# Lab: Introduction to Threat Modeling

During this lab, you will be introduced to the concept of Threat Modeling. This introduction will help you understand the need for threat modeling and will also introduce you to the Microsoft® SDL Threat Modeling Tool v3.

Estimated time to complete this lab: **25 minutes**

###### Before You Begin

There are no prerequisites to this lab.

###### What You Will Learn

After completing this lab, you will have an introductory understanding of Threat Modeling.

**Exercise 1: Review the Introduction to Threat Modeling Slides**

In this exercise, you will review a set of slides designed to give you an introductory understanding of Threat Modeling.

#### Start and log on to the Vista virtual computer

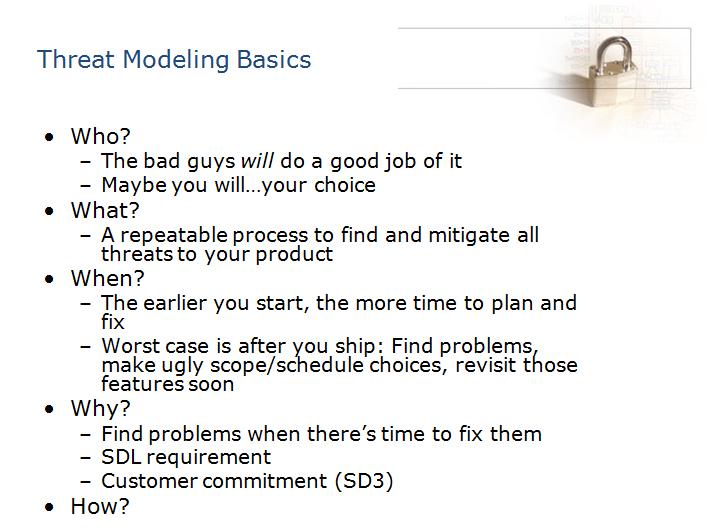
1. In Microsoft Virtual PC, right-click **Vista1** and then click **Start**.
2. At the logon screen, press and hold the right <ALT> button and then press <DELETE>.
3. Log in as administrator with a password of **pass@word1**
4. If you get any errors about failed services, open the Services MMC and verify that all services set as Automatic have started.

Do not start any other virtual computers until this server is completely up.

#### Open the PowerPoint on your desktop

1. On your desktop, double-click the PowerPoint icon for the Threat Modeling presentation.





A threat model helps determine the security risks posed to a product, application, network, or environment, and how attacks can show up. The goal is to determine which threats require mitigation and how to mitigate them.

*Who is involved?*

The bad guys will do a good job of finding the security vulnerabilities—if you choose to do an adequate threat model then you just might thwart them.

*What is threat modeling?*

Threat modeling is a repeatable process that helps you find and mitigate all of the threats to your product.

*When should you threat model?*

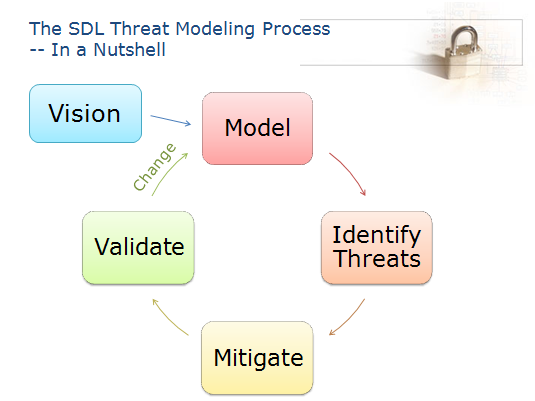
Starting the threat modeling process early in product development will allow more time to plan for and fix threats.

*Why should you threat model?*

Threat modeling will help you find problems while there is still time to fix them. It is also a Security Development Lifecycle (SDL) requirement and allows you to fulfill your commitments to the customer.

