Contents

1 Introduction 2
2 Security Architecture 2
3 Threat Modelling 3
4 Technical Threat Modelling and STRIDE 5
5 Conclusion 7
1 Introduction

In previous chapters we learned about how softwares break and the role of security in a software project. In chapter one we learned about how a data entry gets into the process and interpreted as code because of the confusion between input and code. In chapter two we saw how a valid input can produce output that can be used in an attack. We also learned about the the way data should be treated in the system by validating it as a first step and then treating it as a suspect, and finally encode the output appropriately. In chapter three we looked into software development steps and associated security role in each of them. In this chapter we go through the architectural risk analysis which will tell us about what can go wrong in our system so that we can act accordingly. First we go through an overview of the required terminology for the chapter. Second we talk about the technical threat modelling and business threat modelling, their differences, and visualizing them. Finally we give an overview of STRIDE, a model developed by Microsoft for data flow analysis.

2 Security Architecture

Security architecture is defined by the procedure of choosing design principles in accordance with the desired level of security for the application. These high-level design principles enable the developer to decide with confidence about the security attributes in the application. Two of the most important design principles are security risk analysis and threat modelling.

The important vocabulary in these concepts are risk and threat. Threat and risk are often mean the same thing. They are often used interchangeably. Therefore, security risk analysis and threat modelling are often the same thing. In some resources threats are defined as actors. Therefore, it is important to know the definition of these keywords before going through a security resource. Another important set of keywords are threat, weakness, and vulnerability that you should make sure to not confuse them together. Vulnerability is defined by Information
Technology Security Evaluation Criteria (ITSEC)\(^1\) as:

"The existence of a weakness, design, or implementation error that can lead to an unexpected, undesirable event compromising the security of the computer system, network, application, or protocol involved."

Threat on the other hand is defined by NIST SP800-30 as:

"The potential for a threat-source to exercise (accidentally trigger or intentionally exploit) a specific vulnerability."

Therefore, your system may have a risk, or threat of something but you are not necessarily aware of the existence of a weakness in your system which would lead into a vulnerability. For example injecting code into your system database by someone is a threat but we don’t know about an available injection problem within our system. If there is one then it is the weakness of our system. You can’t remove a threat completely but you can minimize your system exposure to it.

In this chapter we are going through an overview of threat modelling. There is two important types of threat modelling. The first one is business threat modelling and the second one is technical threat modelling.

### 3 Threat Modelling

In general threat modelling is about identifying the things that can go wrong within our system. It is an approach for analysing the application security. By using threat modelling we can identify, quantify, and address the threats within our application. Detecting vulnerabilities is as important as being able to tell its associated level or importance to the business. It is possible to identify vulnerabilities and prevent them in the early stages of development by using threat modelling.

Threat modelling is a process consisted of five steps as shown in the Figure 1. Identifying security objectives, survey the application, decompose it, identify

\(^1\)ITSEC is a standard used by a number of European countries first publish in 1990 by France, Germany, Netherlands, and United Kingdom. It is a standard set used for evaluating security among systems and products.
threats, and identify vulnerabilities.

Figure 1: Threat modelling process

Identifying security objectives implies that every team related to a project like project management, software development, and quality assurance groups should have a clear idea about the security objectives. After identifying the security objectives in application overview section we should analyse the application design to find components, data flows, and trust boundaries. In the next step by looking into the application architecture we should select the modules and features that have a security impact on the application. It is also necessary to evaluate the impact on the system in decomposition step. In the forth step we should demonstrate the threats using models and tools such as STRIDE to have a structured view of the known risks. Finally based on the structured view of threats we can identify the vulnerabilities. You should keep in mind that threat modelling is an iterative process as shown in the picture and should be done throughout the whole application development process.

There is two main threat modellings. Business threat modelling tries to cover
threats that can affect the company business in some way. Business threat modelling looks into the value creation aspect of the business. It is important to estimate the severity of a risk for the business in threat modelling. Technical threat modelling focus more on the architecture and the design of the application which we will go through in detail later. Business level threat modelling is only addresses threats that affect the business. If the value creation logic of the business is clear then it can help the technical risk analysis, so we can use the result of the business threat modelling as a preliminary input of technical threat modelling. It should be done with direct contact with the customer or at least with product manager or someone that understands the business logic of the product. In business risk analysis use cases and misuse cases are produced that addresses the business worries. It is done by identifying the role of system in business value creation.

