UNIVERSITÉ DE SHERBROOKE

Université de Sherbrooke

Département d'informatique

IFT 606

Sécurité et cryptographie

Chargés de cours :

Marc Verreault

Mohammed Ouenzar

Examen périodique

Mercredi 24 février 2016, 13 h 30 à 15 h 30

Cryptographie et attaque

Consignes :

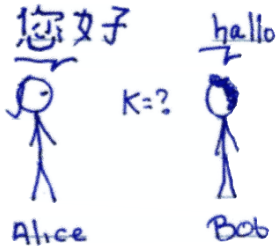
- Répondez directement sur le questionnaire;
- Inscrivez votre nom et matricule sur la première page;
- Cet examen comprend cinq questions réparties sur 17 pages;
- Assurez-vous que toutes les questions s'y retrouvent;
- Seule une page de format 8½ x 14 est permise. Aucune autre documentation n'est permise;
- L'évaluation compte pour 25% de la note finale.

Nom : _____ Prénom : _____

Signature : _____ Matricule : _____

Question 1 : Cryptographie *20 pts*

1. Donnez une définition détaillée de la cryptographie (*2 pts*).



2. Donnez quatre objectifs de la cryptographie (*2 pts*).

3. Expliquez la différence entre le chiffrement par flot et le chiffrement par bloc (4 *pts*).

4. Donnez le nom de la problématique mathématique sur laquelle se base l'algorithme de cryptographie RSA (*1 pt*).

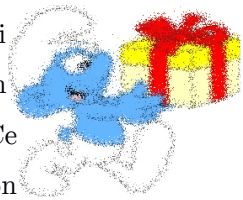
5. Expliquez étape par étape le fonctionnement de l'algorithme RSA en commençant par la génération des clés, jusqu'au déchiffrement du message envoyé (*4 pts*).

6. Comment justifiez-vous la lenteur du chiffrement et du déchiffrement d'un message en utilisant l'algorithme RSA (2 *pts*)?

7. Expliquez la différence entre un certificat électronique et une signature électronique (*2 pts*).

8. Peut-on utiliser la cryptographie symétrique pour assurer le processus d'une signature électronique (*1 pt*)? Expliquez (*2 pts*).

Question 2 : SMURF Attack *20 pts*



L'attaque par rebond ou SMURF attack est une attaque de déni de service avec amplification. L'ordinateur de l'attaquant envoie un paquet spécialement modifié à l'adresse de diffusion d'un routeur. Ce routeur diffuse le paquet à tous les équipements et ordinateurs de son réseau. Ces derniers répondent à l'adresse de la source (celle spécialement modifiée par l'attaquant).

