Homework #3
ISA 656 Network Security Spring 2020

This homework is due on 4/22/2020. This is an individual homework assignment to be done by each student individually. The submission of your homework is your acknowledgement of the honor code statement

I have not taken any help on this assignment from anyone and not provided any help to anyone. The solution has been entirely worked out by me and represents my individual effort.

Instructions for homework submission:

- Use a word processor (MS-Word, LaTex, Framemaker, …) for your solutions. No hand writing submission will be accepted.
- Include your name and G-number at the beginning of your homework submission
- Put all your files (e.g., source code, makefile, description, solutions) in a directory named ISA656_HW3_<your last name>_<your G#> and pack all your files in the direcroty with zip or tar and gzip and name your packed file as ISA656_HW3_<your last name>_<your G#>.zip (or tar.gz). Note, your zip or tarball should extract everything into the directory ISA656_HW3_<your last name>_<your G#>. There will be penalty if your zip or tarball does not extract everything into the directory.
- Submit your packed solution to via blackboard.

This homework requires using and programming at the FreeBSD 5.4 VM. Please create a directory named "ISA656_HW3_<your last name>_<your G#> and do everything of this HW there.

1. Implementing transparent IP level encryption & authentication via divert socket (80 points)

   You are asked to write a (could be C or C++) program that can transparently authenticate all IP packets from/to specified IP address with given key by using the divert socket in the FreeBSD 5.4 VM. Specifically, you need to develop a C program (C++, whatever program that runs in the FreeBSD VM): ip_cryptAuthAll.c such that ip_cryptAuthAll <divert port> <remote IP> <key> (e.g., ip_cryptAuthAll 192.168.10.10 secret) will intercept packets diverted to <divert port> and do the following:

   - for any outgoing packet to the specified <remote IP>
     1) encrypt the original payload `X` (the bytes after the IP header) using RC4 with the given <key> to obtain encrypted payload `Y=RC4(X, key)`
     2) Concatenate the encrypted payload `Y` with 16 bytes of `md5(Y|key)`, and adjust the IP header accordingly (e.g., increase the packet total length, re-calculate the checksum) so that the outgoing packet will have payload `Y|md5(Y|key)`to <remote IP>
   - for any incoming packet from specified <remote IP>
     1) check the packet payload for the md5 authentication with the given <key>. Specifically, divide the received packet payload (those bytes after the IP header) as two parts `Y`, `Z` where `Z` is the last 16 bytes of the packet payload and `Y` is the rest.
     2) If (`md5(Y|key) == Z`), decrypt Y using RC4 with the given <key> to obtain `X=RC4(Y, key)`, change the payload of the incoming packet from `Y|Z` to `X`, and adjust the IP header accordingly (e.g., decrease the packet total length, re-calculate

the checksum) so that the incoming packet will be restored to its original form without keyed md5 authentication before sent to the receiving process.

3) Otherwise, print out error message of failed authentication and drop the incoming packet.

You can develop your ip_cryptAuthAll.c by extending divert-loop.c

Experiments:

Once you have developed such program, you need to run two instances of the FreeBSD VM: VM1 & VM2 and to make VM1 & VM2 have two different IP addresses (e.g., ip1, ip2). They should be able to ping to each other. On VM1, set up appropriate ipfw rules to divert incoming/outgoing traffic from/to VM2 to <divert port>. Similarly, on VM2, set up appropriate ipfw rules to divert incoming/outgoing traffic from/to VM1 to <divert port>

Take a screenshot of the following execution result, and name the screenshot file `ISA656-HW3-6a.*` (here * depends on the format of your screenshot, e.g., jpg)

a) At VM1, run ./ip_cryptAuthAll <divert port> <ip2> secretKey
at VM2, run./ip_cryptAuthAll <divert port> <ip1> secretKey
ping from VM1 to VM2
ping from VM2 to VM1

Take a screenshot of the following execution result, and name the screenshot file `ISA656-HW3-6b.*` (here * depends on the format of your screenshot, e.g., jpg)

b) At VM1, run ./ip_cryptAuthAll <divert port> <ip2> secret
at VM2, run./ip_cryptAuthAll <divert port> <ip1> secretKey
ping from VM1 to VM2
ping from VM2 to VM1

Submit file `ISA656-HW3-6a.*` `ISA656-HW3-6b.*` with all your source code and instructions on how to build your executable from your source code.

2. Short Answer Question (10 points)

   If two hosts are using the above transparent IP level encryption & authentication, would the communication between those two hosts subject to the man-in-the-middle attack? Why?

3. Short Answer Question (10 points)

   If two hosts are using the above transparent IP level encryption & authentication, would the communication between those two hosts subject to the replay attack? Why?