INSA – 5° R.T.

Sujet d'examen
26 janvier 2007

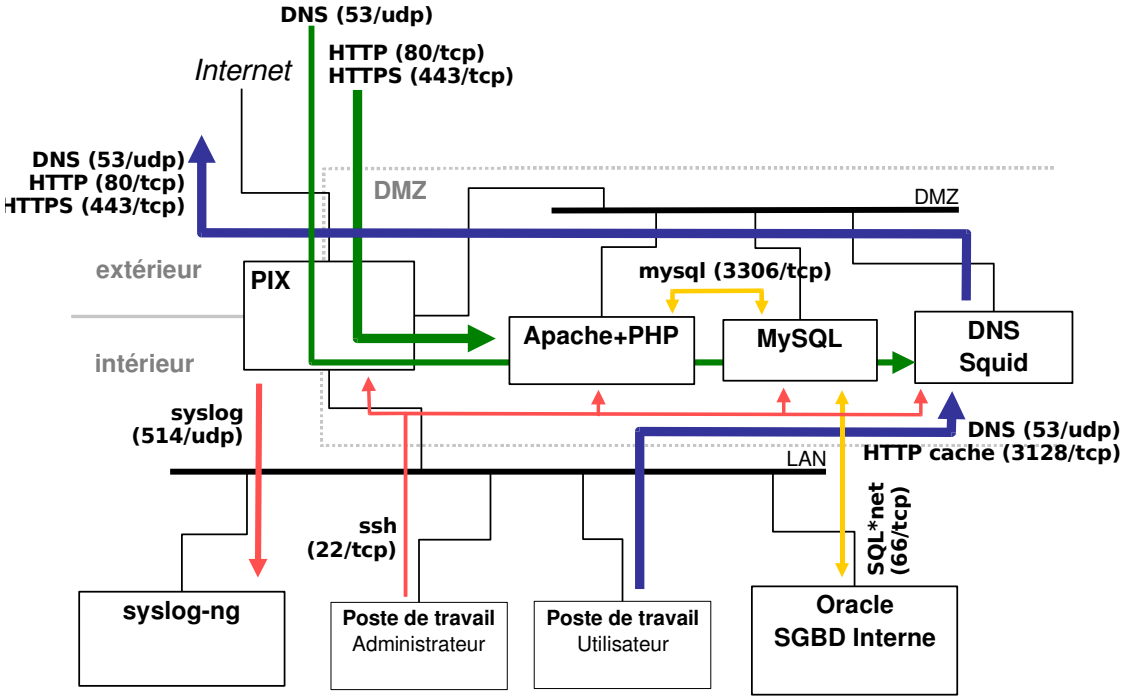
Sécurité des systèmes informatiques

2ème partie

Exercice 1 *(2,5 points)*

On vous propose un poste parmi l'équipe informatique d'une entreprise. Vous êtes responsable des serveurs Web ainsi que de la sécurité informatique en général. Vous êtes directement subordonné au chef de l'équipe informatique. A quels problèmes pouvez-vous vous attendre dans ces conditions ?

Exercice 2 *(4 points)*



Question 1 *(2 points)* : Compte tenu du mode de fonctionnement suggéré par le schéma, expliquez le fonctionnement des différents services réseau fournis à l'organisation utilisant cette architecture.

Question 2 *(2 points)* : Critiquez (avantages/inconvénients) l'architecture utilisée (notamment en terme de sécurité). Proposez des évolutions de l'architecture permettant de pallier certains inconvénients et éventuellement précisez les nouveaux inconvénients associés à vos préconisations. Quel est votre avis sur le niveau général de sécurité offert par cette architecture ?

Exercice 3 *(3 points)*

Dans l'utilisation courante du Web, on peut être amené à remplir des formulaires en ligne. Les données de ces formulaires sont alors envoyées au serveur et traitées, par exemple, par un programme CGI côté serveur. Les langages les plus utilisées jusqu'à présent pour écrire des programmes CGI sont Perl, PHP, C, ASP ou même des Shell.

Voici ci-dessous un extrait de page HTML simple permettant de communiquer son adresse de messagerie électronique à un serveur afin de recevoir des informations commerciales.

```
<TABLE align="center">
      <TR><TD align="center">
      Pour obtenir plus de renseignements sur nos produits, merci
      de nous fournir votre email :</TD></TR>
      <TR><TD align="center">
            <FORM ACTION="/cgi-bin/getmail.pl" METHOD=POST>
            <INPUT TYPE="text" NAME="email"><BR>
            <INPUT TYPE=SUBMIT VALUE="Envoyer">
            </FORM>
      </TD></TR>
</TABLE>
```

Le programme CGI écrit en Perl qui traite ces données est présenté ci-dessous. Ce programme s'active au moment où le bouton « Envoyer » du formulaire présent dans le code HTML est cliqué. Ici, ce programme est simplifié et se limite à émettre un message électronique acquittant la demande à destination de l'adresse fournie.

```
#!/usr/bin/perl
use CGI;

my $q = new CGI;
my $adresse = $q->param("email");

open MAILPRG, "| /usr/lib/sendmail $adresse";
print MAILPRG "To: $adresse \n";
print MAILPRG "From: Dupondt et fils\n\n";
print MAILPRG "Nous avons bien reçu votre demande. Vous recevrez";
print MAILPRG "notre documentation d'ici quelques jours.\n";
close MAILPRG;

print "Content-Type: text/html\n\n";
print "<HTML>";
print "<P align=\"center\"><A href=\"/index.html\">Retour au ";
print "sommaire</A></P>";
print "</BODY>";
print "</HTML>";
```

1. Les possibilités d'action d'un pirate semblent restreintes puisqu'il ne peut que remplir le champ du formulaire. Intuitivement, que peut-il tenter ?
2. Comment peut-il, par exemple, se faire envoyer par courrier électronique le fichier des mots de passe (`/etc/passwd` pour simplifier) du serveur HTTP ?

Précisions concernant le langage Perl :

- `$variable` permet de récupérer le contenu d'une variable. On peut directement utiliser cette syntaxe à l'intérieur d'une chaîne de caractère, comme dans : `"La valeur de la variable est: $variable"`.
- L'utilisation d'un *pipe* `|` avec la fonction `open` permet d'exécuter un programme externe à la manière d'un script *shell*, et de récupérer un descripteur de fichier permettant d'accéder à l'entrée standard du programme exécuté. Ci-dessus, c'est la commande Unix `sendmail` standard qui est lancée pour émettre le message.

Exercice 4 *(2,5 points)*

Le protocole SSH permet de remplacer l'authentification classique d'une session distante avec des mots de passe par une authentification à clef publique. Pour cela, l'utilisateur installe une copie de sa clef publique dans le compte cible sur la machine exécutant le serveur SSH. Sur le client, l'utilisateur dispose d'une copie de sa clef privée dans un fichier qui n'est pas protégé par un mot de passe mais dont les droits d'accès sont restreints à l'utilisateur. Lors de l'authentification, le serveur utilise la clef publique dont il dispose pour vérifier que le client qui se connecte dispose bien de la clef privée correspondante. Si c'est le cas, le serveur accepte la connexion sans demander de mot de passe. (Dans tous les cas, avec ce protocole, les communications entre le client et le serveur sont systématiquement chiffrées.)
Quels sont, du point de vue de la sécurité, les avantages et les inconvénients d'une authentification qui ne demande pas de mot de passe ?