Protection d'un serveur samba 4

Les instructions suivantes vous aideront à fournir à votre serveur Samba une protection contre les vulnérabilités de sécurité.
Même si vous faites les mises à jours, vous pourriez aimer les suggestions suivantes pour vous fournir des niveaux de protection supplémentaires.

Limiter le nombre de connexions simultanées

Samba est capable de limiter le nombre de connexions simultanées lorsque `smbd` est lancé comme un démon (et non depuis inetd).
L'option `max smbd processes` du fichier de configuration `smb.conf` dans la section `[global]`, ce paramètre limite le nombre maximum de processus `smbd` s'exécutant simultanément sur un système et est conçu pour empêcher la dégradation du service aux clients dans le cas où le serveur dispose de ressources insuffisantes pour gérer plus de ce nombre de connexions.

Exemple :

```
[global]
  ...
  max smbd processes = 1
  ...
```

Rappelez-vous que dans des conditions d'exploitation normales, chaque utilisateur aura un `smbd` associé à lui pour gérer les connexions à toutes les actions d'un hôte donné.
Toute autre tentative des clients de se connecter au serveur sera rejetée.

Cacher le répertoire dans les emplacements réseau

Le paramètre `browseable` permet de contrôler, si ce partage est vu dans la liste des partages disponibles dans une vue réseau.
Pour désactiver cela :

```
[sharefile]
  ...
  browseable=no
  ...
```

Masquer les sous répertoires et les fichiers des utilisateurs sans autorisations

Le paramètre `hide unreadable` empêche les clients de voir l'existence des fichiers qui ne peuvent pas être lus.

La valeur par défaut est désactivée.
Pour activer cela :

```
[sharefile]
  ...
  hide unreadable=yes
  ...
```

Utilisation de la protection basée sur l'hôte

Dans de nombreuses installations de Samba, la plus grande menace vient de l'extérieur de votre réseau immédiat.
Par défaut, Samba accepte les connexions à partir de n'importe quel hôte, ce qui signifie que si vous exécutez une version non sécurisée de Samba sur un hôte directement connecté à Internet, vous pouvez être particulièrement vulnérable.

L'une des corrections les plus simples dans ce cas est d'utiliser les options `hosts allow` et `hosts deny` dans le fichier de configuration `smb.conf` section `[global]` de Samba pour autoriser uniquement l'accès à votre serveur à partir d'une gamme spécifique d'hôtes.
Un exemple pourrait être:

```
[global]
  ...
  Hosts allow = 127.0.0.1 192.168.1.0/24 192.168.2.0/24
  Hosts deny = 0.0.0.0/0
  ...
```

Ce qui précède ne permettra que les connexions SMB de `localhost` et des deux réseaux privés 192.168.1 et 192.168.2.
Toutes les autres connexions seront refusées.
Le refus sera marqué comme une erreur.

Utilisation de la protection d'interface

Par défaut, Samba accepte les connexions sur toute interface réseau qu'il trouve sur votre système. Cela signifie que si vous avez une ligne RNIS ou une connexion PPP à Internet, Samba acceptera les connexions sur ces liens.
Ce n'est peut-être pas ce que vous voulez.

Vous pouvez modifier ce comportement en utilisant les options suivantes:

```
[global]
  ...
  interfaces = eth0 lo
  bind interfaces only = yes
  ...
```

Qui indique à Samba d'écouter seulement les connexions sur les interfaces avec un nom commençant par `eth0` et 'lo'.
Le nom que vous devrez utiliser dépend du système d'exploitation que vous utilisez.

Utilisation d'un pare-feu

Beaucoup de gens utilisent un pare-feu pour refuser l'accès aux services qu'ils ne veulent pas exposés en dehors de leur réseau.
Cela peut être une très bonne idée, bien que je recommande de l'utiliser avec les méthodes ci-dessus afin que vous soyez protégé même si votre pare-feu n'est pas actif pour une raison quelconque.

Si vous configurez un pare-feu, vous devez savoir quels ports TCP et UDP autoriser et bloquer.
Samba utilise les éléments suivants:

- UDP/137 - utilisé par nmbd
- UDP/138 - utilisé par nmbd
- TCP/139 - utilisée par smbd
- TCP/445 - utilisée par smbd

Voici, un exemple avec iptables :

```
iptables -A INPUT -p tcp -m multiport --dport 139,445 -j ACCEPT
iptables -A INPUT -p udp -m multiport --dport 137,138 -j ACCEPT
```

Utilisation d'un refus de partage IPC$

Si les méthodes ci-dessus ne conviennent pas, vous pouvez également placer un refus plus spécifique sur le partage `IPC$` qui est utilisé dans le trou de sécurité découvert récemment.
Cela vous permet d'offrir l'accès à d'autres actions tout en refusant l'accès à `IPC$` d'hôtes potentiellement non fiables.

Pour ce faire, vous pouvez utiliser:

```
[IPC$]
  Hosts allow = 192.168.1.0/24 192.168.2.0/24 127.0.0.1
  Hosts deny = 0.0.0.0/0
```

Cela indiquerait à Samba que les connexions `IPC$` ne sont pas autorisées à partir de n'importe où, sauf les sous-réseaux préciser dans l'option `hosts allow`.
Les connexions à d'autres actions seraient toujours autorisées.
Comme le partage `IPC$` est le seul partage qui est toujours accessible de manière anonyme, cela fournit un certain niveau de protection contre les attaquants qui ne connaissent pas un nom d'utilisateur/mot de passe pour votre hôte.

Si vous utilisez cette méthode, les clients recevront une réponse `Accès refusé` lorsqu'ils tentent d'accéder au partage `IPC$`.
Cela signifie que ces clients ne pourront pas parcourir les partages et ne pourront peut-être pas accéder à d'autres ressources.

Je ne recommande pas cette méthode à moins que vous ne pouvez pas utiliser l'une des autres méthodes énumérées ci-dessus pour une raison quelconque.

Mise à niveau de Samba

Bien sûr, la meilleure solution consiste à mettre à niveau Samba vers une version où le bug a été corrigé.
Si vous souhaitez également utiliser une des mesures supplémentaires ci-dessus, ce serait certainement une bonne idée.

Veuillez consulter régulièrement samba.org pour obtenir des mises à jour et des annonces importantes.

source: samba.org