Lecture Network Security
Assignment Sheet 5

Jens Tölle, Wolfgang Moll
Jonathan Chapman, Martin Clauß, Martin Lambertz, Christian Meier

Summer Term 2016

Publication date of this sheet:	09.06.2016
Submission deadline (via email):	21.06.2016, 23:59:59
Discussion of results:	23.06.2016

Results must be submitted via email to
network-security-worksheet@lists.iai.uni-bonn.de
in one archive file named after the scheme
`sheet5_lastname1_lastname2[_lastname3].{tar|tar.gz|tgz|zip}`

Task 5.1 (theoretical): Block Cipher Based MACs

In the lecture you learned about different cipher block modes of operation, namely ECB, CBC, and CTR. Think about which of these modes could be used to create a MAC of a message.

Part (a)

For each of ECB, CBC, and CTR answer the following questions:

- Is it suiteable as a MAC?
- Why or why not?

Part (b)

We will have a closer look at the message length and its influence on the security of the MAC now.

For each of the modes of operation you chose in Part (a) answer the following questions:

- Is it still suitable when dealing with messages of variable length?
- Why or why not?

Task 5.2 (theoretical): RADIUS

There is a service running in the SecLab which authenticates its users using RADIUS. You can access the service by executing the command `authme` on hellgate. The command

will ask for your SecLab credentials and you will receive an "Access granted" or "Access denied" response.

Your task is to provide the following information:

- Which hosts of the SecLab are involved in the authentication process?
- What is the role of each of these hosts in the course of the RADIUS authentication process? Draw a simple sketch of the hosts and the communication between them using the correct RADIUS nomenclature!

Hints:

- You can use `sudo tcpdump -i eth0 -w traffic.pcap` on hellgate to dump all network traffic in the SecLab to the file traffic.pcap.
- Use tools like Wireshark to inspect the captured.

Task 5.3 (theoretical): RADIUS (again)

In the lecture you learned that the RADIUS protocol is not very secure by default. This holds also for the configuration in the SecLab. Try to obtain the RADIUS shared secret used in the SecLab. Write a tool which uses a brute-force attack on the shared secret. Which information is required to launch such an attack?

To solve this task provide:

- The source code of your tool.
- The shared secret.
- A list of the information you needed to perform the brute-force attack.

Hints:

- Have a look at the tips for previous task.
- Use RFC 7511 as a dictionary for your brute-force tool.

Task 5.4 (practical): One-Time Pad

Develop a small client-server application that uses one-time pad encryption. For a one-time pad you should use truly random values. Those values could be obtained under Linux using `/dev/random`. What are the drawbacks of using `/dev/random`? What could you use instead? http://www.onkarjoshi.com/blog/191/device-dev-random-vs-urandom/

The following should happen:

1. create a one-time pad (i.e. the secret key, no key distribution needed)
2. the server listens for connections
3. the client connects to the server
4. after a sucessfull connection, the server sends an encrypted message using the one-time pad

5. the client receives the encrypted message and decrypts it using the one-time pad
6. now the client sents an encrypted message using the one-time pad
7. the server receives the encrypted message and decrypts the message using the one-time pad

Task 5.5 (practical): HMAC

In the lecture you heard about HMAC. Now it is your task to create your own HMAC. Take the hash function of your choice and implement an HMAC using it. Use an appropriate block size for that hash function. To generate the key to be used, execute `name2key` on hellgate. Take the ASCII representation of the output padded to the block size with leading zeros as a key for your HMAC. Use the byte value `0x93` to construct an IPAD, and `0xA5` for the OPAD. Use your HMAC to compute a hash that authenticates this PDF file. Which hash function did you choose? Reason your choice.

Submit

- the source code of your HMAC and
- the HMAC for this PDF document

HMACk(m) = h((k* ⊕ opad) || h((k* ⊕ ipad) || m))

https://en.wikipedia.org/wiki/Secure_Hash_Algorithm