Effective Threat Modeling – using TAM

In my [blog entry](http://blogs.msdn.com/ace_team/archive/2007/05/01/threat-modeling-sanity-check-list.aspx) regarding – **T**hreat **A**nalysis and **M**odeling (TAM) tool developed by (**A**pplication **C**onsulting and **E**ngineering) ACE, I have watched many more Threat Models being built to either check a box on the development checklist or were sincere attempts to understand the threat profile of the application. Most of those were left wanting much more than was produced.

Stepping back a bit, let’s take a quick look at the history the TAM tool.

ACE team released Threat Modeling tool code named Torpedo internally to (Microsoft) MS in 2004. This was v1.0 which was to be used for all the applications developed for MS IT (Microsoft Information Technology). Over 600 TM’s were created using this version but because its target users were security experts soon there weren’t enough experts to churn out the TM’s at the required speed. This version was very useful in finding many design issues with MS internal apps, but sheer size of operations and lack of sufficient security experts demanded another look at the situation.

V2.0 methodology and tool was created to simplify terminology, and easier methodology, revamped look and feel were some of the highlights of v2.0 release. Version 2.0 was released for people to download on the [MSDN](http://msdn2.microsoft.com/) (Microsoft Developer Network) site in 2005. After the first release there has been continuous rising interest in the TAM tool and more and more people inside MS as well as external customers started using TAM. This was also a huge milestone for [ACE Services](http://blogs.msdn.com/ace_team); as it was available for free [download](http://msdn2.microsoft.com/en-us/security/aa570413.aspx). This exposure to wider consumption resulted in customers wanting training and more material around it. Ford and Boeing are some of the external customers who are currently using TAM v2.0 in their internal SDLC processes.

These enhancements have helped in building a lot of awareness around TAM. I have come across practitioners from various disciplines who want to start doing threat modeling to get a view of the possible threats to their system. To begin with, threat modeling using TAM appears to be a very simple and straight forward process right from downloading the tool to producing a feature rich Threat Model. This apparently simple and effective process does warrant certain care and due diligence in order to build a good threat model. Some of the considerations for effective threat modeling are as follows:

1. It is vitally important to have access to people and information pertaining to all aspects of the application there is a significant involvement of non technical personnel in the process.
   1. The TM process starts with capturing business needs or objectives of the application and continues through the development and maintenance phase of the application. Maintaining the Threat Model becomes an ongoing part of the application’s lifespan to account for new and emerging threats and attacks. However, the majority of the TM work is done in the early stages of development, before any code is written. This provides a strong proactive approach to building secure software and prevents costly rework due to retro-fitting security requirements when security bugs are discovered late in the process. Thus the TM process calls for seeking input from business owners to help categorize, and rank the threats identified.
   2. Most of the business owners who sponsor the app development have very little time on hand; so to get a slot on their busy schedule one needs to have good relationships with people who “matter”. This involvement helps the technical group to translate technical risk into Business Impact, which then provides a greater understanding at the business levels so they support the process.
2. Use security principles such as usage of least privilege , reduction of surface area etc to verify assumptions and information as provided by the team
   1. Certain threat categories are not directly evident such as not enforcing either the service accounts or user accounts with least privileges to invoke the code. This may lead to either users or services, which if compromised, expose lot more than just the application code and data to the attacker. TAM proves to be a very powerful tool in such a situation by providing us with data access control matrix. This unique representation allows database architects and application architects to ensure no unauthorized access to data.
   2. It provides a way to very explicitly identify the access and privileges that roles need. They should not be given any other right and privileges than the ones described in the Data ACM.

Figure 1: Data Access Control Matrix

* 1. For attack surface reduction the “attack surface analysis” tool will be of great help. This will help you understand various ways by which any particular component could be accessed. For example for an online store the diagram below illustrates the possible ways to access the database.

Figure 2: Attack surface analysis for an E-commerce web site

1. Use cases should cover a variety of possible actions that an application user or system user could perform.

Use cases need to represent the application from security perspective and may not cover all the different ways of accessing the assets of the application.

Use Cases represent the way in which users and components interact. Ideally you want to ensure that all of the access requirements you specified in your Data ACM, is made possible though one or more Use Cases. However, it is up to your discretion if you want to model the Use Cases based on role access, level of privilege access or data access.

1. Components should have appropriate relevancies identified

The components are related based on the technology or the implementation of it. This is what provides very important information for determining the susceptibility of the component to various targeted attacks on the known weaknesses or usual mistakes made while implementing that technology or architectural component.

1. If you have certain components which use technologies not available for selection in the drop down while documenting the component profile, you can add that technology by going to technology drop down under ”Tools”->”Options”->”Metadata editor” menu. Similarly you can add authentication mechanism, service type and data classification and approximate number of identities in a role by way of weight.