Final exam

DV026G Information Security

Daniel Bosk

Department of Information and Communication Systems,
Mid Sweden University, SE-851 70 Sundsvall
Email: daniel.bosk@miun.se
Phone: 010-142 8709

2019-08-29

Instructions

Carefully read the questions before you start answering them. Note the time limit of the exam and plan your answers accordingly. Only answer the question, do not write about subjects remotely related to the question.

Write your answers on separate sheets, not on the exam paper. Only write on one side of the sheets. Start each question on a new sheet. Do not forget to *motivate your answers.*

Make sure you write your answers clearly, if I cannot read an answer the answer will be awarded no points—even if the answer is correct. The questions are *not* sorted by difficulty.

Time 5 hours.

Aids Dictionary.

Questions 5

Preliminary grades

Each question can be awarded up to three points: one point for E, two points for C and three points for A.

To get an E, you must get at least one point on each question — *i.e.*, no question must be awarded zero points. For a C, you must get two points on at least half of the questions. For an A, you must get three points on at least half of the questions.

If you get zero on one question, you will get Fx and the possibility for oral complementary assessment. If you get two or more zeroes, you must retake the exam.

Questions

The questions are given below. They are not given in any particular order.

(3p) 1. The security of common off-the-shelf devices, such as smartphones and home routers, can have severe implications on entire societies and even national security.

Discuss why this is the case and the responsibility of, *e.g.*, a smartphone-app developer.

Suggested solution An example from the real world, quoting Biller (Bloomberg, 2019):

> Brazil's federal police arrested four people for allegedly hacking 1,000 cellphones belonging to various government officials, including that of President Jair Bolsonaro.
>
> Police detective João Vianey Xavier Filho said the group hacked into the messaging apps of around 1,000 different cellphone numbers, but provided little additional information at a news conference in Brasilia on Wednesday. Cellphones used by Bolsonaro were among those attacked by the group, the justice ministry said in a statement on Thursday, adding that the president was informed of the security breach.
>
> [. . .]
>
> In the court order determining the arrest of the four suspects, Judge Vallisney de Souza Oliveira wrote that the hackers had accessed Moro's Telegram messaging app, along with those of two judges and two federal police officers.

Another example is the Russian meddling in the US election, which relied on exploiting common off-the-shelf technologies used by ordinary people. (We discussed this during the first seminar.)

(3p) 2. Review the following piece of code. Identify potential problems and explain why they are problems.

This is a shell script by the name of pwdauth, it runs with root priviledges.

```
#!/bin/bash
# pwdauth <user>

# get the username from command line
username = $1

echo "Please enter the password for $username:"
read passwd

real_passwd = $(cat /secure/passwds | grep $username)

if [ $passwd = $real_passwd ]; then
  sudo -u $username /bin/bash
  exit 0
else
  exit 1
fi
```

Suggested solution There are several problems. For example the username variable is not protected with quotation marks, meaning it will be parsed. This means that commends and arguments can be injected as part of the username.

When getting the real password, one can pass a new file to search instead of /secure/passwds. Also, one can set passwd to `1 = 1 -o 1` to make the password check always succeed.

Since the program is running as root, one can get root priviledges by exploiting this program.

(3p) 3. You're having dinner with a few enthusiastic entrepreneurs, start-up starters and tech trendsters. They want to create a revolutionizing food ordering app: a user should be able to keep favourite orders (*e.g.*, custom pizzas) for easy reordering, easy but secure payments and delivery.

If the app doesn't have any security, it won't survive at all. (If someone can order pizza at the expense of someone else, this is not going to work.) You need great usability (to compete with the gazillion other food-ordering apps) and great privacy (also to compete with the gazillion other food-ordering apps). (Well, nowadays, maybe the app won't survice the GDPR fines if it doesn't have privacy.)

What properties do you need for the different functions, what user data do they require? (Remember: the principle of data minimization!)

Note: You don't have to rely on existing technologies, such as credit cards or Swish, you must specify the security and privacy properties you need for any function, *e.g.*, the payment system (that includes if you want to use Swish too). There can be separate food and delivery services, they don't have to be the same, whatever you see fit to maximize security, privacy and usability. Remember, you're working with some enthusiastic entrepreneurs who will not hesitate to create another trendy-tech start-up.

