HERVÉ SCHAUER CONSULTANTS
Cabinet de Consultants en Sécurité Informatique depuis 1989
Spécialisé sur Unix, Windows, TCP/IP et Internet

Journées CFSSI

Sécurité et SGBD

Nicolas Jombart <Nicolas.Jombart@hsc.fr>
Alain Thivillon <Alain.Thivillon@hsc.fr>

Plan

- Les SGBD et la sécurité
- Problèmes dans quelques SGBD bien connus
  - Oracle
  - MS-SQL Server
  - MySQL
- Recommandations
- Exemples d'attaques applicatives
  - DOS
  - Injection SQL
  - Démonstrations (Annuaire, PHPNuke)
  - Contres mesures
- Conclusion

Les SGBD dans la sécurité

- Budgets Sécurité
  - Vont d'abord à l'achat de système de sécurité (firewalls, IDS, ...)
  - À la formation
  - À la sécurisation des applications
  - ☞ le SGBD est le parent pauvre de la sécurité
- Complexité
  - Les BD sont une affaire de spécialistes:
    - au niveau de leur gestion : DBA
    - au niveau de la programmation
  - On ne peut pas sérieusement faire de l'Oracle deux fois par an
  - Quand c'est le cas, la sécurité est encore pire !

Les SGBD dans la sécurité - 2

- Rôle du DBA
  - Maintenir le SGBD
  - Gérer les comptes, les applications, ...
  - Pas de formation sécurité : ne peut pas « imaginer » les attaques possibles
- Mises à jour des systèmes
  - D'expérience, 80% des serveurs de BD meurent avec le système et le SGBD initial: (Informix 7.2, Oracle 7.2, ...)
  - « If it works, don't fix it »
  - Conséquence : de nombreuses failles système et applicatives ne sont JAMAIS corrigées, surtout sur les réseaux internes
- Criticité des applications
  - Arrêts impossibles
  - La sécurité passe en dernier

Les SGDB dans la sécurité - 3

- Le SGBD est souvent un composant
  - Installé par ou avec un logiciel tiers
    - ERP (SAP, Lawson)
    - DataMining
    - Gestion de parc (SMS, ...)
    - Pour SQL Server, voir par exemple (223 Applications recensées) : http://www.sqlsecurity.com/DesktopDefault.aspx?tabindex=10&tabid=13
  - Géré via ce logiciel tiers
  - Dans une version "croupion" (MSDE)
  - Dans un mode d'installation par défaut
- ☞ Sa configuration de sécurité est bien souvent encore plus obscure !
  - Et personne ne veut/peut prendre la responsabilité de modifier le paramétrage

