Security Lab – Security Requirements Engineering and Threat Modeling

1 Introduction

In this lab, you are given the description of a web application, based on which you will perform a threat modeling process to define additional security requirements.

The goal of this lab is to collect experience and get familiar with the security requirements and threat modeling process you learned in the lecture.

2 Description

The company ACME wants to provide a market platform on which any registered user can offer and buy products at a fixed price. Since there's no suitable software product available that the company could use as a basis, an own platform is developed. In the following, you find information about various details about the platform, including some security requirements that have already been defined. Read the entire description without losing too much time with the details to get a first overview; when solving the actual task you'll likely come back to this description to analyze the details.

In reality, you would acquire this information by interviewing stakeholders and engineers and extracting information from already available artifacts.

2.1 Business idea and functionality

In the following, the business idea and basic functionality of the market platform are described.

- The market platform is a web application and can be used with the web browser (without additional client-side software).
- Only registered users can offer or buy products. During registration, a user must select user name and password and enter a valid e-mail address.
- Products are described in a similar way as on auction platforms such as ebay: The user enters all details about a product (product category, title, description, images, price, delivery options, expiry date of the offer). The product is then offered on the platform until it is bought or until the expiry date is reached.
- Users can search for products and registered users can buy products at any time.
- Every user has a cash account on the platform. A user can deposit money in his account by making a payment with his credit card. In addition, a user can withdraw money by transferring it to a bank account
- Buying a product always happens via the platform-internal accounts of buyer and seller. The amount that was defined for the product is transferred from the account of the buyer to the account of the seller. Product can only be bought if there's enough money in the buyer's account.
- The maximum price for a product is CHF 2'000. In addition, the balance of an account must never by higher than CHF 10'000. If the balance is higher, the user must reduce it below CHF 10'000 within 7 days (e.g. by buying products or transferring money to an own bank account).
- Offering products is free and unrestricted for registered users. The market platform earns money by getting 2% of the price for every product sold; this amount is collected during the money transfer between the accounts of buyer and seller when a product is bought.
- Delivery of products happens directly between seller and buyer and is outside the control of the market platform.

- Usage of the application is optimized for simplicity. This should help reaching also users that so far avoided online market platforms.
- Protecting the users' privacy is paramount. ACME would consider a breach where critical user data (passwords, credit card information, transaction data...) were stolen to be highly critical that would threaten the existence of the platform. Likewise, illegitimate access to a user account is also considered highly critical.
- High availability is important because any non-availability of the platform results in reputation damage and users may switch to other, similar platforms.

2.2 System components

In following, some information about the system components is provided.

- In a first phase, the system is kept as simple as possible because in the beginning, only a small number of users will be expected. Later, replicating system components will help to deal with a large number of users.
- In the first phase, one web application server and one database server are used (two separate hardware components).
- A hardened Linux system is used as operation system, on which only the necessary software components are installed. The systems are updated (patched) regularly.
- Tomcat is used as web application server and MySQL as database server (accessed via port 3306). These components are regularly updated as well.

2.3 Operation and user groups

In the following, you find information about the planned operation and the different user groups.

- Both server components will be operated in the computing center of a well-established provider on dedicated hardware.
- The servers will be located in the network 188.27.20.0/29 and can be reached at 188.27.20.2 (web application server) and 188.27.20.3 (database server). There are no further systems in this network and a packet filtering firewall makes sure that only the following traffic can get into the network: FTP, SSH, HTTP, HTTPS, MySQL.
- The provider offers some defensive measures (filtering, redundant network access) against DoS and DDoS attacks (in the sense of flooding). ACME plans to use these measures during operation and currently (and therefore also for this lab) they are considered adequate.
- On the web application server, extensive logging (by the operation system and the web application) is done on the local file system. No logging is done on the database server because any database access of users is always done via the web application server and therefore logged already there. Logging should help to determine the cause of critical events (e.g. illegitimate access to another user's account) after such an event has occurred.
- With respect to user groups, there are users, marketplace administrators, marketing people and system administrators. The groups are described as follows:
    - Users are the actual customers of the platform that offer and buy products. Any Internet user can be a user of the platform and they access the web application via HTTP (port 80) and HTTPS (port 443).
    - Marketplace administrators are employees of ACME that analyze and – if necessary – remove offers from the platform, e.g. if illegal products are offered. If a user repeatedly breaches the rules of the platform, an administrator can exclude him. The administrators perform their tasks also via the web application, also using HTTP and HTTPS. Technically, marketplace administrators are web application users with additional permissions that grant them access to the administrative areas and functions.

