Academic Year: Winter Semester 2021/22

6090: Security of Computer and Embedded Systems

Problem Sheet 2

Security Fundamentals & Access Control

In this problem sheet, we will

- Deepen our knowledge of the core principle of information security: Confidentiality, integrity, and availability
- Deepen our knowledge of access control models and related concepts such as identification, authentication, and authorization

1. Information Security Fundamentals

In the lecture, we discussed the fundamental principles of information security. The following exercises will deepen your understanding of these basic principles.

There are three principles that, together, are called the *CIA triad*. As motivated in the lecture, it is often hard to achieve all three at the same time. Do you recall them? To freshen up your knowledge, you might want to read the first chapter of [1].

Exercise 1: *CIA Triad*

The fundamental information security principles are often called the CIA triad. Which three principles form this triad?

- ☐ Authentication
- ☐ Confidentiality
- ☐ Accessibility
- ☐ Identity
- ☐ Availability
- ☐ Classification
- ☐ Intimate
- ☐ Integrity
- ☐ Control

Exercise 2: *Threat Analysis*

In this exercise, we consider three variants of a significantly simplified payment process for debit cards (or credit cards). We assume a debit card that is equipped with both a *magnetic strip* and a *chip*. Now let us consider three different methods for verifying the identity of the card holder and authorizing a transaction[1].

- **Signature (and magnetic strip):** The card terminal reads the card data (card number, expiry date, and name of the card holder) from the magnetic strip. To authorize a transaction, the card holder needs to sign a paper slip.
- **Offline PIN verification (and secure chip):** The card terminal reads the card data from the chip. To authorize a transaction, the card holder needs to enter a PIN (personal identification number, usually a 4-5 digits). The PIN is checked by the secure chip.
- **Online PIN verification:** The card terminal reads the card data from the chip. To authorize a transaction, the card holder needs to enter a PIN which is checked by the issuing bank using a secure online connection.

Compare the three methods with respect to

1. Two different security mechanism an attacker would need to circumvent to misuse a card (e.g., use a stolen card)
2. The different approaches with respect to integrity of the data used for verifying the identity of the card holder
3. The confidentiality of the data used for verifying the identity of the card holder
4. The availability of the process for verifying the identity of the card holder

[1] Note that, in reality, all three variants are used. While most point-of-sales (POS) terminals are using the "offline PIN verification", they usually allow a fall-back to the "signature" approach in case the chip is unreadable. The "online PIN verification" is widely used in ATMs (that, depending on your bank and card type, also allow you to change your PIN).

2. Access Control Models

In this exercise, we deepen our knowledge of role-based and discretionary access control. You might want to read Chapter 4 (Access Control) of [2] to extend your knowledge in these areas.

Exercise 3: *RBAC*

Consider a simple university that has students, demonstrators, and lecturers that need to work with lecture material. The lecture material contains slides, exam papers, and solutions for exam papers.

1. Model the following security policy using role-based access control (*RBAC*).
    - Lecturers can read and write all types of lecture material
    - Demonstrators can read and write slides
    - Demonstrators can read exam papers and solutions
    - Students can read slides and exam papers

    Let's assume we have the following subjects and objects:

    - Subjects: **`elif`** (lecturer), **`bilge`** (lecturer), **`kavun`** (demonstrator), **`alice`** (student), **`bob`** (student)
    - Objects: **`6090_slides`** (slides), **`6090_exam`** (exam paper), **`6090_solutions`** (solutions)
2. Try to simplify our RBAC model by introducing role-hierarchies (hint: the set of permissions of demonstrators is a super set of the rights of students).
3. Extend the RBAC permissions with constraints that allow for specifying that lecturers should only be able to write exam papers where they are the owner.

Exercise 4: ***Multi-level Access Control***

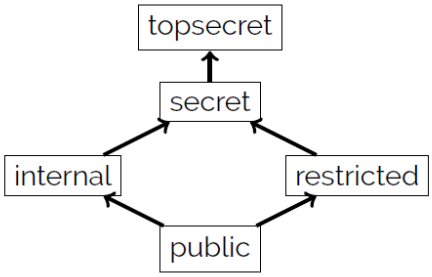
From spy movies we know the idea of assigning labels to documents that describe the confidentiality, e.g., top secret, secret, public. Such access control models are widely used in government or military applications (a famous example is the Bell-LaPadula model). The core idea of such access control models is to assign labels to data (documents) and to order the labels in a hierarchy.

For example, consider the following multi-level security setting in which all data is labelled with one label out of the following set of labels:

$$\{\text{topsecret}, \text{secret}, \text{internal}, \text{restricted}, \text{public}\}$$

Access by a user to a (labelled) file is granted if the user has a sufficiently high clearance level.

The labels are ordered in the following hierarchy:



Assume that data labels are applied at file granularity, i.e., a label is always assigned to the whole content of a file.

Answer the following questions. For each of the questions, your answer should be a subset of $\{\text{topsecret}, \text{secret}, \text{internal}, \text{restricted}, \text{public}\}$.

1. Write down the user clearance level(s) required to read files that are labelled “restricted”.
2. Write down the user clearance level(s) required to read files that are created from the content of two files, one with the label “restricted” and one with the label “internal”.