You're expected to use (*i.e.*, show) your knowledge and skills from the topics access control, authentication, accountability and usability. You will also use (*i.e.*, show) your knowledge of some high-level properties from cryptography. (Don't leave the exam room early, spend more time on this question instead, remember it's the size of 4–5 questions.)

Suggested solution Keeping favourite orders *etc.* can be done by storing the data in the smartphone app. That way it never leaves the user, hence it will not cause any problems. Then the user can back it up using the same means as for his or her other data.

For delivery, the address must be given to the delivery service, otherwise they don't know where to deliver.

The actual order, *e.g.*, pizzas and custom pizzas, must be given to the food service, otherwise they don't know what to prepare. The food service can be different from the delivery service: the delivery service only needs to know they deliver food (so they can keep it warm or cold), the food service doesn't need to know where their products are delivered.

Both services must be compensated, the food service wants money for their products, the delivery service wants money for the delivery. The food service shouldn't learn the cost of the delivery, that leaks some information about the destination (*e.g.*, longer distance costs more). The delivery service shouldn't learn the price of the food, that leaks information about the client (pricy food, wealthy client, good for robbery?).

The payment system must support anonymous payments, *i.e.*, if Alice pays Bob twice, Bob cannot link those payments together (and thus not link them to Alice). It must also be secure, so that Alice cannot pay from Bob's account.

The delivery service must deliver the right food to the right address. The food service must give the right food to the right delivery person. There must be an identifier, such as an order number, that is the same for both. The delivery person must prove that he or she is authorized to pick up the food (otherwise someone can say they're the delivery person although they're not and simply steal the food). We can solve this as follows: When a user orders food, he or she will first book and pay the delivery service. While doing this, he or she can give the delivery service a token. Then the user proceeds to pay the food at the food service, saying that the person picking the food will present that token. This token will be hard to guess and will only be used once.

(3p) 4. Browser fingerprinting attemtps to build a unique fingerprint for a web browser to track it across the web without making it store cookies. Whenever a browser connects to a web server it gives some information to it, *e.g.*, browser make and model (*e.g.*, Firefox v59.0) which fonts are available *etc.* Explain how this can be used to create a unique fingerprint to track browsers (*i.e.*, users) across the web. How much data is required? (You don't have to give any numbers, just how to calculate them.)

Suggested solution Browser versions vary, how often and when people update varies too. So there will be several browser versions used at the same time. On top of that, several browsers as well. Add a variety of fonts, plugins *etc.* These can be combined in many ways.

If there are many possible combinations, then the probability of two users sharing the same setup is small. Thus these data yield a unique fingerprint to identity a user (*i.e.*, user's web browser).

We can see each property as a random variable and use entropy as a measure of how much information they provide. We would then need the (base 2) logarithm of the number of people in the world to uniquely track everyone.

(3p) 5. Smartphones is a good example of where users a intentionally locked out. For example, a user is allowed to install a pay-for app to try and later change their mind to get their money back. This requires that the user can install the app on their phone, but that they cannot access the binary to make copies. This way, when they change their mind, then the phone can remove the binary and ensure the user has no copy remaining.

Explain what is technically needed to achieve this and discuss the limitations.

Suggested solution The phone must prevent physical access to its components. Otherwise one can access the storage and copy the binary. The phone must prevent the user from running software on the phone that can read the storage (to copy the binary).

Assume that the phone comes with a clean operating system. Then the operating system can ensure that the user cannot read any part of the storage. Any upgrades must be authenticated, to not break the chain.

However, there are bugs, which sometimes can be exploited to make the operating system execute unapproved code. This code can be used to extract the binary once, or used to modify the operating system.

To prevent this, the phone's hardware must boot only approved software (operating systems). *I.e.*, software which is digitally signed, then it cannot be modified without detection.

For the remote app store to verify all this, we need remote attestation. *I.e.*, a small hardware component that can verify the integrity of the running software. It will then issue a digital signature of the system state, which the app store can read and verify that the phone is not "rooted" or "jail-broken".