

Sécurisez vos applications .Net : Partie 1

Sécurisez votre Serveur Windows pour vos applications ASP.Net

18/04/2004
Par Elise Dupont

Article paru sur 

L'intégralité des tutoriaux et des codes sources sont disponibles sur <http://www.dotnet-tech.com/tutotiels/>

Droit de diffusion:

L'ensemble ou partie de ce document ainsi que le code mis à disposition, ne peut être diffusé sur d'autres sites Web sans l'autorisation au préalable de son créateur.

Avant Propos :

Voici le premier article d'une série, vous expliquant (modestement) comment écrire, déployer et maintenir des applications .Net sécurisées. Le but est de vous donner le plus d'astuces et d'explications possibles pour vous permettre de limiter la surface d'attaque de votre serveur Web, et de votre application .Net. Si toutefois vous trouvez une aberration dans mes articles, merci de m'en faire part afin que je corrige mon erreur.

Ce premier document vous donne donc les grandes lignes les plus importantes pour sécuriser votre serveur web, ce n'est pas une liste exhaustive, il y a sûrement d'autres tâches à accomplir mais ce document vous donne au moins les points les plus importants. Ce document est basé sur Windows 2000, sachant que depuis Windows 2003 quelques modifications ont été apportées. Il

se concentre sur l'aspect « configuration du serveur », pour ce qui est de « comment développer une application sécurisée » c'est un autre sujet que je traiterai dans cette série d'articles prochainement.

Sommaire:

[1. Le système](#)

[2. Mise à jour et patches](#)

[3. IISLockDown et URLScan](#)

[4. Administrer les Services et limiter les Protocoles](#)

[5. Administrer les Comptes Utilisateur](#)

[6. Fichiers et Répertoires](#)

[7. Sécuriser l'application \(Authentification et limitation des protocoles\)](#)

[8. Audit et fichiers de traçage](#)

[Conclusion](#)

1. Le système

Mettre en place une procédure d'installation et de configuration du Système. Cette solution est préconisée mais peut toutefois ne pas être implémentée dans le cas où le Serveur Web est déjà existant et que l'on ne souhaite pas procéder à une réinstallation. Les recommandations les plus fréquentes sont :

- Installer le système sans IIS
- Installer ensuite les patches et mises à jour du système
- Installer IIS après l'installation du système
- Installer les patches pour IIS

2. Mise à jour et patches

2.1 Menace :

Beaucoup d'attaques sont causées par des publications de bulletin de sécurité exposant les failles découvertes. Dans la plus part des cas, quand une nouvelle vulnérabilité du système est découverte, le mode d'emploi pour exploiter cette vulnérabilité est lui aussi publié dans les heures qui suivent.

Si l'on ne patch pas et ne mets pas à jour le système fréquemment, cela augmente la vulnérabilité du Serveur exposé sur le Web.

2.2 Contre-mesures :

La toute première action à mener est d'activer le système de mise à jour automatique afin que le serveur soit vulnérable le moins longtemps possible. Microsoft fournit un outil de diagnostic, MBSA ([Microsoft Baseline Security Analyzer](#)) qui analyse et détecte les failles de sécurité et les erreurs de configuration sur un serveur, pour les produits suivants :

- Windows (NT4, 2000, XP, 2003)
- IIS
- SQL Serveur
- Internet Explorer
- Office

Cet outil scanne aussi les mises à jour de sécurité manquantes, et permet ainsi de dresser une liste des actions à mener pour configurer correctement le système. Cet outil gratuit n'est à l'heure actuelle disponible que pour

les systèmes anglais.

3. IISLockDown et URLScan

3.1 Menace :

Par défaut, le système ouvre un grand nombre de services, de ports, de protocoles, qui s'ils ne sont pas nécessaires devraient être verrouillés afin de limiter la « surface d'attaque ». En effet des nombreux services et protocoles sont réputés pour ne pas être sûrs (comme le protocole FTP par exemple). Depuis Windows 2003, beaucoup moins de services sont ouverts par défaut, mais ce n'est pas de la même façon pour Windows 2000.

3.2 Contre-mesures :

Microsoft fournit deux outils gratuits, IISLockDown (fourni par défaut dans IIS6) et URLScan. IISLockDown est un outil qui permet d'automatiser certaines phases de sécurisation. Cela réduit grandement la vulnérabilité d'un serveur. Il fournit de nombreux modèles de sécurisation basés sur le type de Serveur (Serveur Web uniquement, Serveur de Base de données...) Les modèles désactivent des fonctionnalités du serveur dites « à risque » ou sécurise ces fonctionnalités si elles sont nécessaires. En addition, IISLockDown installe URLScan. URLScan permet à l'administrateur de restreindre les types de requêtes HTTP auquel le serveur peut répondre, en se basant sur un ensemble de règles que l'administrateur peut contrôler. En bloquant des requêtes HTTP spécifiques l'outil permet d'empêcher d'éventuelles requêtes externes dangereuses d'atteindre le serveur.

