CS261 Fall 2007
Professor David Wagner Scribe: Peter Lau

Lecture 1 – August 28, 2007

Overview

What is computer security? Let us first consider a more familiar subject – cryptography. Ron Rivest defines cryptography as the problem of communication in the presence of adversaries. Similarly, computer security is the problem of doing computer science in the presence of adversaries. Note that both subjects deal with malicious adversaries. This is the key difference between computer security and the related area of computer reliability, which deals with proper operation of a computer system in the absence of adversaries.

In this class, we will learn how to build and defend systems against attacks. But to do this, we first need to know how to attack systems. Although most attacks are fixed, some may still be unpatched, so this raises ethical issues. Do not use your knowledge to break into systems that are not your own. We will first discuss what a security analysis entails, then go through a simple example.

Security Analysis

If we were given a new system we have not seen before, how do we perform a security analysis of a system? Generally, we can approach the task in 3 steps:

1. **Define the scope of the problem**
   a. What are the security goals you are trying to achieve?
   b. What resources are you trying to protect? In computer security, we are usually trying to protect information or data of some sort.
   c. What do we want to ensure? In computer security, we are usually trying to ensure confidentiality, integrity, and availability.
   d. In this class, we will generally not focus on the details of this step.
2. **Develop a threat model**
   a. What threats is our system going to be exposed to? Given a set of goals, we need to consider all the ways in which adversaries could violate our goals.
   b. What are the attacker's capabilities and limitations? Adversaries have different levels of expertise and motivations for attacking our system. No system is 100% secure against all possible adversaries, so to keep the model manageable, we should limit the list of our attackers by the scope of the problem defined in the previous step. We may also rule out certain adversaries because it may be unlikely that they will attack our system. For example, a professional bike thief is probably not going to steal a toddler's tricycle because it is not worth his time.
   c. First, be as complete as possible in creating this model by considering all possible scenarios. Then afterwards, begin to narrow down the threat model based on the scope of the problem.
   d. Together, steps 1 and 2 help you understand the problem domain, and it generally does not involve many technical details.
3. **Security Analysis**
   a. We need to decide given the scope of the problem and the threat model, does the system achieves the security goals we listed earlier?

b. The final step is the hardest and there are no clear guidelines as to how to perform the security analysis. Unlike the first 2 steps, this one is usually very technical.

Example Security Analysis: Bike Security

Consider the problem of keeping your bike security. After riding your bike to work every day, you lock it to some bike stand. Our system here would consist of the bike, the bike lock, and the stand to which it is locked to. If we were performing a real security analysis of a system, we would probably need the schematics for each part of our system and consider the general surroundings. However, since the goal of this example is to demonstrate the steps required in a security analysis, we will overlook these specifics and concentrate only on the high-level ideas.

Security Goals

What do we want to ensure? Most importantly, we would like to have full use of bike at any time. Our bike is no good to us if the bike or any of its pieces are damaged or missing, so we wish to ensure the bike's integrity. Even if someone does not walk off or attack our bike, its availability to us may still be compromised. An ex-boyfriend/girlfriend could lock the bike to the stand with his or her lock, which we do not have the keys for. If we parked the bike in an area with lots of construction (such as UC Berkeley), we may find that the construction crew has closed access to the bike stand. Thus, we would like to ensure both the bike's integrity and availability to us.

Our second goal may be to maintain the resale value of the bike. Looking at aesthetics, a bike covered in spray paint or dirt will certainly meet our first security goal, however, we would like to keep it looking as good as possible, even if we do not plan to sell the bike in the future.

Threat Model

To develop a threat model, we need to consider our possible adversaries. Since our current example deals with physical security, we may want to consider an attacker's physical characteristics such as strength and dexterity. In computer security though, an attacker's physical characteristics are less important than his or her technical skills. To keep our example more in line with computer security, we will consider the attacker's bike theft expertise. For each of the possible attackers, we can list his or her available tools and methods for stealing or damaging bikes. Note that we should concentrate on the attacker's abilities, but not list out the possible methods of attack. Such a list would probably be very long and incomplete. By considering only the set of attack capabilities in our threat model, we subject our system to more avenues of attack.

