Exercises Lecture 6: Authentication

October 5, 2020

Online Password Attacks

Solve the Haaukins challenge #25 *"FTP Server Login"* using "Patator". https://ais20au.haaukins.com/

Your job is to access the FTP server using Patator. Patator uses a brute-force / guessing approach to password-cracking.

Hint: You can find information about the FTP server by scanning your own subnet with NMAP.

The machine includes a password list at "/usr/share/john/password.lst". You will have to clean out the comments in the file to use it with patator.

Protocol Design

Consider a Bank, AEDRON, which is designing several authentication methods for their customers to pay using credit cards. Their first idea is to store the customer's PIN in plaintext on the card. In order to obtain the credit card and PIN code, the customer first performs the following enrollment protocol:

1. Customer (C) visits the bank office and provides valid identification, e.g., valid ID or passport, employment agreement, etc.
2. The bank uses the authentication system (S) to generate a credit card and a PIN. The PIN (pin_card) is stored in plain text in the card. The card becomes the authentication token (T).
3. The PIN and card and send by regular mail in two separate letters. Consider a customer (C) who owns a credit card (T), a card reader (R) and the bank's authentication system (S).

Your task is to write the authentication protocol using the protocol design syntax presented in class.

Password Recipes

Password recipes prevent users from creating weak passwords. In this exercise, you will use the library `passay` (http://www.passay.org/) to implement password recipe validators in Java.

Write a Java program using `passay` implementing any combination of the following rules:[1]

1. Passwords must be 15-25 charaters long.
2. Passwords must contain at least one upper-case character.
3. Passwords must contain at least one digit.
4. Passwords must not contain whitespaces.
5. Passwords must not contain any of the characters: <,>,-.
6. Passwords must only contain letters from the latin alphabet (*i.e.*, a-z and A-Z, but not digits or special characters).

The file `PassayExercise.java` provides a template to implement the rules (and implements the first one). First download the library `passay-1.6.0.jar`[2] and that it is placed in the same directory than `PassayExercise.java`. Then to execute the program, simply run:

[1] Note that not all rules can be active at the same time.

[2] You can download the library from here (https://repo1.maven.org/maven2/org/passay/passay/1.6.0/passay-1.6.0.jar)

```
$ javac -cp .:passay-1.6.0.jar PassayExercise.java
$ java -cp .:passay-1.6.0.jar PassayExercise <password_to_check>
```

Hint: The tutorial in https://www.tutorialspoint.com/passay/index.htm helps you getting started and implementing the rules above with `passay`.