Comment protéger son adresse email sur le web

Le Spam est devenu un véritable fléau. Certaines études annoncent que ces courriers non sollicités représentent jusqu'à 80% des courriers échangés dans le monde.

Comment les spammeurs ont-ils obtenu votre adresse ?

Comment éviter qu'une adresse email ne se retrouve dans la base de données d'un spammeur ?

1- Méthodes utilisées par les spammeurs

Génération Automatique

Le spammeur va choisir un domaine cible et va tenter a@domaine.com puis aa@domaine.com, etc. Il peut aussi utiliser un dictionnaire de mot et tenter chaque mot. Cette méthode n'est pas efficace et une adresse normale du type prenom.nom@domaine.fr n'a que peu de chance d'être générée de la sorte.

Vérolage

Les spammeurs ont de plus en plus recours à des virus qui n'ont pour but que de récupérer une liste d'adresses email contenues dans les fichiers ou dans le carnet d'adresses de l'ordinateur infecté. Une fois la liste constituée, le virus peut soit envoyer des spams lui-même, soit envoyer la liste d'adresses chez le spammeur qui pourra ensuite la re-utiliser ou la revendre. Il est dur de se protéger de ce genre de méthode puisqu'il suffit qu'un virus contamine l'ordinateur d'une personne ayant reçu un seul mail de votre part. Maintenez quand même votre antivirus à jour pour ne pas être responsable du spam de vos correspondants !

Piratage

Il n'est pas rare que des pirates en s'introduisant dans des systèmes informatiques tombent sur des listes de milliers voir de millions d'adresses email de clients de banque ou autre. Ils ne se gênent pas pour revendre ces listes aux spammeurs.

Activité légale

Il se peut aussi que le spammeur monte une affaire tout à fait légale qui nécessite l'adresse email du client. Par exemple, les sites de webcards (e-carte postales) ou de news diverses, etc. Méfiez vous des sites internet douteux, utilisez une adresse email alternative pour ce genre de chose, cela évitera que votre adresse email personnelle ou professionnelle soit polluée.

Achat

Il existe de nombreuses sociétés qui revendent des listes d'adresses emails légalement ou illégalement constituées. Souvent, ces listes proviennent d'autres spammeurs qui ont revendu leur propre carnet d'adresses constitué à coups de piratage ou de vérolage. Malheureusement, ce genre d'activité n'est pas illégal.

Moissonnage des newsgroups

Les newsgroups regorgent d'adresses emails, il est très facile pour les spammeurs de les récupérer. Il n'est pas nécessaire d'utiliser une adresse email valide pour poster sur les newsgroups. Utilisez plutot une adresse alternative.

Moissonnage du web

Il s'agit probablement de la technique la plus utilisée. Le spammeur va utiliser un programme appelé “robot” qui analysera des millards de pages web à la recherche d'adresses email.

2- Protéger son adresse email sur le web

Il est souvent préférable et même parfois indispensable de laisser son adresse email aux visiteurs de son site internet. Il faut aussi savoir que de nombreux documents électroniques (documents word, pdt, ps, etc) sont indexés par les moteurs de recherche, rendant disponible sur le web toutes les adresses emails qu'ils contiennent.

Voici une liste de techniques permettant de déjouer les spammeurs dans leur quête.

2.1 -Les méthodes de base

Les méthodes suivantes permettent d'afficher votre adresse email, avec un lien ou non, en utilisant des mécanismes basiques pour empêcher un spammeur de récupérer votre adresse.

Simple

principe : l'adresse est simplement écrite dans la page :

```
trapwiki@univ-nantes.fr
```

- protection nulle, le spammeur est ravi
- vos visiteurs doivent copier coller l'adresse

Simple avec un lien

principe: l'adresse n'apparaît pas sur la page elle-même, il faut cliquer sur le lien.

```
<a href="mailto:trapwikilink@univ-nantes.fr">cliquez ici pour me contacter</a>
```

résultat :

cliquez ici pour me contacter

- protection nulle
- l'adresse est en clair dans le code, un robot collecteur utilise le source, pas le rendu

Écrite

principe : écrire l'adresse en toutes lettres :

```
trapwikiat AT univ-nantes DOT fr
```

- protection faible,
- un robot collecteur peut facilement recomposer l'adresse,
- vos visiteurs doivent taper manuellement votre adresse email dans leur logiciel de messagerie.
- applicable à un document électronique (word, pdt, etc)

Écrite avec lien

principe : faire un lien avec l'adresse écrite en toutes lettres :

```
<a href="mailto:trapwikiatlinkATuniv-nantesDOTfr">Contactez moi</a>
```

résultat :

Contactez moi

- protection moyenne,
- l'URL mailto trahit toujours la présence ici d'une adresse email,
- Vos visiteurs doivent penser à corriger l'adresse email manuellement après avoir cliqué.