Exercise 5: *DAC*

Recall the scenario from Exercise 3. Again, we want to implement the informally given security policy. This time, we assume that our system only supports a POSIX-compliant discretionary access control (*DAC*) model.

Users (subjects, usually specified in **`/etc/passwd`**) belong to groups (usually specified in **`/etc/group`**):

- A user is member of one or more groups
  - Groups can have zero or more members
  - Files are owned by one user and one group
- Access rights on files include separate read and write permissions for
  - The user owning the file
  - The group owning the file
  - All other users (excluding the file owner)

For example, a file (device)

```
crw-rw----+ 1 root    video      81,   0 Nov 5 12:02 video0
```

can be read (**`r`**) and written (**`w`**) by the user **`root`** (the first **`rw`**) and all members of the group **`video`** (the second occurrence of **`rw`**). Other users can neither access the file in read, nor in write mode.

Answer the following two questions:

1. Implement the security policy for our university in this DAC model by encoding roles into groups and cleverly setting access rights on the files representing the objects.
2. Can any non-hierarchical RBAC model be translated into a POSIX-style DAC model? Give an informal argument/justification for your answer.

Exercise 6: *POSIX File Systems*

As we have already seen in Exercise 6 (you might want to re-read the brief introduction to DAC in the description of Exercise 6), most Linux (i.e., POSIX-compliant) file systems support, as default, a variant of DAC. In this exercise, we will discuss two not so well-known features of the POSIX file permission system: the *set-gid* and *sticky-bit* on directories.

For this exercise, it is recommended to team up with a colleague or to work on a Linux machine on which you have multiple *user accounts* (they should be regular users, as the system administration user **`root`** is handled differently in certain cases). Furthermore, we assume that you have a group (called **`mygroup`** in our example) on your system, which

- is not your default group and
- contains at least two different users as members.

1. Create four directories with the following access rights:
   ```
   /tmp/problem01> ls -l
   total 12
   drwxrwxrwx 2 kavun mygroup 4096 Nov 5 12:11 test01
   drwxrws--- 2 kavun mygroup 4096 Nov 5 12:12 test02
   drwxrwxrwt 2 kavun mygroup 4096 Nov 5 12:12 test03
   d------rwt 2 kavun mygroup 4096 Nov 5 12:11 test04
   ```
   Note the set-gid bit (**`chown g+s`**) on **`test02`** and the sticky-bit (**`chmod 1777`**) on **`test03`**.
2. How did you create the directories? Explain the numerical encoding of access rights used by the program **`chmod`**.
3. What owner and access right information do freshly created files in **`test02`** have? Compare this to freshly created files in **`test01`**. What behaviour is enforced by the set-gid on directories?
4. Create different user files in the (world-writable) directories **`test01`** and **`test02`**. Try to delete those files (also include a test where you try to delete files owned by user *A* while you are logged in as user *B*). How does the behaviour in **`test01`** and **`test02`** differ? For which system directories is the behaviour enforced by the sticky-bit important?
5. Try to create files in the directory **`test04`** as owner of the directory. Who can access the directory **`test04`**?

For this exercise, you need to open a shell (e.g., a terminal in Linux or OS X).

References

1. J. Chirillo and E. Danielyan. Sun Certified Security Administrator for Solaris 9 and 10 Study Guide, Chapter 1: Fundamental Security Concepts. McGraw-Hill, 2005. URL https://www.mhprofessionalresources.com/downloads/products/0072254238/0072254238_ch01.pdf

2. Ross J. Anderson. Security Engineering: A Guide to Building Dependable Distributed Systems. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2001. The complete book is available at: http://www.cl.cam.ac.uk/~rja14/book.html

3. Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. Handbook of Applied Cryptography. CRC Press, Inc., Boca Raton, FL, USA, 5th edition, 2001. The complete book is available at: http://cacr.uwaterloo.ca/hac/

4. D. Elliott Bell and Leonard J. LaPadula. Secure Computer Systems: A Mathematical Model, volume II. In Journal of Computer Security 4, pages 229–263, 1996. An Electronic Reconstruction of Secure Computer Systems: Mathematical Foundations, 1973.

5. M Golla, M Wei, J Hainline, L Filipe, M Dürmuth. What was that site doing with my Facebook password? Designing Password-Reuse Notifications. Proceedings of the 2018 ACM SIGSAC Conference, 2018.

6. Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D Ullman. Protection in Operating Systems. Communications of the ACM, 19(8):461–471, 1976.

7. Konstantin Beznosov. Requirements for Access Control: US Healthcare Domain. In Proceedings of the 3rd ACM workshop on Role-based Access Control (RBAC), page 43, New York, NY USA, 1998. ACM Press.

8. Achim D. Brucker and Helmut Petritsch. Extending Access Control Models with Break-glass. In Barbara Carminati and James Joshi, editors, ACM Symposium on Access Control Models and Technologies (SACMAT), pages 197–206. ACM Press, New York, NY USA, 2009.

9. eXtensible Access Control Markup Language (XACML), version 2.0, 2005.

10. Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based Access Control Models. Computer, 29(2):38–47, 1996.

11. Ravi S. Sandhu, David F. Ferraiolo, and D. Richard Kuhn. The NIST Model for Role-based Access Control: Towards a Unified Standard. In ACM Workshop on Role-Based Access Control, pages 47–63, 2000.