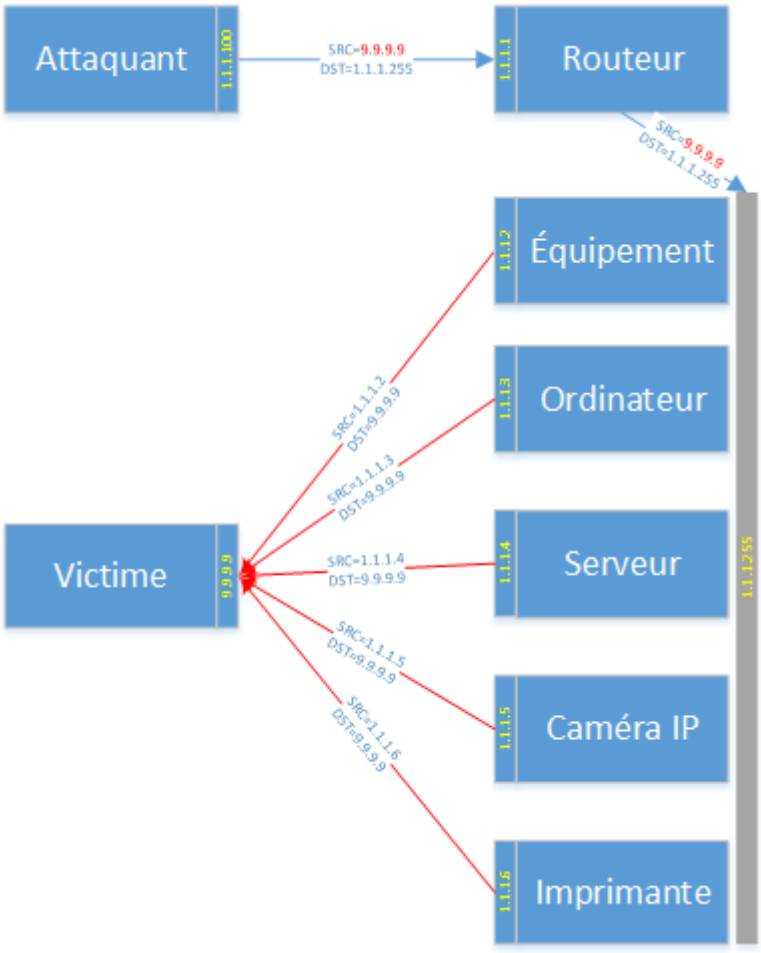


Illustration 1: SMURF attack

1. Expliquez comment l'attaquant peut s'assurer qu'il n'y a pas de défense contre ce type d'attaque avant de la lancer (*10 pts*)?

2. Proposez une variante de l'attaque SMURF pour laquelle la démultiplication s'appliquerait à un serveur DNS (*10 pts*).

Question 3:Buffer overflow *20 pts*

Le programme «loadFontBO.c» suivant illustre une vulnérabilité qui a été publiée (CVE-2015-2426).

```
 1 #include <stdio.h>
 2 #include <string.h>
 3
 4 //
 5 // Illustration du «Underflow vulnerability» de Adobe (CVE-2015-2426).
 6 // Cette vulnérabilité est due à la mauvaise manipulation de
 7 // variables entières short et int. La vulnérabilité est connue sous le nom
 8 // «OpenType Font Manager» et survient dans la librairie dynamique «ATMFD.dll»
 9 // livrée par Adobe.
10 //
11 // Dans cet exemple, on tente de démontrer que lorsqu'on charge le «buffer»
12 // pour la police de caractère dans «fontBuffer», si le fichier chargé
13 // est plus grand que 2^16 octects, on peut écrire/corrompe la mémoire.
14 // Marc Verreault 2016-02-20
15 //
16 static int  fontBufferLength=0x8000; // 32768 (buffer plus grand que 2^16)
17 static char bufferRejet[32768];      // vecteur qui précède en mémoire
18 static char fontBuffer[32768];       // vecteur dans un espace mémoire suivant
19
20 int loadFont(char *myFontBuffer, char *filename) {
21   char localFileName[256];
22   short sindex;
23   int count=0;
24   FILE *F;
25
26   strcpy(localFileName, filename);
27
28   F = fopen(filename,"r");
29
30   for (sindex=0; sindex<=fontBufferLength; sindex++) {
31     // myFontBuffer[sindex]=fgetc(F);
32     myFontBuffer[sindex]='o'; // pour pouvoir identifier qu'on a écrit la mémoire
33     if (sindex == (short)0x3fff | sindex == (short)0x8000 ) {
34       printf("sindex=%d bufferRejet1[0]=%c bufferRejet1[32767]=%c\n",
35              (int)sindex,bufferRejet[0], bufferRejet[32767]);
36       count++;
37     }
38     if (count>=3) break; // pour sortir de la boucle infinie
39   }
40   fclose(F);
41 }
42
43 int main(int argc, char *argv[]) {
44 char fileName[256];
45
46   if (argc!=2) {
47     printf("Usage go2 filename\n");
48     return(1);
49   }
50
51   loadFont(fontBuffer, argv[1]);
52 }
```

1. De manière générale (sans considération pour le code précédent), qu'est-ce qu'une attaque de type Buffer Overflow permet à un attaquant de faire (*5 pts*)?

À l'exécution du code compilé de la page précédente, on obtient :

```
#
maverreault@CH05CHUSW7AUDI2 ~/_
$ ./loadFontBO courierBig.ttf
sindex=16383 bufferRejet1[0]= bufferRejet1[32767]=
sindex=-32768 bufferRejet1[0]=o bufferRejet1[32767]=
sindex=16383 bufferRejet1[0]=o bufferRejet1[32767]=o
```

2) Dans le code source, la fonction main de *loadFontBO.c* fait appel à la fonction loadFont(). Expliquez le résultat obtenu lors de son exécution (*10 pts*).

3) Comment pourrions nous exécuter du code arbitraire sur le système utilisant la fonction vulnérable loadFont (5 *pts*)?

Question 4 : Social engineering *20 pts*

Un attaquant a choisi d'utiliser Kali pour mener une attaque d'ingénierie sociale avec le «Social Engineering Toolkit». Dans le kit, il a utilisé un vecteur d'attaque par site WEB pour lequel il crée un site d'hameçonnage hébergé sur sa machine (Kali).