*How?*

The SDL Threat Modeling tool was created by Microsoft, to ease the threat modeling process.



This diagram represents the Threat Modeling Cycle and the set of tasks involved in creating a threat model.

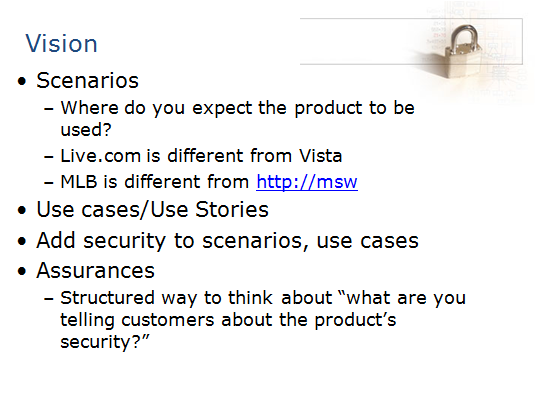
At a high level, Threat Modeling consists of a number of activities done during the design phase of product development. These activities begin with envisioning the product as it will be used by typical users in a typical environment, and then identifying all of the potential threats to the product and to assets accessed via the product.

Next you build the threat model by creating a number of Data Flow Diagrams (DFD), which model the trust boundaries of the application and its components and the flow of data between the application and its environment as well as the flow of data between components within the application.

Knowing what needs to be protected, and how it will be attacked, enables you to choose appropriate mitigations for each threat.

Finally, we need to review the threat model, the DFD elements that need protection and the mitigations or defenses/countermeasures, to ensure that the mitigations do indeed protect the assets from the threats.

Threat modeling is an iterative process, which should be started early during the product design and updated throughout the lifecycle. As your product develops and you change the design, you will need to cycle back and revisit the threat-modeling process to verify whether the design change will enable any new threats.



The first set of tasks is to identify the various use cases and usage scenarios for the product and to determine what security assurance about the product is being provided to the customers.

These initial tasks provide a higher-level perspective of the product use and help you determine the entry points, trust boundaries, etc.

Determining the security assurances helps you set a benchmark that can be used in identifying the threats against the product. Verifying whether the security guarantees have been met can help determine the risks the product faces.



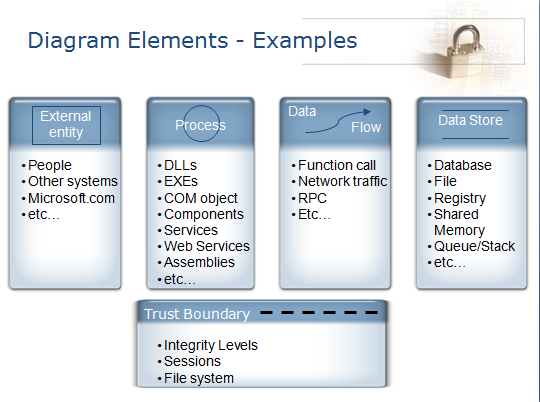
The first step in modeling is to decompose the system into its key components. This can be achieved by creating Data Flow Diagrams (DFDs). These diagrams depict various elements such as processes, data stores, data flows, trust boundaries, etc.

The important step here is to identify the trust boundaries. A trust boundary is a location where the level of trust changes. Identify the systems, subsystems, and identities that your application trusts.

The following are some examples of trust boundaries:

* The perimeter firewall.
* The database server trusts calls from the Web application's identity.
* The data access components trust the business components to pass fully validated data.

These diagrams are not static and need to be updated as the product evolves.



DFDs are a graphical representation of the various processes, data stores, data flows, and trust boundaries involved in a system.

Here we see the elements that make up a DFD and the symbols used to depict them.

The procedure of creating DFDs is to start with a simple, high-level DFD that has just a couple of processes and data stores and keep building upon it. This is sometimes called a ‘context’ diagram.