  - Marketing people are also employees of ACME that access the web application via HTTP and HTTPS to perform, e.g., statistical analyses about current products, sold products and so on. Just like the marketplace administrators, marketing people are also users with additional permissions to access specific areas of the web application.
  - System administrators are employees of ACME that perform administrative tasks (e.g. updates) and log file analyses. To do so, they can access the web application server with FTP (ports 20 & 21) and SSH (port 22) to download log files for analysis (FTP) und modify web resources (SSH). In addition, the can access the database server via the MySQL port (3306) to make necessary adaptations.
- Credit card payments (to deposit money on the internal account) are done by the users within the web application, but the actual processing of the payment is done via an external company (a so-called Payment Service Provider (PSP)), which is the standard in today's online business. To make payments for the users as easily usable as possible, the credit card information (name, number, CVS code, expiry date) is stored in the database, which frees the user from entering this again during future payments.

2.4 Security requirements

ACME is aware that this is a security-critical application and as a result, some security requirements have already been defined:

- HTTP can only be used as long as a user is not logged in (e.g. to browse products).
- To perform the login, HTTPS must be used. Once a user (or marketing administrator or marketing people) has logged in, only HTTPS can be used. In addition, user registration must use HTTPS.
- The web application server uses an EV certificate.
- The web application server must be configured in a secure way. Only SSL 3.0 and TLS 1.0 – 1.2 are supported. The server supports only secure cipher suites (AES, 3DES or RC4 with key lengths ≥ 128 Bits, no MD5, no anonymous DH).
- During registration (and password change), selected passwords are checked for password strength (at least 10 characters, at least 1 capital letter, at least 1 digit).
- If a user fails to authenticate correctly five times in a row (correct user name but wrong password), the account is locked. The user must contact ACME support to unlock the account.
- Passwords are stored in the database in plaintext. The reason is that if a user forgets his password, ACME can send him his currently used password and the user does not have to perform a cumbersome password change.
- All actions must be logged and can be assigned to a user (users, marketplace administrators, marketing people and system administrators).

3 Task

You task is to perform a complete threat modeling process based on the scenario described above and to define additional security requirements. Every step below contains information about how the results should be documented. With respect to tools, you can use whatever you like, what's important is that the instructor can understand your solution. Everything from hand-drawn diagrams and manually filled in tables to documents created with a program is fine.

3.1 Step 1: Business goal and security goals

Describe first the business goal of the application. Try to do this with one concise sentence.

Then define at least three and at most five reasonable security goals of the application. They should support the business goal. Don't go into the details too much but try describing high-level security goals.

You get the necessary information for this task from the business idea and the functionality (section 2.1).

3.2 Step 2: Collect information

The next step consists of collecting information to truly understand the system to be developed. For this, study the entire section 2 (again), so you understand the system as well as possible. If you have questions, ask the instructor.

You don't have to hand in anything in this step.

3.3 Step 3: Network und data flow diagrams

Based on the collected information, you should now be able to draw a network and a data flow diagram. You can draw them either manually or by using a tool. With respect to tools, you can use e.g. Visio[1] (Windows), Omnigraffle (Mac) or the freely available OpenOffice Draw or Dia (available for multiple platforms). But don't waste too much time installing or using these tools, because what counts is that you draw usable diagrams and not that they look perfectly.

Start by drawing the network diagram. Focus on the main components (servers, firewalls...), the user groups, and which users can access which components with which protocols.