Les risques

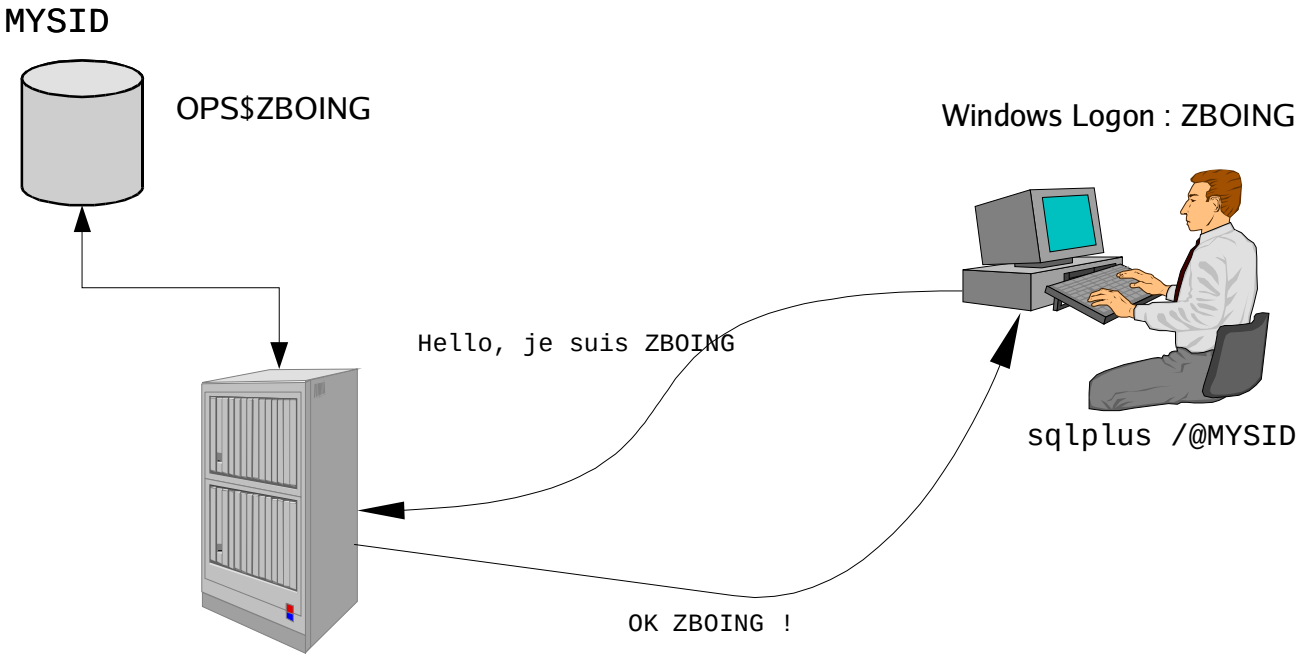
- Attaques sur le SGBD lui même
  - Failles connues classiques (buffer overflows, bugs d'authentification, ...)
  - Failles dans les applications associées: serveurs Web d'administration, démons snmp, programmes setuid root installés par le SGBD, ...
- Mauvaises configurations
  - Modes d'authentification dégradés (.rhosts, ...)
  - Mots de passe par défaut
  - Fichiers de la BD non sécurisés (lecture par tous)
- Interception de mots de passe
  - Par écoute du réseau
  - Par lecture de fichiers de configuration sur disque

Les risques - 2

- Attaques sur les applicatifs
  - Injection SQL sur les applications Web
  - Détournement des requêtes effectuées par un ERP
  - Autorisations trop larges
- Attaques sur l'OS via le SGBD
  - Ecriture/lecture de fichiers, exécution de commandes
    - La base de données tourne avec des privilèges différents
    - Contournement de la politique de sécurité
      - 'safe_mode' de PHP
      - chroot
  - Critique chez les hébergeurs Web mutualisés
    - `load data infile '/web/data/a/anotheruser/db.param' INTO hack ...`
    - `select into outfile '/web/data/a/anotheruser/...' FROM hack`