The first and highest level DFD is the context diagram which models the system and its interaction with external entities. In some older diagrams, the product is represented as a multi-process or complex process (two circles, one inside the other).

At this point, how the application works internally should be ignored.

The focus should be on what requests the system processes, what responses it generates, and the data sources with which it interacts.



Once the application has been decomposed into components or assets (threat targets), the next step is to determine the categories of threats that each component is susceptible to.

There are six threat categories that can be applied to the components. Use the acronym “STRIDE” to remember them:

**S**poofing – an attacker can use another user’s or machine’s identity to access restricted information

**T**ampering – making unauthorized modifications to data

**R**epudiation – denying having performed a malicious activity

**I**nformation Disclosure – exposure of restricted or sensitive data to unauthorized users

**D**enial of Service – denying service to valid users e.g. crashing a server

**E**levation of Privilege – Unauthorized users gaining restricted privileges to perform malicious activities

Understanding what security property is associated with a particular threat can guide you in choosing the appropriate mitigation technology.



Here we look at how the STRIDE categories of threats are applicable to the various elements of a WRD.

For example, data in transit can be tampered with (Tampering), revealed to unauthorized users (Information Disclosure) and prevented from being available to valid users (Denial of Service)



Once the threats for each component in the system have been identified, the next task is to apply mitigations to those threats.

Mitigation techniques should be identified for each type of threat discovered. Ways to do this include applying domain knowledge, checking lists of standard mitigation approaches or researching techniques applied by similar software packages. Examples of mitigation techniques are authentication, authorization, etc.

In addition to applying mitigation techniques, the product can be redesigned to eliminate the vulnerability. This may require more time and effort, but might be useful in cases of severe threats.



In addition to researching and applying mitigations that have been used by others, you can invent your own mitigations for the issues found in your product.

This is, however, a challenging task, and needs to be done with utmost care because failure to apply proper mitigations isn’t obvious and sometimes will only be detected by a security expert. If this expert turns out to be the adversary, then your product will be exploited!



Once the application has been decomposed into its components, threats identified, and mitigations applied, the final step is to validate.

The validation task includes the following:

* Check whether the Data Flow Diagrams match the final code. If the design has been modified since creating the DFDs, they need to be updated accordingly.
* Verify whether STRIDE threat types have been identified for each component of the DFD, especially those that touch a trust boundary.
* Check whether proper mitigations have been applied to the threats identified.
* Check that all the above tasks have been accomplished before shipping.



During the threat modeling process, it is important to also consider the code that your application is dependent on.

Spend some time reviewing all the licensed components, shared DLLs and other libraries developed by a 3rd party that your application uses from a security perspective. Vulnerabilities in those components can also result in your application being exploited!

Also verify the validity of the assumptions made in implementing the application. Security problems are usually a result of invalid assumptions made about the product’s implementation and usage.

Since threat modeling is an iterative process, it is possible to keep repeating the various tasks and updating the threat model until the product is finally shipped.

For this reason, it’s extremely important to know when the threat modeling process can be considered complete. Validating the following helps you determine when to stop:

* DFDs match frozen code.
* Threats have been identified and mitigations applied to each component in the DFD.
* The mitigations are validated by QA by testing whether the application is still vulnerable to the threats for which the mitigations are in place.
* When the threat model has been reviewed and approved by an external security expert.

## Exercise 1: Testing Your Knowledge

#### Questions

1. What is the purpose of a threat model?

2. When should you begin the threat modeling process?

3. What is the first activity you should do during the threat modeling design phase?

4. What is the benefit of determining the security assurances?

1. An important step in modeling is to identify the trust boundaries. What is a trust boundary?
2. What is a data flow diagram (DFD)?
3. What is the definition of the acronym STRIDE?
4. How do you identify mitigation techniques for each type of threat discovered in the threat modeling process?
5. What is the final step of the threat modeling cycle?
6. Why is it important to verify the validity of the assumptions made in implementing your application?