COSC 434/534: Network Security

Homework 1 **Due: March 17, 2017**

Ground Rules. You may choose to work with up to two other students if you wish. Only one submission is required per group, please ensure that all group members names are on the submitted copy. Work must be submitted electronically via email. One student in your group MUST email the instructor (mschucha@utk.edu) in order to be given a group ID which will be needed for this assignment. You will not be able to authenticate to servers prior to receiving confirmation that your group is correctly registered.

1. I am who I say I am. [**20 points**] Part 1's client logs into the server to retrieve its secret by submitting a password in the clear. This is obviously defeatable by a man in the middle. You will act as the man in the middle. Download the client from the course website, run it providing your group ID. The client will log in with your group's password, and fetch the secret for your group. You are responsible for learning both what your group's password is and what your secret is. Both of these are going to be random Base64 values. You can trivially achieve this using packet capture methods like tcpdump or wireshark.

```
Usage:  java -jar Part1Client.jar <group name>
```

- Recover your group's password, 10 pts.
- Recover your group's secret information, 10 pts.

2. My password is stonk.[**25 pts**] Client number 2 uses a variant of digest authentication. You can have client 2 log into the server as well. Of course client number 2 is not the best at picking passwords. I want you to execute an offline dictionary attack against user number 2. Client/Server 2, works as follows. The server presents a random challenge string to the client. The client then computes the SHA-256 of the string formed by the following: the user name then a colon character, then the challenge string followed by another colon character then the user's password. Since the SHA-256 hash is cryptographically secure, this would normally prevent us from figuring the user's password, but this user picked a poor password, which opens us up to an offline dictionary attack. As an adversary you only need to see one exchange between client and server, and then without communicating with the server you can extract the password. This can be done via brute force by computing responses for trial passwords to the observed challenge/response, and looking for the password that results in the observed response. You will find the dictionary on the course website exceptionally handy in this regard. Your code should work without needing to contact the server besides one recorded transcript.

```
Usage:  java -jar Part2Client.jar <group name>
```

The code requires the cracklib-small directory to live in the same directory you're executing the jar from.

- Present your group's source code for the offline dictionary attack, 10 pts.
- Recover your group's password, 10 pts.
- Recover your group's secret information, 5 pts.

3. I always wanted to be Alice. [30 points]

Part 3's client/server is using the same authentication scheme as part 2, however a much stronger password is selected, in this case it is a random string. Client 3 is also smarter, and will not log into the server when you're watching. It will happily authenticate to any other server speaking the same protocol. Of course this does not directly yield the password to the attacker, however an adversary can still use this to their advantage. Use the fact that the client will authenticate through you in order to retrieve the secret for your group.

`Usage: java -jar Part3Client.jar <group name> <server ip> <server port>`

The real server, which the Part3Client will not connect to is located at:

`hostname: taranis.eecs.utk.edu, TCP port 15153`.

- Preset your group's source code for conducting your man in the middle attack, 20 pts.
- Recover your group's secret information, 10 pts.

Extra credit portions will be released as soon as they are fully tested.