Solutions, exercises, set 4

Computer Security

Exercise 4.1

We are using Transport Layer Security where the key exchange is done using RSA. Encryption is done using the 128-bit RC4 cipher and HMAC is done using MD5.

Exercise 4.2

In GSM, the Authentication Center (AuC) generates a challenge $RAND$, the appropriate response RES and a session key K_C and sends all of this to the Visitor Location Register (VLR). The VLR sends $RAND$ to the cell phone which calculates the response $SRES$ and sends it back. If $SRES = RES$, authentication has succeeded.

But how can the GSM phone know that it's talking to the correct network? Well, it can't. Attempting to solve this problem, several subsequent problems arise as described in the course literature. The UMTS authentication is outlined below:

The random challenge $RAND$ and the authentication vector $AUTN$ are authenticated by the use of a Message Authentication Code (MAC). This convinces the cell phone that it received the challenge from someone who shares its secret key K. There is also a sequence number SQN to counter replay attacks. Note that there is no authentication of the network per se, but the keys have to be fresh and the data authenticated. This can (with high probability) only be done if the network knows the secret key, so it seems reasonable to assume that it is indeed the expected network.

UMTS continues its authentication by using a scheme similar to that of GSM.

Exercise 4.3

An attacker might (persuade someone to) send some specially crafted content to STDIN of the program. If the program puts everything into a buffer with a too small size, the attacker might be able to run arbitrary code. See instructions for laboratory 3 for a more detailed description that won't be repeated here.

Exercise 4.4

We let R be something that uniquely determines the row we want to probe. In our case R is the predicate `name='Erika'`. We let our general tracker T be given by `Nationality = 'Sweden'`. Using a tracker attack we make the queries

```
Q1=SELECT SUM(Loan)
FROM Students
WHERE Nationality = 'Sweden'
```

```
Q2=SELECT SUM(Loan)
FROM Students
WHERE NOT (Nationality = ’Sweden’)

Q3=SELECT SUM(Loan)
FROM Students
WHERE name = ’Erika’ or Nationality = ’Sweden’

Q4=SELECT SUM(Loan)
FROM Students
WHERE name = ’Erika’ or NOT (Nationality = ’Sweden’)
```

Erika’s loan is given by $Q3 + Q4 - Q1 - Q2$.

Exercise 4.5

RC4 does not describe how an IV is to be used with the cipher. In the WEP implementation it was decided that the IV was to be concatenated with the key and that the result would be the new key. This can be seen as that a part of the key is known to the attacker. It turned out to be quite easy to recover the key in this situation. In SSL, there is no IV the IV and the key are hashed in order to produce the key used in RC4. Knowing the IV would not give any information about the hash value, provided that the hash function is secure.

Exercise 4.6

The SPI in the header is needed to determine which algorithms are used.

Exercise 4.7

The payload length is based on the number of 32-bit words in the header.

Exercise 4.8

a) By placing a known value before the return address, we can check that this value has not changed before using the return address to continue execution. This value can be a random value that is difficult for an attacker to guess. This of course assumes that the random value is saved somewhere where it cannot be read by an attacker. It can also be a fixed value, see part b.

b) If the canary is NULL, then the attacker, when overwriting the return address, must make sure that the canary is overwritten with a NULL byte. Otherwise, when checking the canary the buffer overflow will be detected. However, overwriting the canary with a NULL means that we must have a NULL in the source string sent to the strcpy() function. This will be treated as an end of string by strcpy() and the overflow will not be allowed to continue. This will protect against buffer overflow attacks, at least when strings operations are used.