Returning to our example, we may want to consider the following qualities of our possible adversaries:

- **Expertise**: The attackers may have varying degrees of general, technical skills: none, some, and advanced, and varying degrees of bike theft expertise.
- **Tools and Methods**: The attackers may have no tools, basic tools, or power tools. He or she may also employ social engineering to gain other avenues to attack your bike.
- **Access**: The attacker may have access to the keying machine for your bike lock model. He or she may have insider information on the lock and know how to disable it without a key. The attacker may have obtained a spare key to your lock through a friend.
- **Motivation**: The attacker may be motivated by different reasons: economic, revenge, ego/pride, or necessity. We consider this because it helps us judge the attacker's capabilities.

Looking at the spectrum of our possible adversaries, we can see that it becomes more costly to secure our bike against a highly motivated and capable bike thief. We can limit the scope of our threat model by applying basic economic reasoning. Adversaries will not attack a system if it costs more to bypass security measures than they have to gain. For example, a highly motivated bike thief is unlikely to invest $200 to break your $20 bike lock and steal your $100 bike. In our example, we may choose to protect our bike from at most casual attackers with no bike-specific tools.

Security Analysis

Although there are no clear-cut steps to follow in a security analysis, we can begin by first performing a reliance analysis and then evaluating it.

Reliance Analysis

In a reliance analysis, our goal is to determine what set of components and properties we depend on to ensure the security of our system. This set of components is called our reliance set; if any component in our reliance set fails, our system may easily be compromised. Going back to our bike example, the bike lock and bike stand are the most obvious members of our reliance set. If our bike is equipped with quick releases, then they belong in our reliance set as well since we named integrity as one of our security goals. (Our bike would not be much use to us if it were missing wheels or the seat.) The bike frame may also be in the reliance set, depending on how easy it is to disassembly or break. Note that our reliance set may change if the security goals or threat model changes. If we did not include full availability in our security goals and we considered only attackers without tools, we would not include the bike frame in our reliance set (assuming the frame was relatively solid).

After determining the components in our reliance set, we need to decide what functions and properties of each component we depend on. Some components provide multiple functions, but we might rely on only one and not the others. We can illustrate this point with a real-life example – A parent trusts a bank with her money, but not her children, and trusts her relatives with her children, but not her money.

Going back to computer security, a reliance analysis usually involves looking at a large architectural diagram and circling the components in the reliance set. One requires technical expertise to know how the components function and how the components depend on each other.

Reliance Evaluation

In our reliance evaluation, we need to decide given the reliance set and the properties we are dependent on, is our trust well placed? We may need to consider the specifications of the components and the capabilities of an attacker. This is hardest part of a security analysis since no general guideline exists.

General Points on Computer Security

Some closing remarks regarding the field of computer security:

- 3 common security goals are in computer security are to ensure:
  - **Confidentiality** – We wish to prevent others from gaining unauthorized access to our data. For example, we do not want an attacker to read our emails or be able to infer anything he or she would have otherwise not known.
  - **Integrity** – We want to ensure that no attacker has tampered with the data we received.

    - **Availability** – We want to ensure that we have access to all of the computer system's functions and processing ability. This includes CPU cycles and network access.
- Although the word "*trust*" used in regular language connotes a positive response, the opposite is true in computer security. When a component is "trusted", it is a member of the reliance set, and we are dependent on it for the system's security. Ideally, we wish to reduce the number of trusted components so the system has fewer points of failure.
- In computer security, a *trustworthy* component is one that has no or is free of defects.
- *Trusted Computing Base* is another term for the reliance set.
- *Transitive Trust* occurs when one entity trusts on another, which in turn trusts on another. If security is involved, then the first entity depends on a third party that he or she may not trust. Unlike mathematical equality, trust is not transitive.
- The problem of *Trusted Path* occurs when two parties trust each other, but not the communication channel between them. A closely related security goal here is authentication – how do you verify the identity of whom you are communicating with?
- Threat modeling and security analysis are essential aspects of computer security. In this class, we will study the latter in detail.
- Computer security is usually about economics because an attacker will not break into a system if the cost of breaking into it is greater than he or she is willing to invest.