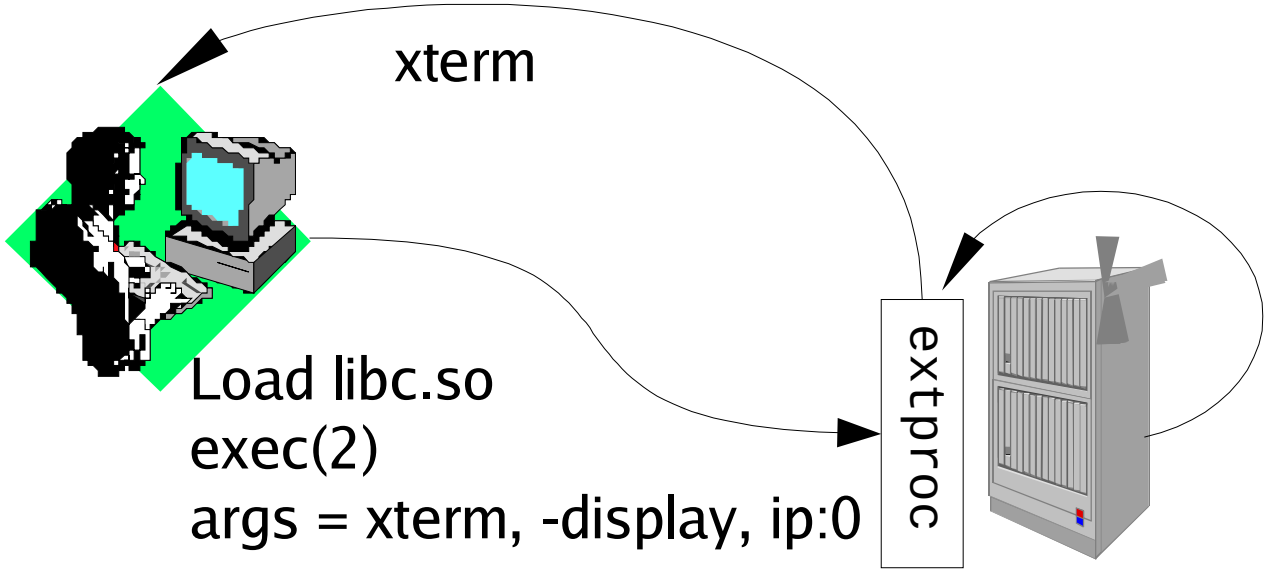
Exemples : Oracle - 1

- Problème le plus évident: utilisation d'une fonctionnalité dangereuse: « `remote_os_authent` »
  - Oracle fait confiance au client pour authentifier l'utilisateur
  - S'il envoie ZDOING, il est transformé en OPS$ZBOING (pas de mot de passe)
  - C'est la configuration par défaut des installations SAP !



Oracle - 2

- EXTPROC
  - C'est un service Oracle que le Listener connaît dans les installations par défaut
  - Sert à Oracle pour exécuter des procédures stockées dans des .DLL (Windows) ou des .so (Unix) (c'est une sorte de RPC Oracle)
  - Il est possible de l'utiliser sans s'identifier ☞ possibilité d'exécuter du code à distance avec les privilèges d'Oracle



Oracle - 3

- Multiples vulnérabilités dans IAS (Serveur HTTP Oracle 9i)
  - Version ancienne d'Apache
  - Débordement de buffer dans le module PL/SQL
  - Double décodage : `http://oracleserver/pls/dadname/admin_/help/..%255Cplsql.conf`
  - Exécutions de commandes SQL via OWA_UTIL
    `http://oracleserver/pls/dadname/owa_util.cellsprint?p_theQuery=select * from sys.dba_users`
  - ...
- Débordement de buffer TNS Listener ...
- comptes et mots de passe par défaut (SCOTT/TIGER, SYS/CHANGE_ON_INSTALL, SYSTEM/MANAGER, ...)
- Il faut suivre les correctifs :
  - http://technet.oracle.com/deploy/security/alerts.htm

MS-SQL Server

- Produit complexe
  - Tourne avec les droits LocalSystem
  - Application des Service Pack parfois douloureuse
- Deux modes d'authentification
  - NT Only : authentification sur le domaine NT/2000
  - Mixed-Mode : idem + comptes internes. Attention potentiellement tous les utilisateurs du domaine ont accès au serveur !
- Jusqu'à SQL2000, compte « SA » sans mot de passe par défaut à la création du système
  - Des centaines de serveurs compromis par ce biais
  - Utilisation de nombreuses procédures externes, dont xp_cmdshell ☞ compromission totale du serveur