L'exploit va livrer du code à l'ordinateur victime qui va lui permettre l'exécution de commandes à partir de l'interface «Windows Reverse_TCP Meterpreter» sur le https (TCP/443).

Une fois le site d'hameçonnage préparé, on peut voir, avec un balayage *nmap* qu'un nouveau service est exposé :

```
root@kali:~/ift606-exam2016# nmap -n 10.0.2.13

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-02-20 16:45 EST
Nmap scan report for 10.0.2.13
Host is up (0.000012s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
80/tcp   open  http
1935/tcp open  rtmp
3333/tcp open  dec-notes
6666/tcp open  irc
7777/tcp open  cbt
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 1.70 seconds
```

Ce service est sur le port 8080 et est une version «hackée» du serveur WEB du port 80. Dans cette version «hackée», on retrouve 20 «exploits» différents prêts pour infecter une victime.

L'attaquant doit amener l'utilisateur sur le réseau local à cliquer un lien contenant l'URL : http://10.0.2.13:8080.

La victime utilise une version récente de Windows, à jour, et obtient l'écran suivant lorsqu'il clique le lien depuis son navigateur «Internet explorer».

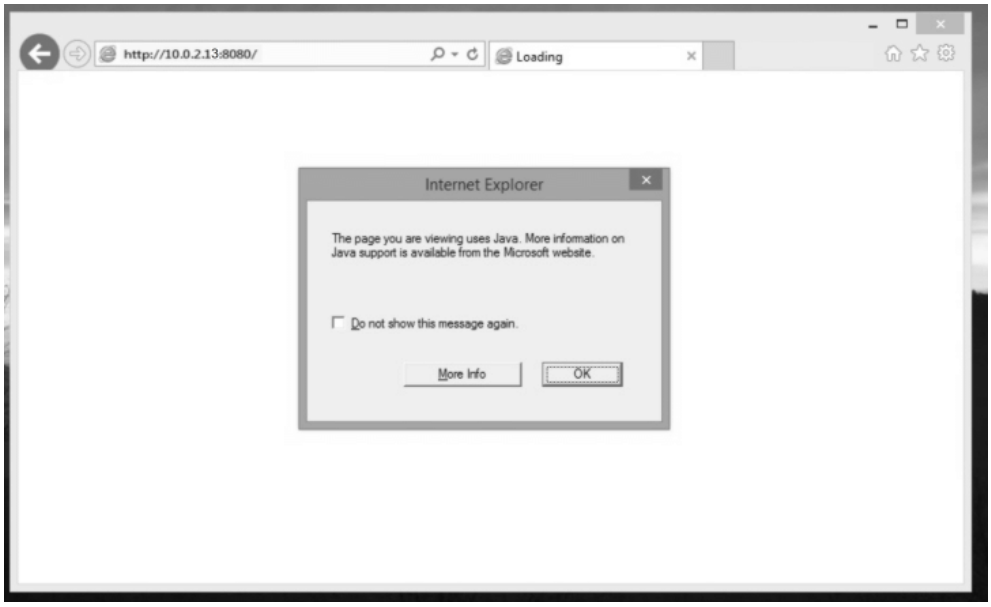


Illustration 2: Écran obtenu sur l'ordinateur de la victime lorsqu'elle clique l'URL http://10.0.2.13:8080

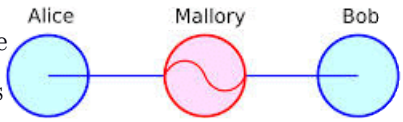
1. Donnez et expliquez des techniques d'ingénierie sociale pouvant servir dans ce contexte (8 *pts*).

2. Expliquez pourquoi l'ingénierie sociale représente un risque important (8 *pts*).

3. Est-ce qu'une attaque «path-traversal» a du être utilisée dans ce processus? Expliquez pourquoi (2 *pts*).

4. Est-ce qu'une attaque «SQL injection» a du être utilisée dans ce processus? Expliquez pourquoi (*2 pts*).

Question 5 : Attaque Man-in-the-middle *20 pts*



L'attaque de l'homme du milieu ou Man-in-the-middle (MITM) est une attaque qui a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis.

1. Décrivez comment une attaque Man-in-the-middle peut être accomplie sur un réseau WIFI et les conséquences d'une telle attaque (*10 pts*).

2. L'attaque MITM est de type composite, c'est à dire qu'elle comprend des attaques de type «Read» et «Spoof». Expliquez. (*10 pts*).