Fausse adresse

principe : ajouter une portion à supprimer évidente dans l'adresse afin de la rendre invalide dans un lien.

```
<a href="mailto:trapwikifake@REMOVE_THIS_PART.univ-nantes.fr">Contactez moi</a>
```

résultat :

Contactez moi

- protection moyenne,
- la portion rajoutée doit être évidente afin de ne pas tromper vos visiteurs,
- un robot collecteur ne se laissera pas tromper non plus,
- vos visiteurs doivent penser à corriger l'adresse dans leur logiciel
- applicable à un document électronique (word, pdt, etc)

Utilisation de charactères unicodes #1

principe : utiliser des caractères spéciaux interprétés par le navigateur pour remplacer le @ et les points.

```
<a href="mailto:trapwikiunicode&#64;univ-nantes&#46;fr">Contactez moi</a>
```

résultat :

Contactez moi

- protection moyenne,
- transparent pour le navigateur
- le mailto trahit la présence d'une adresse email
- un robot collecteur remplacera aussi les caractères spéciaux par les bons

Utilisation de charactères unicodes #2

principe : tout encoder en unicode y compris le “mailto :”

```
<a
href="&#109;&#97;&#105;&#108;&#116;(...)&#116;&#101;&#115;&#46;&#102;&#114;"
>Contactez moi</a>
```

résultat:

Contactez moi

- bonne protection,
- plus de mailto,
- pour convertir votre adresse en unicode: http://www.pinnacledisplays.com/unicode-converter.htm

Encodage hexadécimal

principe : Encoder l'adresse en caractères hexadécimaux interprétés par le navigateur.

```
<a href="mailto:%74%72%61%70%77%69%6B%69%68%65%78%61%40%75%6E%69%76%2D%6E%61%6E%74%65%73%2E%66%72">Contactez moi</a>
```

résultat :

Contactez moi

- protection moyenne,
- utilisation de mailto trahissant la présence d'une adresse,
- le robot peut décoder comme votre navigateur le fait
- une page pour convertir votre adresse en hexadecimal : http://www.theproblemsite.com/codes/hex.asp

Utilisation de commentaires HTML

principe : insérer dans l'adresse une série de commentaires HTML qui ne seront pas rendus par le navigateur

```
t<!-- @. -->r<!-- @. -->a<!-- @. -->p<!-- @. -->w<!-- @. -->i<!-- @. -->k<!-- @. -->i<!-- @. -->etc.
```

résultat :

trapwikicomment@univ-nantes.fr

- protection plutôt bonne
- impossible d'utiliser cette protection dans un lien
- un robot puissant pourra passer outre

Remplacement de l'arobase par une image

principe : Remplacer l'arobase par une image dont le nom est sans rapport.

```
trapwikiimgat<img src="images/separateur.jpg">univ-nantes.fr
```

résultat :

trapwikiimgat@univ-nantes.fr

- protection plutôt bonne
- impossible d'utiliser cette protection dans un lien
- un robot avancé pourra reconstituer l'adresse
- applicable à un document électronique (word, pdt, etc)

Utilisation d'une image

principe : Utiliser une image contenant l'adresse email.

```
<img src="images/adresse.jpg">
```

résultat :

- bonne protection si l'image n'est pas trop simple
- impossible à utiliser dans un lien
- applicable à un document électronique (word, pdt, etc)

Utilisation d'une applet flash ou java

principe : même principe que la méthode de l'image, mais en utilisant flash ou java pour afficher l'adresse.

résultat :

mailto_fixe_texte_sujet.swf

- bonne protection
- oblige les visiteurs à avoir le plugin flash ou java d'installé
- peut être utilisé comme un lien

Je dis ça, je dis rien

principe : décrire l'adresse email sans l'écrire en entier :

«*si vous désirez me contacter, mon adresse email est **arnaud** suivi d'un **point** puis **abelard**, ensuite **arobase** et enfin **univ**, **tiret** (le signe -, pas le signe _) **nantes point** et pour terminer **fr**.*»

- bonne protection
- agaçant pour le visiteur (s'il a la chance de parler français)

2.2- Méthodes CSS

L'utilisation des styles permet de déjouer les robots.

L'élément "after"

principe : utiliser le pseudo-élément CSS 2 «after» pour ajouter du contenu dans un élément HTML:

```
.adresse:after { content: "trapwikicssafter\40univ-nantes.fr"; } /* \40 = @ */
```

```
<p class="adresse"></p>
```

résultat :

.adresse:after { content: "trapwikicssafter\40univ-nantes.fr"; }

Contactez moi:

- bonne protection,
- ne fonctionne qu'avec les navigateurs récents: Firefox 1.5, Netscape 8, IE 7(?),
- ne peut être utilisé dans un lien.

L'attribut unicode-bidi

principe : l'attribut unicode-bidi est utilisé pour changer l'orientation du texte d'un élément html. Dans notre cas, nous taperons l'adresse de droite à gauche et nous aurons un style CSS qui affichera dans le bon sens:

```
p span.reverse { unicode-bidi:bidi-override; direction: rtl; }
```

```
<p>Contactez moi:<span class="reverse">rf.setnan-
vinu@idibsscikiwpart</span></p>
```

résultat:

p span.reverse { unicode-bidi:bidi-override; direction: rtl;}

Contactez moi: rf.setnan-vinu@idibsscikiwpart

- bonne protection,
- mieux supporté que le pseudo-élément after,
- ne peut être utilisé dans un lien.