MS-SQL - 2

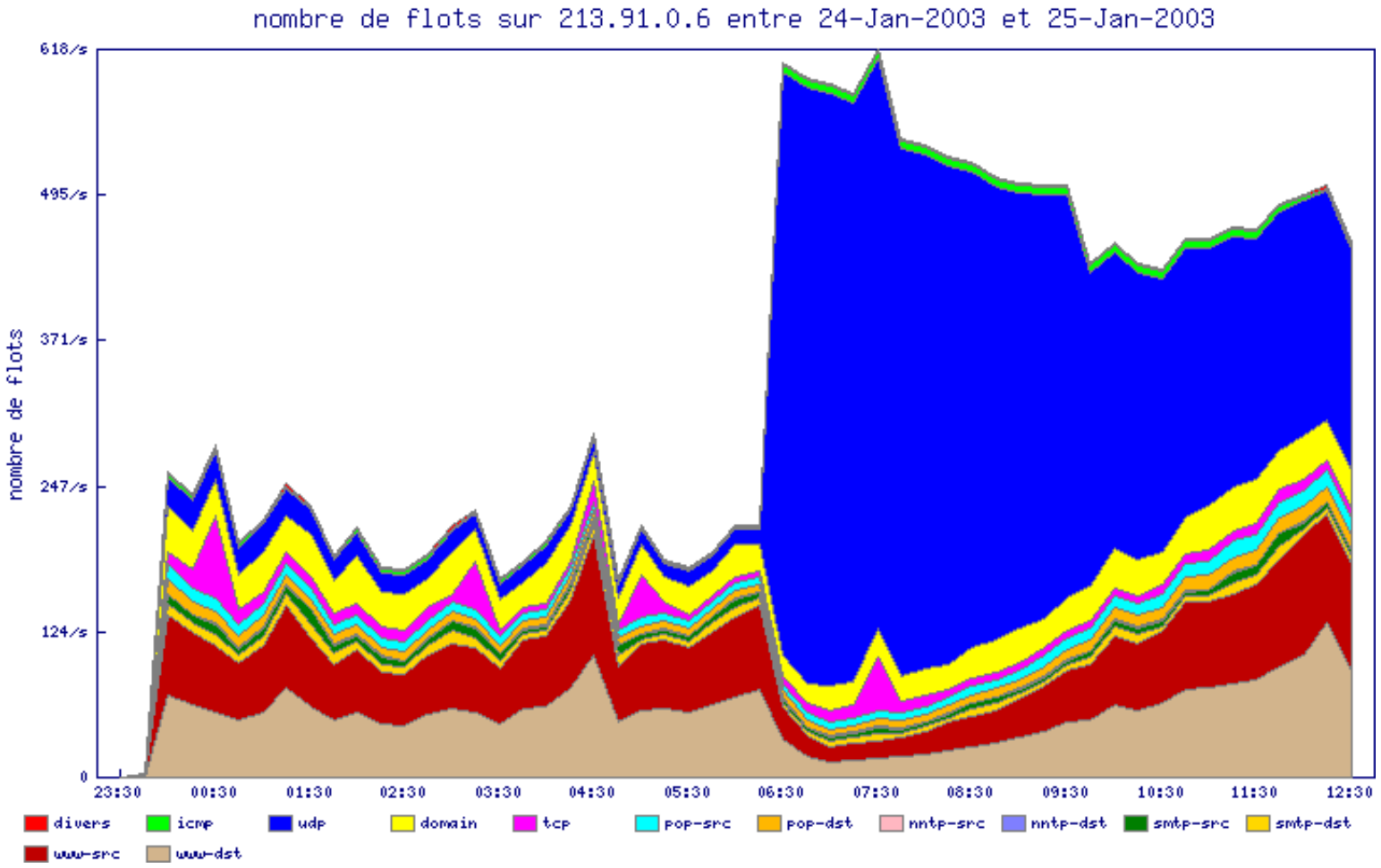
- Ver SQLSlammer
  - Utilise le service de localisation des serveurs (UDP/1434)
  - Basé sur bug découvert par NGSSoftware/D.Lichtfield
    - nombreuses erreurs de programmation
    - buffer overflow, heap overflow, déni de service réseau
  - Le ver se renvoie par UDP
    - ça consomme peu de ressources systèmes (pas besoin de rouvrir un socket à chaque paquet)
    - pas d'attente, pas de ACK
  - ☞ Déni de service réseau immédiat sur les infrastructures des hébergeurs

```
"I'm working on it for some friends, and I'm seeing about 900mbits/second on a
gigabit link coming out of their hosting facility. "
```

  - Deux mois après: ...

```
Mar 24 18:41:10 ng0 @0:24 b 69.22.18.6,1168 -> 217.128.134.100,1434 PR udp len 20
404 IN
```

MS-SQL - 3



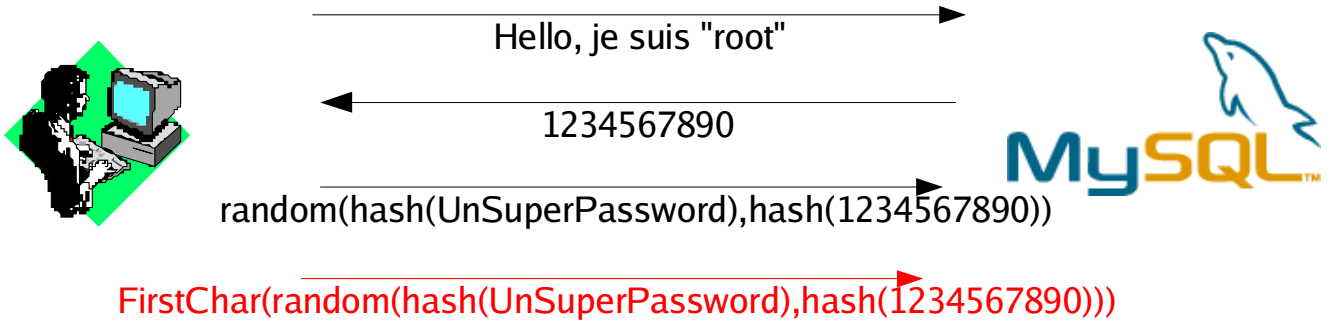
Sur un /17 (32768 adresses IP)