Then, draw a high-level data flow diagram. Choose a similar level of detail as in the first example in the lecture. Consider again all user groups and also the most important data stores (e.g. log files) and the privilege boundaries. Take care that every data store has at least one reading and one writing process. Also, give every element a reasonable name.

3.4 Step 4: Identify threats

You'll use STRIDE for threat modeling, which already identifies the basic threat categories. Study the relevant slides from the lecture material so you understand STRIDE and can apply it below in step 5.

In addition, think about who may actually attack the application and what the attack goal of these groups may be. This is important because the assumptions about the attacker will influence the threats and the sophistication of the attacks you'll consider during threat modeling, which again influences the resulting security requirements. Write down at least three different attacker groups and specify for each group why you consider it to be realistic and what the corresponding attack goals are.

3.5 Step 5: Identify vulnerabilities

Apply STRIDE by going through the elements in the data flow diagram (from step 3) and identifying corresponding threats and vulnerabilities. Document the vulnerabilities in a table as below (the example is from the lecture):

Nr.	Element	Cat.	Description
V1	Librarians	S	No minimal password strength is required, which will likely result in weak passwords. In addition, librarians use a shared account, which increases the probability that passwords will be written down near the librarians' computers,
V2	...	...	...

„Nr." identifies the vulnerability, „Element" corresponds to the element in the data flow diagram, „Cat." identifies the STRIDE category (S, T, R, I, D or E), and „Description" should contain enough text to comprehend the vulnerability.

[1] For Visio, there exists the Microsoft SDL Threat Modeling Tool, but it's certainly not mandatory for this lab: http://www.microsoft.com/security/sdl/adopt/threatmodeling.aspx

To successfully solve this step, you should describe at least 10 significant vulnerabilities, where each element type (external entity, data flow, data store, (multiple) process) must be used at least twice and each STRIDE category at least once.

You don't have to completely go through each element in your data flow diagram in detail, as this would require much more time than you should invest in this lab. Instead, focus first on the elements that appear to be most interesting as targets of attacks, which should allow you to efficiently find 10 vulnerabilities. Also, consider the security goals from step 1 and the assumptions about realistic attackers from step 4 to identify realistic and relevant vulnerabilities. And don't make security assumptions: If a specific security measurement is not included in section 2, then assume it does not exist.

3.6 Step 6: New security requirements

Based on your list of vulnerabilities from step 5, define security requirements to eliminate these vulnerabilities. Be creative and think about how the vulnerabilities can be eliminated (or at least reduced in criticality) efficiently, with reasonable effort, and without significantly reducing the usability of the application. You should cover all vulnerabilities from step 5.

Document the security requirements in a table as below:

Nr.	Description	Vuln.
R1	Passwords must have at least the following complexity: ≥ 10 characters, at least one digit, and at least one special character	V1
R2	Only personal user accounts are used.	V1
R3	...	...

„Nr." is the number of the security requirement, „Description" should contain enough text to comprehend the requirement, and „Vuln." identifies the vulnerability/-ies in step 5 that is/are addressed with the requirement. Note that multiple requirements may address the same vulnerability (as in the example above), but a requirement may also address multiple vulnerabilities.

Lab Points

For **4 Lab Points** you must hand in your solution to the tasks in section 3 (all steps except step 2) to the instructor. Your solution must follow the requirements given in the steps and be documented as described (diagrams, tables etc.). You get the points as follows:

- 1 point for the business and security goals (step 1) and the network diagram (step 3)
- 1 point for the data flow diagram (step 3) and the realistic attacker groups (step 4).
- 1 point for the list of the vulnerabilities (step 5).
- 1 point for the new security requirements (step 6).

You can hand in pdf and doc(x) formats via e-mail, but you can also hand in a handwritten paper solution (on paper or scanned). Solutions that were obviously copied from others won't give any points.

If you hand in your solution via e-mail, use „SecLab - Security Requirements Engineering - group X - name1 name2" as the subject; corresponding to your group number and the names of the group members.