One of the main sources for visualising threats during threat modelling process is the threat trees. The problem with these tools is that for real world projects they get over complicated and they end up looking more like graphs but not trees. It is suggested to use them in order to note down the business level threats from the business threat modelling as an input for technical analysis. For technical threat modelling it is better to draw pictures in the process. By drawing pictures everyone can participate in the process and it prevents pre-drawn pictures which are the result of only creators opinion. The advantage of drawing pictures in the process is that everyone can take part in it. Therefore, at the end everyone has a better understanding of the design and the threats by participating. Data flow diagrams and message sequence charts are good approaches for drawing things. Data flow diagram shows the processing blocks and available data flows between them. (Here I will go through the concept more by providing pictures)

### 4 Technical Threat Modelling and STRIDE

Technical threat modelling identifies risks that are not necessarily business level risks. The information that should be analysed in technical threat modelling are organized in three categories. The first one is data processing. We saw in the previous chapters that processing data can be a point of failure in security. If the processing somehow breaks, there's a possibility that attacker take the control
of the whole processing entity. So it is important to identify these controllable processing blocks in the system in order to prevent such attacks. The second category is the data movement which has two types: Data in motion and data at rest. The data flows and the ways that data is stored should be identified. Finally the last category is about interfaces. Every person interact with the system through an interface. Therefore, the attacker also interact with the system through on of these interfaces. This is the reason why we should also identify each interface within our system. The sum of all interfaces results in the attack surface.

In real life the functionalities of a system may not be clear and the developers use abstractions. Usage of abstractions are essential at the beginning of the development but they may hide the wrong assumptions. Therefore, it is important that during the security analysis and threat modelling we should break these abstractions into detailed elements that are meaningful in terms of security clarity. We use black boxes at the start to show the abstracted functionalities but we should break them into meaningful elements so that each element show how something is done or what is actually done at this point. At the end we should clearly know about the following elements. The points at the system that process data should be broken in process level elements. Statically linked libraries and dynamic libraries and plug-ins should have been listed and identified. Framework, underlying technologies like virtual machines, and the languages they have been implemented in should have been identified. Every data storage point, their flow in the system and how they are being transmitted should be clear. Another important aspect creating data flow diagrams are security boundaries. A barrier between two processing blocks that is enforced over these blocks is called security boundary. Whenever the elements in a box do not trust each other we should split them.

After drawing the data flow diagram we should find the appropriate security service for each discovered data store and data flow but we should keep in mind that not every one of them needs all types of security services. For example public information can benefit from integrity protection services but we don’t need to provide confidentiality services. Microsoft security development life-cycle has a system called STRIDE for these purposes. STRIDE is a classification scheme system developed by Microsoft. It is used to characterise known threats. Each letter in the name stands for a characteristic. Spoofing identity, tampering with data,
repudiation, information disclosure, denial of service, and elevation of privilege. STRIDE can be used as a discussion between the group members for data flow analysis. The group should focus on each data flow or data separately and do a technical discussion on them. The group should start analysing from the highest protocol layer and then move to the lowest ones and they should try to ask questions at each step. The questions should be in the mentioned categories. Spoofing identity aims at the authentication like impersonating someone else, for example users should not be able get the attributes of another user. Tampering with data is about the data integrity. Repudiation is claiming not to have performed an action. Information disclosure looks into authorization of the user, information should not be exposed to someone not authorized to see it. Denial of service is about the availability of the service to authorised users. Elevation of privilege is about performing actions that someone is not authorised to perform for example by injecting code into the system.

5 Conclusion

Threat risk modelling is an essential approach for developing a system. By doing threat modelling we can have a better design and architecture model before the implementation. In technical threat modelling we identify a broad range of threats in our system. These threats should be analysed in order to know their impact on the business logic of the system. We should know when to stop covering threats and which threats should be addressed in our design.

References


bastopol, CA. O’Reilly


[5] Software Security Course Lecture Notes at University of Helsinki
https://wiki.helsinki.fi/display/swsec/Software+Security