MS-SQL - 4

- Problème dans l'authentification (débordement de buffer)
  - http://www.securityfocus.com/bid/5411
  - (Aout 2002)
- Passage du mot de passe en clair pour les logins non NT
  - Simple XOR avec 0xA5
- Débordement de buffer/de tas dans les procédures externes
  - Un seul octet à écraser et le système de privilèges est ignoré
- Procédures dangereuses
  - xp_readerrorlog : lecture de fichiers
  - xp_regread : lecture de la registry
- SQL Agent Job : submission de jobs, permet de monter les privilèges

MySQL - 1

- SGBD "Light" et facile à mettre en place
  - Très très utilisé dans les « petits » sites
  - Système de privilèges mais fonctions manquantes (vues notamment)
- 2000 : gros problème d'authentification
  - Dans la phase d'auth, le serveur ne vérifie que le nombre de caractères envoyé par le client : avec 32 essais il est possible de compromettre n'importe quel compte
  - Ressurgit en 2002 dans COM_CHANGE_USER (changement d'identité)



Mysql - 2

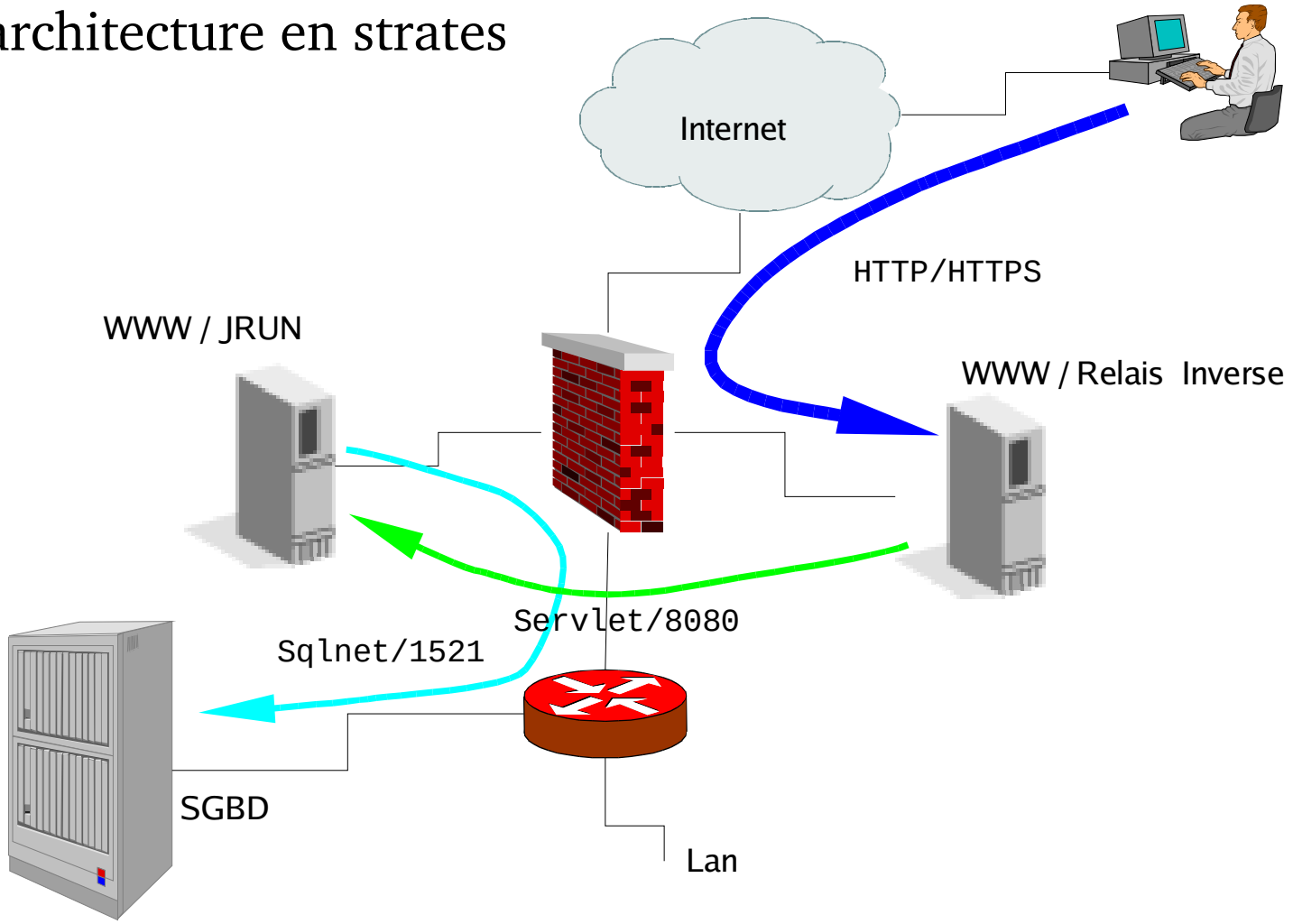
- Mauvaise génération du challenge
- 2001: Buffer Overflow dans SELECT
- Plusieurs corruptions de mémoire dans certaines fonctions
- Lecture du fichier de configuration dans $DATADIR/my.cnf
  - Tous les utilisateurs ayant le privilèges 'FILE' peuvent écrire dans ce répertoire
  - ☞ changement possible des paramètres de démarrage du serveur
- Fonctions intéressantes dans MySQL
  - Association (host,password) : limitation des adresses IP Sources
  - Limiter les adresses en écoute (127.0.0.1)
  - Utilisation locale de sockets Unix : plus de réseau