4. Administrer les Services et limiter les Protocoles

4.1 Menace :

Les services sont les premières vulnérabilités d'un serveur, qui peuvent être exploitées pour pénétrer un système, installer ou exécuter des virus, des vers, voire des Chevaux de troie. De plus il existe il faut se prémunir des attaques de type Déni de service, qui consistent à paralyser temporairement des serveurs afin qu'ils ne puissent être utilisables (Introduction au Denial Of Service)

4.2 Contre-mesures :

Il est fortement conseillé de fermer tous les services et protocoles « à risque » ou « non nécessaires » au fonctionnement des applications. Cela s'effectue en plusieurs points :

- Désactiver les services non utiles (en général il s'agit du Navigateur, de Messenger, du Netlogon, de Telnet, des TCP/IP Services Simples, et du Spooler).
- Désactiver les protocoles FTP, SMTP, NNTP.
- Désactiver le protocole WebDAV, NetBIOS et SMB
- Renforcer la pile TCP/IP (en configurant divers paramètres au niveau de la base de registre afin de protéger le serveur (au niveau réseau) contre des attaques telles que SYN, IEMP, SNMP, et de protéger le système ADF.SYS). Cette étape permet d'éviter ce que l'on appelle communément les "denial of service" ou déni de service. De nombreux articles à ce sujet sont disponibles sur le net, notamment : Harden the TCP/IP Stack for Denial of Service Attacks (Windows 2000/XP) et HOW TO: Harden the TCP/IP Stack Against Denial of Service Attacks in Windows Server 2003

5. Administrer les Comptes Utilisateur

5.1 Menace :

Une mauvaise gestion des comptes peut permettre à un utilisateur non autorisé entrant dans le système, de bénéficier des privilèges associés au compte utilisateur qui est en session sur le Serveur. Et donc, en fonction des droits données, pourrait éventuellement exécuter des programmes non autorisés.

5.2 Contre-mesures :

Il est fortement conseillé de :

- Supprimer tous les comptes inutilisés
- Désactiver le compte « Guest »
- Renommer le compte de l'administrateur s'il se nomme « Administrateur »
- Désactiver le compte IUSR
- Utiliser le principe des « moindres privilèges » (The Challenge of Least Privilege) pour le compte qui sera utilisé afin de tourner l'application sur le serveur. Cela permettra de réduire la portée de dommages en cas d'intrusion.
- Restreindre les accès à distance
- Désactiver les Sessions Nulles, qui permettent à une personne d'ouvrir une session anonyme sur le serveur.

6. Fichiers et Répertoires

6.1 Menace :

Les attaques de type « Répertoires transversaux », permettent à un attaquant d'exécuter des programmes systèmes et utilitaires. De même les partages représentent une menace.

6.2 Contre-mesures :

Il est fortement conseillé de :

- Désactiver l'option « Chemin parent » dans IIS.
- Placer le dossier de l'application Web Service sur une partition autre que la partition système, ce qui permet d'éviter d'atteindre des répertoires systèmes aux niveaux supérieurs.
- Supprimer tous les partages de fichier et répertoire non nécessaires, en incluant aussi les partages par défaut de la machine.

7. Sécuriser l'application (Authentification et limitation des protocoles)

7.1 Menace :

Une fois le serveur sécurisé, il reste peu de chances de laisser une vulnérabilité ouverte. Cependant on peut supposer qu'en cas d'intrusion sur le Serveur, il est préférable de ne pas laisser la chaîne de connexion au serveur SQL sous un format lisible. Enfin on peut aussi envisager de limiter au niveau applicatif les requêtes HTTP en supprimant les requêtes de type HTTP Get et HTTP Post.

7.2 Contre-mesures :

Il est fortement conseillé de :

- Stocker la chaîne de connexion au serveur SQL sous un format encrypté
- Désactiver les protocoles HTTP Get et HTTP Post au niveau du serveur d'application.

8. Audit et fichiers de traçage

Dans le cas où il y a malgré tout eu une intrusion sur votre serveur, le seul moyen de comprendre et d'analyser la faille afin de la réparer, est d'avoir gardé des traces sur votre serveur. Il est donc fortement conseillé de conserver des logs et des audits de chaque action faite, du type accès la base de données par exemple. Cette étape ne vous aidera pas à sécuriser votre serveur, mais à agir en cas d'attaque.

En général il faut conserver une trace de :

- chaque tentative d'authentification qui a échoué
- toute action sur le système qui a échoué

Conclusion

Je n'ai évidemment pas cité TOUT ce qu'il faudrait faire, car on pourrait en faire quasiment un livre, mais si déjà vous respectez ces principes de base, cela devrait limiter la surface d'attaque. Evidemment si quelqu'un veut vraiment pénétrer votre système et a les connaissances requises, il a de fortes chances d'y arriver, c'est pourquoi il ne faut pas négliger le dernier point que j'ai exposé. Mon article n'a pas aussi exposé comment effectuer toutes ces tâches, de peur de vous inonder de détails, mais si toutefois vous êtes intéressé sur les démarches à suivre, n'hésitez pas à me contacter, et je vous conseille vivement aussi de vous procurer le livre « Improving Web Application Security » dans la collection « Patterns & Practices » chez Microsoft.

Voir aussi dans la même série :

- [Partie 1 : Sécurisez votre Serveur Windows pour vos applications ASP.Net](#)
- [Partie 2 : Les règles de base](#)



L'ensemble ou partie de ce document ainsi que le code mis à disposition, ne peut être diffusé ailleurs sans autorisation préalable

elise.dupont@europe.com - www.dotnet-tech.com - 2004