M1 - Outils de l'Internet (Année 2010/2011)

TD n° 7 - Star WEP : La saga complète

Dans ce TD nous étudierons le protocole de cryptage WEP (*Wired Equivalent Privacy*) encore utilisé dans les réseaux Wi-Fi. Nous verrons comment il fonctionne et comment il est facile de le casser (et donc pourquoi il ne faut pas l'utiliser).

Exercice 1. *Épisode IV : Un nouvel espoir*

Le WEP fonctionne à l'aide d'une clé commune secrète de 40 ou 104 bits (notée K). Étant donné un message M à transmettre, la procédure est la suivante :

a. On hache le message M à l'aide de l'algorithme CRC32 et on lui adjoint le résultat. On obtient alors un message $M' = M.\,\mathrm{CRC}(M)$.

b. On génère un *vecteur d'initialisation* (IV) de 24 bits que l'on ajoute à la clé. La clé devient donc $K' = \mathrm{IV}\,.K$ de 64 ou 128 bits.

c. À partir de la clé on génère une suite S d'octets pseudo aléatoire à l'aide de l'algorithme RC4 [1] de même longueur que M'.

d. On effectue un XOR bit à bit de M' et S.

e. On transmet l'IV (en clair) suivi du résultat du XOR.

L'algorithme RC4 « mélange » des octets à l'aide d'une permutation s des entiers de 0 à 255 (stockée sous forme d'un tableau) qui évolue par permutations au cours de l'algorithme. La permutation est initialisée à l'aide de la clé par :

```
s = range(256)
j = 0
for i in range(256)
  j = (j + s[i] + key[i % len(key)]) % 256
  s[i], s[j] = s[j], s[i]
```

Puis, les octets pseudo-aléatoires sont générés par les instructions suivantes :

```
i, j = 0, 0
def octet():
  i = (i + 1) % 256
  j = (j + s[i]) % 256
  s[i], s[j] = s[j], s[i]
  return s[(s[i] + s[j]) % 256]
```

1. Expliquer pourquoi les deux longueurs possibles des clés font que l'algorithme RS4 est rapide à exécuter.

2. En ne travaillant que sur 8 lettres (0-7), cryptez le message « 250577 » à l'aide de la clé « 1337 » et de l'IV « 42 ».

3. Comment peut-on décrypter le message reçu connaissant la clé secrète ?

1. l'algorithme RC4 est la propriété de la RSA, mais il existe des équivalents libres (obtenus par désassemblage ou *reverse-engineering*) tels que ARC4

4. Justifiez en quoi ce protocole assure *a priori* la confidentitalité et l'intégrité du message. Qu'en est-il de l'authentification ?

Exercice 2. *La menace fantôme*

On va maintenant déterminer quelles sont les faiblesses potentielles du cryptage WEP.

1. Que peut-on faire si l'on connaît la clé WEP ?

2. Que peut-on faire si l'on connaît un flux d'octets pseudo-aléatoires correspondant à un IV donné ?

3. Que peut-on faire si l'on sait quel est le message (en clair) qui est envoyé et que l'on intercepte la version cryptée ?

4. Expliquer l'utilité de l'IV.

Exercice 3. *Que la force soit avec toi*

Une première méthode d'attaque est la technique de *brute force*, consistant à essayer toutes les clés possibles jusqu'à trouver la bonne. Avec une clé de 40 bits, une telle recherche durerait environ un mois sur une machine actuelle.

1. Expliquer comment on peut exploiter la psychologie des utilisateurs moyens pour réduire considérablement le temps de recherche dans un grand nombre de cas.

2. Proposer des méthodes pour diminuer le risque d'une attaque par brute force (une méthode a déjà été exposée dans la description du protocole de cryptage).

Exercice 4. *La pire contre-attaque*

1. Rappeler le déroulement d'un *défi* visant à authentifier un utilisateur en vérifiant qu'il connaît la clé secrète.

2. Expliquer comment on peut utiliser cette technique (si le réseau l'utilise) pour obtenir un flux d'octets correspondant à un IV donné.

Ces failles ont été jugées peu dangereuses parce qu'elle ne semblent pas permettre une attaque complète...

Exercice 5. *L'attaque des clones*

On a plus tard découvert que certains IV étaient « faibles » c'est-à-dire qu'ils ne mélangent pas assez la clé et qu'il est possible de retrouver certains octets de la clé avec une bonne probabilité connaissant le flux d'octets.
En écoutant environ un million d'échanges, il était alors possible de retrouver la clé (ou de limiter suffisamment la recherche pour terminer rapidement à l'aide d'une technique brute force).
Ces premiers IV faibles ont été identifiés et ils ne sont plus utilisés.
Puis on a découvert des IV *encore plus faibles* (meilleure probabilité de trouver un octet de la clé) pour lesquels il fallait observer environ 500.000 échanges.
Il peut sembler long d'obtenir un million d'échanges, et pourtant...

1. Expliquer comment en renvoyant des copies de messages volés il est possible de générer un grand traffic sur le réseau.

Le temps moyen pour casser une clé tombe à quelques heures. La solution envisagée est alors de changer de clé à intervalles rapprochés.

Exercice 6. *Fragmentation*

Le coup de grâce est porté par une spécification de la norme 802.11 : la fragmentation des paquets.
Il est prévu que l'on puisse découper des messages en des fragments beaucoup plus petits.

1. Expliquer quel est l'intérêt de permettre la fragmentation dans un réseau sans fil.

2. Les messages échangés par WEP contiennent des en-têtes (comme presque tous les protocoles). Étant donné que ces en-têtes sont prévisibles, expliquer comment on peut obtenir les premiers octets d'un flux correspondant à un IV.

3. Expliquer comment la fragmentation permet d'émettre n'importe quel message sur le réseau.

Il ne reste plus qu'à pouvoir décrypter les messages...

Exercice 7. *La revanche des sites*

Lorsque le réseau Wi-Fi est relié à Internet, la borne Wi-Fi sert de routeur vers l'extérieur. Les communications entre le routeur et Internet se font par câble et ne sont donc pas cryptées.

1. Expliquer comment, en utilisant la fragmentation des paquets on peut décrypter un message en le redirigeant vers un site extérieur (que l'on contrôle).
Indication : Les messages envoyés au routeur commencent par le destinataire...

Exercice 8. *Diffusion*

Si l'on ne peut pas utiliser de redirection vers un site extérieur, on ne peut pas utiliser la méthode précédente. Il est cependant possible d'obtenir des flux d'octets plus longs en utilisant la diffusion (*broadcasting*).
Lorsque le serveur reçoit un message à diffuser, il le décrypte, le ré-encrypte à l'aide d'un nouvel IV et le transmet.

1. Sachant que l'on peut obtenir les premier octets d'un flux (pour un certain IV), expliquer comment on peut obtenir tout un flux (la longueur maximale d'un flux est de 1500 octets) en utilisant la diffusion.

On peut ainsi construire un dictionnaire des flux correspondant à chaque IV...
Cependant, si l'on intercepte un message donné que l'on veut décrypter, on peut également utiliser la diffusion pour obtenir le flux correspondant à ce message.

2. Étant donné un message intercepté (crypté selon un IV fixé) dont on connaît les premiers octets (et donc les premiers octets du flux), expliquer comment on peut utiliser la diffusion pour obtenir un octet supplémentaire du message (et du flux) en envoyant au plus 256 messages sur le réseau.
Indication : Le serveur ne diffuse que les messages correctement encryptés.

Référence : *The Final Nail in WEP's Coffin*, Andrea Bittau, Mark Handley et Joshua Lackey.