Contre mesures - 1

- Ne jamais exposer un serveur BD sur Internet
  - S'il y a besoin d'accès directs au SGBD, il faut utiliser des tunnels, un firewall avec authentification forte ouvrant le flux, ...
  - Il faut etre sûr de filtrer les ports SGBD (1521, 1527, 3306, 1434, 135 ...)
  - Attention aux hébergeurs pas chers !
- Ne jamais partager un serveur BD
- Durcissement de l'OS
  - Va limiter l'exploitation de failles
  - Appliquer les procédures classiques
    - non installation des composants annexes
    - limitation des services réseaux
    - application régulière des correctifs de sécurité
    - signature du système

Contre mesures - 2

- Appliquer le principe de défense en profondeur et de cloisonnement par une architecture en strates



Contre mesures - 3

- Durcissement installation SGBD
  - Changer les mots de passe par défaut (Oracle)
  - Supprimer les comptes par défaut
  - Ne pas installer les exemples, les applications annexes, ..
- Séparation des privilèges
  - Recenser les rôles dans l'application (admin, mise à jour, lecture, ...)
  - Appliquer ces rôles dans les privilèges attribués
- Audit des applicatifs
  - Parler sécurité avec les développeurs (SQL Injection, XSS, débordements de buffer, validation d'entrées ...)
  - Recherche des points critiques, des flux utilisateurs, ...
  - Audit des sources Java, ASP, PHP, Perl, C , ...

Déni de service

Objectifs de l'attaquant :
prendre de la ressource sur la base de données

- Demander un nombre trop grand d'enregistrements
  - Jointures mal définies
- Chercher à prendre de la ressource (CPU) et de la mémoire
  - moteur de recherche
  - exemple des robots (Google, dir.com)

Injection SQL

Principe : Insérer du code SQL forgé dans une requête construite dynamiquement pour modifier son comportement

Application utilisant une entrée de l'utilisateur (nom, n°, information, etc.) :

- Construction de la requête
- Exécution dans la base de données associée à l'application
- Traitement/présentation de la réponse

Injection SQL : méthodes

Plusieurs méthodes :

- Modifier le comportement

  SELECT * FROM table WHERE login='hsc' or 'x'='x'

- "Coller" deux requêtes

  SELECT * FROM table WHERE login='hsc' ; SELECT * FROM passwords;

- Utilisation de fonctions du SGBD
  - master.xp_cmdshell (MS-SQL)
  - |shell("ping.exe 10.20.30.40")| (MS-SQL)
- Utilisation de UNION, ou de sous-requêtes
- ...

