4/10 (19/15)

Département informatique
Faculté Electronique et Informatique
USTHB

Le 15/10/2014
concours Doctorat LMD

Epreuve de Sécurité Informatique
Durée : 2h Documents non autorisés

Partie 1 : cryptographie

Falsification existentielle du schéma de signature ElGamal
Soit p un nombre premier, g un générateur de Z_p^* et une clé publique $y = g^x \in Z_p^*$ où x mod(p−1) est la clé secrète. L'espace des messages est $M_n = Z_p^*$.
La signature du message $m \in Z_p^*$ est le couple (r, t) tel que $0 \leq r, t < p - 1$ et qui satisfait l'équation $g^m = y^r r^t \bmod p$.
L'algorithme de signature génère tout d'abord $k \in Z_p^*$ aléatoirement tel que pgcd(k, p− 1) = 1. Puis, il calcule $r = g^k \bmod p$ et $t = (m - xr)/k \bmod (p - 1)$.

1. Vérifier que la vérification fonctionne.
2. Montrer une falsification existentielle sur ce schéma.
3. Montrer que ce schéma peut être cassé si le même random k est utilisé pour calculer la signature de deux messages différents m_0 et m_1.
4. Est-ce que ce schéma est sûr si on prend k, k + 1, k + 2, . . . où k est une valeur aléatoire pour signer des messages différents $m_1, m_2, \ldots$?

5/10 (14/15)

Département informatique
Faculté Electronique et Informatique
USTHB

Le 15/10/2014
concours Doctorat LMD

Partie 2 : Sécurité Base de données

Soit un extrait d'une BD relative à la révision périodique des pièces de machines industrielles :

Pièce (Code-Pièce, Nom, Materiel, Fournisseur, Etat-pièce)

Technicien (Code_Tech, Nom, Prénom)

Vérification (Code-Pièce, Code_Tech, Date, Observation)

Un technicien externe à la société (sous-traitant) effectue des tâches de maintenance sur chaque pièce à des dates programmées dans l'année. A la fin de chaque intervention, le technicien saisi ses observations dans la table Vérification et mis à jour l'attribut Etat-pièce si la pièce est défectueuse.

Vous êtes Administrateur de cette BD et vous devez donner l'accès à la BD pour ce technicien. Cet accès est soumis aux contraintes suivantes :

a. Au bout de 4 tentatives de connexion non réussies, le compte est bloqué
b. La durée de l'intervention sur la BD ne doit pas dépasser 30 minutes.
c. Le technicien ne peut créer plus d'une session sur la BD.
d. Le mot de passe expire dans 2 jours.

1. Créer un utilisateur technicien
2. Imposer les restrictions citées en a. à d. pour cet utilisateur.
3. Considérons les trois scénarios suivants :
   - Le technicien n'a que le droit de consulter pour chaque code pièces, les cinq dernières dates de vérification par ordre chronologique.
   - Le technicien peut voir le contenu de toutes les tables de la BD (même celles non représentées ici).
   - Le technicien peut seulement modifier les deux attributs 'Etat-pièce' et 'Observation'.
4. Quels sont les mécanismes de sécurité vous permettant de limiter son intervention sur la BD dans chaque scénario.
5. Donnez les requêtes utilisées dans chaque scénario.
6. Si vous recevez plusieurs techniciens durant l'année. Comment complétez-vous la solution précédente afin d'assurer le même degré de sécurité. Donnez les requêtes correspondantes.
7. Vous développez une application web via laquelle le technicien puisse saisir ses observations à distance.
   - Citez deux types de menaces potentiels sur la BD dans ce type d'application.
   - Comment vous améliorez la protection contre ces menaces.

6(14/15)
10

Département informatique **Le 15/10/2014**
Faculté Electronique et Informatique **concours Doctorat LMD**
USTHB

Partie 3 : Sécurité Système

1. Pour chacun des codes suivants, précisez en justifiant vos réponses:
   a. Son rôle.
   b. Les vulnérabilités présentes, si elles existent.
   c. Les modifications à apporter au code afin d'éliminer ces vulnérabilités

i. Code 1

```
1.  void main(){
2.        char inbuf[40];
3.
4.        fgets(inbuf, 40, stdin);
5.        printf("%s \n", inbuf);
6.  }
```

ii. Code 2

```
1.  void main(){
2.        char inbuf[40];
3.
4.        fgets(inbuf, 40, stdin);
5.        printf(inbuf);
6.  }
```

i. Code 3

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  #include <stdlib.h>
4.
5.  int main(int argc, char *argv[]){
6.        unsigned short s;
7.        int i;
8.        char buf[100];
9.
10.       if(argc < 3) exit(-1);
11.
12.       i = atoi(argv[1]);
13.       s = i;
14.
15.       if(s >= 100) exit(-2)
16.
17.       printf("s = %d\n", s);
18.
19.       memcpy(buf, argv[2], i);
20.       buf[i] = '\0';
21.       printf("%s\n", buf);
22.
23.       return 0;
24. }
```

Rappel

1. La fonction `atoi`

```
#include <stdlib.h>
int *gets(const char *ptr);
```

La fonction **`atoi()`** convertit le début de chaîne de caractères pointée par **`ptr`** en entier de type **`int`**. Elle renvoie le résultat de la conversion.

2. La fonction `memcpy`

```
#include <string.h>
void *memcpy(void *dest, const void *src, size_t n);
```

La fonction **`memcpy()`** copie **`n`** octets depuis la zone mémoire **`src`** vers la zone mémoire **`dest`**. Elle renvoie un pointeur sur **`dest`**.

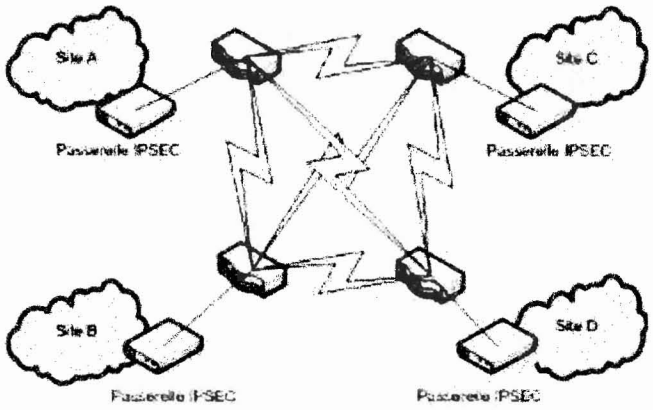
7(14/15)
10

Département informatique
Faculté Electronique et Informatique
USTHB

Le 15/10/2014
concours Doctorat LMD

Partie 4 : Sécurité Réseau

1. Décrire le principe de confidentialité des flux réseau présenté sur le schéma suivant :



2. Proposer une méthodologie générique pour élaborer une stratégie de sécurité réseau