Universität des Saarlandes
Lehrstuhl für Softwaretechnik
Prof. Andreas Zeller, Stephan Neuhaus
Design of Secure Software Systems
Exercise 1

1 Risk Management

1.1 Introduction

The difficult thing about risk management is not to execute the algorithm; rather, it's estimating the probabilities and the cost. These are non-technical decisions, and difficult ones, too. This exercise explores some of them.

This exercise has no "correct" solution. Rather, one of the objectives of this exercise is to make you think like a security professional. Whereas the sales manager thinks about all the great things that you can do with all the nifty software that you can use, the security professional always thinks about what can go wrong.

Another objective is to show you the variety of different opinions and solutions (all of them reasonable) that can result from a security analysis. There are usually no absolutely correct solutions.

In this exercise, you will be asked to assign probabilities to events that you cannot estimate realistically. In fact, the material from the lecture will not be sufficient to solve this exercise. This is normal and very much like real life. But don't despair! The exercises to the other lectures will not be as "soft" as this one.

1.2 The Exercise

Your company has written a service where managers can access their contact data online. In effect, you store your customers' address books and make them available to the customers over the Web. The basic architecture contains a number of Web servers, which are linked for load balancing, and a number of replicated database engines that store the customers' address books. You offer no other services.

There is just such a service at `http://www.plaxo.com/`.

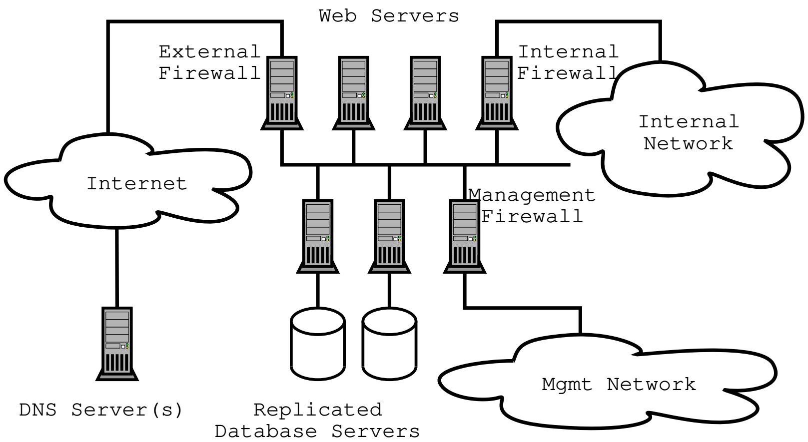

Figure 1: Architecture of Web Contact Management Service

Assume that the architecture of this system is as depicted in Figure 1.

- Make recommendations for the hardware and software (operating system, database system, web servers) in the DMZ; base your recommendation solely on security aspects. Assume that the firewalls are as restrictive as possible, without actually changing the payload of IP packets (i.e., there are no proxies involved).

- In what language would you write the application itself in order to make it as secure as possible? Why? Consider as many languages as you know. For example, C/C++ with Perl; PHP; Enterprise Java Beans; Java Server Pages; C#; VB.NET.
- Identify as many possible risks as you can of operating such a service. Let yourself be guided by CIA.
- You now have the components that make up the software system. Based on your experience (or hunches, even) with these components, and based on your idea of how such a software would be built, estimate the probability that the risks from the previous item actually do occur.
- Assign occurrence and mitigation costs in Euros to each of the risks.
- Now compute the severities and rank them. Surprised?
- Your CEO wants to add a forum to the services that your company offers. What additional risks are there, and how would you counter them?

2 Social Engineering

You want to find out your co-students' grades in the latest Computer Science exam, given by Professor X. However, you are not authorized to know them. There is a public list that maps a student's ID to the student's grade, but you don't know which student has which ID. Name-to-ID mappings and ID-to-name mappings are maintained by many administrative units in the University. For example, the Prüfungsamt has one such list. Professor X and his secretary have a list for the exam that you are interested in; and your co-students themselves know their student IDs. Assume that simply asking for the ID will get you nowhere.

- Devise at least two strategies to get the information by social engineering.
- Devise ways to defend against your attacks. Demonstrate that your attacks would now fail.
- People are naturally lax in following security procedures. What can you do to make sure that your defenses will actually stay in place once you've enacted them?