FAQ – Lab3 – Version 0.1

Q: I got the client to receive the valid key from the authentication server. The client decrypts it and sends the data to the server.
But the server exits with the following message:

AP_REQ received
key error -1

What does this message means?

A: It means that the server is receiving an invalid session key. This is the session key that comes in the ticket sent by the AS to the client (ticket.kcs).

Q: I have a problem sending the ap_rep.nonce. I have checked the values after encrypting and decrypting it back and it is the right value but when I send it the client receives a 0 for some reason.

A: Here is a suggestion on how to encrypt the time_t variable

The way I did it was passing the (time + 1) to a string with "sprintf" and then encrypting it. On the client side, I used "atoi" to pass the string back to an integer. That is the way your client/server will work with my client/server. You don't need to htonl or ntohl the time_t variables.

Q: a. One issue I am trying to resolve is when the Server receives an AP_ERR packet, Is it just supposed to print an error and keep running, print an error and exit or some other option.

b. Also will the AP server care at all what is in the Client_ID field in the ap_err packet?

c. and when sending an ap_err packet, which ID do I put in the client_id field? From the ticket or the auth?

A: a. When the server receives an AP_ERR, it will exit gracefully. You could print and ABORT and a quick message to explain what happened.

b. The server does not care what is in the client ID field.

c. Put the ID that is coming from the AUTH.

Q: what do I have to do with the first timestamp the client sends to the AS. Should I do any comparison with timestamp2?

A: Timestamp1 is only there for the sake of completeness. Just pass it as part of the AS_REQ packet but you don't need to use it again. In a real world application it is used.

You could use it to limit the validity of the AS_REQ packet to certain period of time. For example if current_time > timestamp1 + 5 minutes, the message is not valid. But in this lab, for simplicity, we are not using it.

Q: The session key is generated by the DES_random_key function right ?
I am doing:
RAND_seed("dlonjpokdfiu3uhekljdno3", 23) ;
DES_random_key(&(cred->kcs)) ;
Also, do I have to convert this generated key into network byte order ? Let me know.

A: DES_random_key generates the session key.
You only need to convert to network byte order those variables that are integers (int, long, short or similars).

Q: How are we supposed to find the client's IP address for the client code?

A: Here is a way of doing it:
1. Get local pc hostname; you could use the function "uname"
2. Use gethostbyname. It will return a "hostent struct"
This is the hostent struct:

```
struct hostent {
        char    *h_name;        /* official name of host */
        char    **h_aliases;    /* alias list */
        int     h_addrtype;     /* host address type */
        int     h_length;       /* length of address */
        char    **h_addr_list;  /* list of addresses as long int*/
}
```

You can get the local ip address from the h_addr_list. Now, notice that h_addr_list is a pointer to a pointers array. So what you really have is something like:

h_addr_list[0] ---> ipaddress[0]
h_addr_list[1] ---> ipaddress[1]

and so on.
Also, you don't need to htonl the ip address since it is already in network byte order.

Here are some sites that might help you
http://rabbit.eng.miami.edu/info/functions/internet.html
http://www.codecomments.com/message380247.html

Q: a. Is this the correct way to generate the key :
DES_random_key(&(cred->kcs)) ;
where: cred is the pointer of struct credential

b. Also, I was testing my authentication server with the client and server provided by you and your client aborts with a message saying:
AS_REP received
Wrong session key received
Session key received: 18a2d48c0a2d48
ABORT

A: a. It is correct. But, don't forget to initialize the RNG necessary for the function DES_random_key to work properly. Take a look at
http://www.openssl.org/docs/crypto/des.html#

b. This error comes from the fact that the session key generated by the AS does not satisfy the DES requirements. You should review your code to generate the session key.

Q: a. what machine do we need to run this on. both sun and msee linux

b. I am not sure about reading in the key off the command line? I got it working for lab1 on the linux machines as a long long integer. But I had to do a conversion because of the endianess. Can you provide some help as to reading in the key as a string and putting it into a usable number?

A: a. Your code only needs to work on msee190 Linux PCs.

b. One way for reading the key from the command line is as follows:
1. The key you will get from the command line is a 16 characters string.
2. Use a loop to read two characters every cycle of the loop.
3. Once you read two characters, use "strtol" to convert these two characters to a hexadecimal value and store the resultant value in another array
4. At the end of the loop you should have converted the key you get from the command line from a 16 chars string to an 8 unsigned char array.
5. After you get step 4 correctly, use the function DES_set_key_check to convert the key from unsigned char to DES_key_schedule and at the same time, to verify that the key satisfy DES requirements.

Q. I'm confused about the behavior of the client after getting an AP_ERR packet.
a. Does it exit, or does it try to go through the authentication again?

b. Does the server exit when it gets an AP_ERR, or does it just start over again listening for AP_REQ packets?

A: a. If the client gets an AP_ERR, it exits gracefully.

b. If the server gets an AP_ERR, it exits gracefully The sender of the AP_ERR also exits gracefully after it sends this package.

Q: what print statement do you use to print the session key ?

A: It is printf("%x",key[k]),
Where "k" is the position of the array (DES_Cblock) element you want to print.