L'attribut display

principe : le principe est le même que pour les commentaires HTML : on va insérer dans l'adresse email des éléments html avec un style les rendant invisibles :

```
p span.invisible { display: none; }
```

```
<p>Contactez moi: trapwikicssdisplay<span class="invisible">@</span>@univ-
nantes<span class="invisible">@</span>fr</p>
```

résultat :

p span.invisible { display: none; }

Contactez moi: trapwikicssdisplay@@univ-nantes.@.fr

- bonne protection,

- ne peut être utilisé dans un lien.

2.3- Méthodes avancées

Il existe d'autres méthode pour cacher son adresse email aux spammeurs sans la cacher pour vos visiteurs. La plupart font appel à des scripts, javascript ou php.

javascript simple

principe : utiliser une fonction javascript afin de n'afficher l'adresse qu'au moment du rendu par le navigateur (le javascript est traité par le navigateur et pas par le serveur).

```
<script language=JavaScript>
```

```
function genereadresse(prenom,nom,reste) {
  document.write("<a href=" + "mail" + "to:" + prenom + "." + nom + "@" +
reste + ">" + prenom + "." + nom + "@" + reste + "</a>")
}
```

```
</script>

<!-- affichage de l'adresse dans la page web -->
<script language=javascript>genereadresse("trapwiki","js","univ-
nantes.fr");</script>
```

résultat :

Cette technique peut être couplée avec plusieurs méthodes simples abordées ci-dessus

- bonne protection
- peut être appliquée de manière automatique sur le site web de l'université

génération automatique d'image

principe : Le principe est le même que lors de l'utilisation d'une image sauf que cette fois-ci nous générons dynamiquement l'image

script adresse.php:

```
<?
/* adresse.php: fichier de génération d'image */

# récupération des paramètres
$nom=$_GET["nom"];
```

```
$prenom=$_GET["prenom"];

if(!isset($nom) && !isset($prenom)) $texte="erreur";
else if(!isset($nom)) $texte=$prenom;
else if(!isset($prenom)) $texte=$nom;
else $texte=$prenom."."$nom;

# création de l'image basée sur notre fond
$im = @imagecreatefromjpeg("fond-email.jpg");

# déclaration du bleu "UN"
$bleu = ImageColorAllocate($im, 0, 56, 105);

# police à utiliser
$fonte = "../fonts/verdanab.ttf";

# ajout du texte sur le fond
imagettftext($im, 12, 0, 10, 35, $bleu, $fonte, $texte);

#affichage de l'image
ImageJPEG($im);

#destruction de l'image en mémoire
ImageDestroy($im);
?>
```

utilisation :

```
<img
src="http://serveur.domaine.tld/adresse.php?prenom=trapwiki&nom=imgauto"
alt="adresse auto générée">
```

résultat :

- bonne protection
- peut être appliquée de manière automatique sur le site web de l'université

AJAX (Asynchronous Javascript and XML)

principe : utiliser une requête http asynchrone (en tache de fond après chargement de la page) pour afficher l'adresse email à partir d'un code utilisé comme identifiant de l'élément html devant contenir l'adresse.

```
<body onload="montrer_email()">
<script type="text/javascript" src="cacheremail.js"></script>
<span class="adresscachee" id="abelard-a"></span>
<span class="adresscachee" id="toto-j"></span>
```

Au chargement de la page, la fonction montrer_email() récupère tous les éléments dont la classe est “adressecachee” puis fait une requête http POST vers un script php, lui passant l'id de l'élément. Ce script php peut par exemple, utiliser l'id pour faire une requête LDAP dans l'annuaire pour retourner l'adresse email correspondant.

- bonne protection
- le navigateur du visiteur doit supporter les cookies, le javascript, etc.
- complexe à mettre en place
- plus d'information sur http://rajeczy.com/arpad/lib/index.php?p=misc/obfuscate_ajax

Formulaire

principe : ne pas rendre publique son adresse email et proposer à vos visiteurs un formulaire vous envoyant un mail

formulaire:

```
<form action="contact.php" method="post">
  <fieldset>
    <legend>Contactez moi</legend>
    <label for="nom">Votre nom</label>
    <input type="text" name="nom" value="" id="nom" size="30"><br>
    <label for="adresse">Votre adresse</label>
    <input type="text" name="adresse" value="" id="adresse" size="30"><br>
    <label for="message">Votre message</label>
    <textarea name="message" id="message" rows="5" cols="30"></textarea>
    <input id="submit" type="submit" name="submit" value="Envoyez">
  <fieldset>
</form>
```

script php:

```
<?
/* contact.php: envoie un mail */
mail ("votreadresse@votreserveur.com","contact web", $_POST["nom"]."
(".$_POST["adresse"].") vous envoie ce message:\n".$_POST["message"]);
?>
```

résultat:

Contactez moi
Votre nom
Votre adresse
Votre message Envoyez