Le problème

- Login/Mot de passe entrés dans un formulaire
- Un script en perl

```
$name = $query->param("name");
$password = $query->param("pass");
$sth = $dbh->prepare("SELECT password WHERE name=$name");
$sth->execute();
$row = $sth->fetchrow_hashref();
if($$row{"password"} eq $password) {
   # Ok ...
} else {
   &login_error;
}
```

- $name peut contenir un nom plus des caractères de contrôle
  - ☞ Il est donc possible de modifier la requête en y ajoutant des commandes

Injection SQL : autre exemple

- Vulnérabilité IMP (Webmail), 8 janvier 2003

```
$sql="select username from $default->db_pref_table
  where username='$user@$server'";


http://webmail.server/imp/mailbox.php3
  ?actionID=6&server=x&imapuser=x';somesql+--&pass=x"
```

Injection SQL : écueils

- Quotes

  Il est souvent nécessaire de commencer une injection en fermant l'apostrophe (**"** ou **'**)

- Lors d'une insertion (variable située en milieu de requête)

  Il faut supprimer la fin en utilisant les commentaires (**-- /*** etc.)

- Il est souvent nécessaire d'encoder les caractères dans l'URL
  - **%20** pour espace
  - Ou double-encodage : **%2520**
- Cependant les possibilités et méthodes d'injection dépendent :
  - Du SGBD utilisé
  - Du driver utilisé pour la connexion au SGBD (limitations supplémentaires)
- On ne voit pas toujours les résultats (couche de présentation)

Injection SQL : messages d'erreur

Certaines applications renvoient à l'utilisateur le message d'erreur du driver utilisé pour se connecter.

Attaque par tâtonnements :

- **http://webserver/script.asp?id=0--%20%28SELECT%20*%20FROM%20table%29**

```
Microsoft OLE DB Provider for ODBC Drivers error '80004005'
[Microsoft][ODBC Microsoft Access Driver] You have written a subquery that can
return more than one field without using the EXISTS reserved word
in the main query's FROM clause. Revise the SELECT statement of the subquery
to request only one field.

/script.asp, line 15
```

Injection SQL : UNION

Trouver le nombre de colonnes d'une table

Avec un nombre incorrect :

```
http://webserver/script.asp?id=1%20UNION%20SELECT%20*from%20table

Microsoft OLE DB Provider for ODBC Drivers error '80004005'%
[Microsoft][ODBC Microsoft Access Driver] The number of columns in the two
selected tables or queries of a union query do not match.
/script.asp, line 15
```

Avec un nombre correct :

```
http://webserver/script.asp?id=1%20UNION%20
SELECT%20a,%20,%20b,%20c,%20d,%20e,%20f,%20g,%20h,%20i,%20j*from%20table

Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
[Microsoft][ODBC Microsoft Access Driver] The SELECT statement includes a
reserved word or an argument name that is misspelled or missing, or the
punctuation is incorrect.
/script.asp, line 15
```

Contre-mesures

- Valider les entrées
  - Ne protégera pas que des problèmes d'injection SQL
- Utiliser les fonctions spécialement conçues du driver
  - Exemple avec perl DBI : $dbh->quote(); mysql_escape_string(), etc.
  - Attention, ça ne règle pas tout : il devient possible d'insérer du code SQL dans la base elle-même (par exemple, création d'un compte admin'--)
  - Problème des entiers : vérifier que les entiers sont bien numériques ...
- Ne donner à l'utilisateur (SGBD) de l'application que les droits nécessaires
  - Par exemple SELECT dans le cas d'une simple consultation
  - SELECT, INSERT et UPDATE dans la plupart des cas

Conclusion

- Les SGBD sont:
  - Complexes
  - Leur sécurité n'est pas toujours maîtrisée
- Les risques sont réels et parfois ignorés
  - Expériences HSC
- Il faut sensibiliser la chaîne:
  - Appels d'offre/cahier des charges
  - Développeurs
  - Recettes
  - Dba, Administrateurs, Réseau, ...
- Questions ?

Références

- NGSSoftware : http://www.nextgenss.com/
- SQLSecurity : http://www.sqlsecurity.com
- About.COM: http://databases.about.com/cs/security/
- Sans: http://www.sans.org/rr/win/SQL_sec.php