alphabet.indexOf(cc);		// L'index du caractère courant dans le tableau alphabet
		if (index > -1) {			// Si l'index est positif (le caractère est dans le tableau)
			index += (i+1) % taille;	// Nouvelle index dans le tableau en faisant attention à ne pas déborder
			if (index < 0) {		// Si le nouvelle index est en dehors du tableau
				index += taille;	// Ajout à l'index la taille du tableau pour être sûr d'être dedans
			}
			sortie += alphabet.charAt(index);		// Ajout du nouveau caractère codé
		}
		else {					// Sinon l'index est négatif
			sortie += cc;			// Ajout du caractère courant sans codage
		}
	}

	document.write(sortie); this.sortie = null;		// Ecriture dans le document
}


//decode("<html>code à décoder </html>");
//decode("/9AP8O\"g>B(à©sén8H{J%\"]89i_N");
//code("/9AP8O\"g>B(à©sén8H{J%\"]89i_N");

decode("<h1>Hello world </h1>");
code("/9$fue'sO%Zzl'xO)!ll{U");
```