Chapter 13 – Security Engineering

By Michael Almandeel

13.1. Explain the important differences between application security engineering and infrastructure security engineering.

-Application security is about implementing an application in such a way that it resists attacks, with the goal of controlling the set of ways a program is capable of executing, ie behaving. Attacks on applications are generally aimed toward influencing a program to behave in unintended ways.

-Infrastructure security is about configuring the various elements of an organizations entire it infrastructure(generally c.o.t.s products) to resist attacks. This configuration also has the goal of controlling the set of ways the system can behave, but here we are talking about the collective system (or subsystem) as a whole resisting an attack rather than a specific application or element within it.

13.3. Explain why there is a need for both preliminary security risk assessment and design risk assessment during the development of a system.

The preliminary risk assessment stage is used to gain a general understanding of the various security risks associated with a system, and the potential costs associated with each of these individual risks. Later in the design risk assessment, this information will be used to further define exactly what the vectors of attack are, and how to implement the code to protect against them. The potential costs calculated in the previous activity will be used to prioritize which security features are most important and therefore require the most attention.

13.5. Explain, using an analogy drawn from a non-software engineering context, why a layered approach to asset protection should be used.

I would like to use the analogy of a state prison. In this case, we will assume a prisoner escaping is equivalent to an unauthorized user gaining access. They are designed with several layers of containment a prisoner must breach before escaping:

-their cell

-their cell block

-the general area which connects these blocks and leads to the main entrance

-the yard, which is protected by armed guard towers and tall barbed wire fences

Because a prisoner must negotiate each of these layers before they can escape, it both buys time for a response to prevent the prisoner from escaping further, and presents the option that a prisoner will encounter a layer that he cannot breach or bypass. Contrast this with a yard containing sets of self contained cells without the protection provided by cell blocks and the general prison itself. It would make escape much easier.

13.6. Explain why it is important to use diverse technologies in the development of secure systems.

If one were NOT using diverse technologies, finding a valid attack vector for any part of the system infrastructure would effectively put the entire system at risk. The reason we use diverse technologies in the context of security is to compartmentalize risk , and isolate the damage that can be done by any given attack vector. Naturally, many more attack vectors will be needed to take an entire system down vs its non diverse counterpart.

13.8. Explain why it is important when writing secure systems to validate all user inputs to check that these have the expected format.

User inputs are one of the main attack vectors used by hackers. The hacker attempts to inject a string into the user input that will cause the the program to behave in an unitended way, most notably by causing a stack overflow, but many sql and script injection attacks are done this way as well. By restricting the set of input strings to those that cause your program to behave as expected, it prevents malicious users from executing these attacks.

13.9. Suggest how you would go about validating a password protection system for an application that you have developed. Explain the function of any tools that you think may be useful.

Before we discuss validation of the system, lets discuss what one might look like:

-It would store in a secure location a file that contains the hash associated with each users password

-It would store in a secure location a file that contains the access permissions associated with each user -Logging in would consist of checking that the hash of the password entered by the user matches the stored hash, and generating a temporary token for the user, which will have some specified time of validity(ie after 2 hours, token will be invalid).

-The user uses the provided token to request access to resources, and if the system deems that token valid and authorized to access said resource, the resources will be provided,

To validate each part of this system we might:

-ensure the hash mechanism works as intended, as well as trying various methods for gaining access to and overwriting the hash file

-ensure the access permission mechanism works as intended, as well as trying various methods for gaining access to and overwriting the user permissions file

-ensure that the process of logging in and generating a token works as intended, as well as trying various methods to generate malicious tokens and by either uploading them to the server, or gaining access to the generated token from the server in transit to the client via a man in the middle attack.

-ensure that resources are only provided to the users who should have access to them, by using the token to authorize the resource request. Various methods will be tried to coerce the system to provide said resources without a valid system generated token, either by generating an appropriate false token, or discovering a mechanism to obtain the resources without providing a token at all.