Exercise Sheet 4

Introduction to IT-Security

Submit your solutions via Gitlab.
This exercise has no CI tests.
Deadline: *Wednesday, December 17th, 09:00 a.m. CET 2025*

Authentication and Authorization

1. (8 points) The following is a simplified version of a traditional UNIX password file generated with `htpasswd -d`. We also provide it to you in the repository.

   ```
   alice:Frgciyg7sqTr2
   bob:Jh6IDIaYgseW2
   carol:qaQ7syxcwV/XE
   dave:iipXoIpZljw8U
   eve:ms5Sp3iZA2ZJI
   frank:n8ikAG3YQhCDA
   greg:HjwKZxPwokvr6
   john:Tk0g16rm7X3DU
   kelly:1fK.om7dhUjKQ
   norman:5EojqN.nhDxog
   ```

   Make yourself familiar with tools for cracking such passwords. Recover as many passwords as possible and put them into your report. Explain your results.

2. (2 points) To protect from Bob's attacks, Alice invents her own scheme for storing passwords: a password of n bits $p = p_1 p_2 \dots p_n$ is stored as $c = c_1 c_2 \dots c_{n-1}$, where $c_i = p_i \oplus p_{i+1}$. Is this scheme secure? Explain your answer.

3. (3 points) Alice, Bob and Carol each design a challenge-response protocol. Alice uses a random number for the challenge, Bob picks a timestamp and Carol uses a counter. Compare the three approaches. Identify one advantage and one disadvantage for each approach.

Weird Service Reloaded

4. (10 points) **Master:** Re-consider the weird network service of Bob running on TCP port 6666 of the host `weird.exercise.itsec.ias.tu-bs.de`.

   (a) (3 points) Analyze and reverse-engineer the authentication.

   (b) (2 points) Why do you succeed?

   (c) (4 points) Instead of a man-in-the-middle attack implement a client that makes use of this information to communicate with the service and extract the dialog.

   ```
   usage:  weird_client.py [-h] [--out FILE] IP/DOMAIN PORT
   ```

   This tool should report *all* captured (text) messages in and write them to the output file or, if not specified, to standard output (`stdout`).

   (d) (1 point) Provide the retrieved flag in a file called `src/flag.txt`