

## Effective Threat Modeling – using TAM

In my [blog entry](#) regarding – Threat Analysis and Modeling (TAM) tool developed by (Application Consulting and Engineering) 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](#) (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](#); as it was available for free [download](#). 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.
  - a. 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.
  - b. 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. Ideally the application architect/lead developer role is expected to perform the actual threat modeling:

a. Threat modeling calls for collecting information such as business goals which are to be achieved through the application, to fleshing out a myriad of details such as roles for users and services, the number of users in each role, and components in the system. All this information may not be available with an individual developer. Therefore Threat Modeling is best performed by architects or lead developers in the team. Note that developers still play a role in the process by implementing identified countermeasures during the development phase.

3. The default attack library contains a fairly comprehensive list of the known attacks that exists today. It should suffice for most Threat Modeling tasks. However, if a new attack emerges, or if there are some custom attacks that you face in your organization, you can customize the Attack Library to suit your needs.

a. As is with many good tools TAM is also customizable to the environment in which it is used. TAM allows users to add attacks they feel are more relevant to the operating environment of the application or remove attacks that are not. This includes countermeasures and steps to be taken by developers to implement the countermeasure or for testers to test the implementation. The relevancies will help with the identification of suggestions in terms of probable countermeasures applicable to a component based on the relevancy that you have added.

4. Discuss the production environment configuration with appropriate teams; many times development teams do not have sufficient information regarding production environments. The lack of knowledge with regard to deployment scenario such as service accounts to be used and the privileges assigned to these accounts can cause confusion. The classic example is of impersonation, if impersonation is used, user accounts are flowed to the application and authorization is based on that. But if impersonation is not used, the web server identity or the identity of the action invoked by user is visible to the application. If the deployment is not done correctly, the impersonation settings may be incorrect and could result in simple failure of the application, to an accidental elevation of privilege problem.

5. Use security principles such as usage of least privilege , reduction of surface area etc to verify assumptions and information as provided by the team

a. 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.

b. 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.

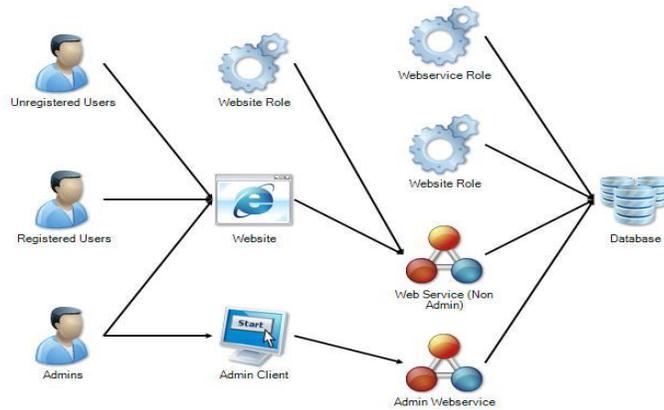
Data Access Control Matrix 

Data	User Roles			Service Roles		
	Admins	Registered Users	Unregistered Users	Database Role	Webservice Role	Website Role
Customer Accounts	C R U D	C R U D	C R U D		C R U D	
Customer CCs	C R U D	C R U D			C R U D	C R U D
Product Information	C R U D	C R U D	C R U D		C R U D	C R U D
Order	C R U D	C R U D	C R U D		C R U D	C R U D
Logs					C R U D	C R U D

 Item has a Use Case defined but no Conditions  
 Item has a Use Case defined and Conditions  
 A Use Case has not been defined

Figure 1: Data Access Control Matrix

c. 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**

6. 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.

7. 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.

8. 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.

9. Service roles performing certain actions across layers of application need to have a corresponding identity defined.

It is imperative that a service has to run with a certain service role and corresponding identity. The service roles and corresponding identities have to be provided to the TAM tool so that you can use it to complete the use cases comprising of multiple hops.

It also provides the team a way to let the infrastructure group know what service roles they need, and what the privileges on those roles have to be.

It also provides a way to track identity changes through the application as well as knowing if any component is going to do impersonation, additionally allows us to identify trust boundaries in the application.

10. Use the analytics to check completeness/ coverage of various avenues for accessing data elements

The analytics provide you with a way to review and audit the information entered into the Threat Modeling tool. There are various views of analytics such as the subject object matrix. This view helps you understand how subjects (roles) interact with objects (Components), essentially this becomes the list of allowable actions in your application. This is a good place to identify if a role should be performing an action on specific object.

Sample Subject Object Matrix is:

Components	User Roles			Service Roles		
	Admins	Registered Users	Unregistered Users	Database Role	Webservice Role	Website Role
Website	1. browse catalog 2. create order	1. browse catalog 2. create order	1. browse catalog 2. create account 3. login 4. create order			
Admin Webservice	1. submit product info 2. gets the product feed					
Database					1. retrieve catalog 2. submit product info 3. save account 4. Saves order information	1. verify credentials
Database : Customers						1. gets the products from
Admin Client	1. add product info 2. Get the product feed from					
Web Service (Non Admin)						1. Gets the catalog data from 2. Saves the account data 3. Verifies the credentials from 4. Saves order information
Log Store						

■ Allowed access  
 Tip : mouse over each allowed interaction to view the supported use case

Figure 3 Subject Object Matrix

11. Similarly there “Component Access Control Matrix” under Analytics menu would help application team evaluate permissions for each of service roles and user roles on various components. This way it can be ensured that principle of least privileges is followed and none of the user roles has more than necessary permissions.

Component Access Control Matrix						
Components	Service Roles			User Roles		
	Database Role	Webservice Role	Website Role	Admins	Registered Users	Unregistered Users
Website				[1]	[2]	[3]
Admin Webservice				[4]		
Database		[5] Customer Accounts CRUD Customer CCs CRU Product Information CRU Order CRU	[6] Customer CCs CRU Product Information CRU Order CRU			
Database : Customers			[7]			
Admin Client			Calls: [Downloads Product Feed] Admin Webservice gets the products from Customers			
Web Service (Non Admin)						

Figure 1 - Component Access Control matrix

12. A use case comprises of multiple hops that complete the user or system action. Often the use cases have some net data effect which could be either of Create, Read, Update or Delete certain data. Document the net data effect that the use case achieves as part of the last call in the multi hop calls till the data store.

13. Seek risk response input from the business users/ stake holders

14. Risk response allows the business team to respond to threats appropriately based on the level of risk. Some threats can be accepted or reduced based on the on the Likely hood and Impact as determined by the Business team (for impact) and the technical team (for likely hood).

Many threats identified by the tool might have to be mitigated outside the development team. For example sometimes mitigations may require using SSL but it may have performance impacts on the application. This calls for attention of application owners.

15. Reports for testers

There are various reports that can be generated using TAM. The "Test Team report" is for testers and is very useful for security testing of each and every use case. It provides testers with step by step instructions and sample test strings to test the application for the identified potential security vulnerabilities and provides guidance regarding security testing of the application.

## Summary:

Major corporations are rapidly adopting proactive approach to security due pressure from the society in general and Federal Government.

Threat modeling, when done properly keeping above points in mind will help immensely in bolstering security efforts by the organizations and provide application owners a sound strategy to avoid common mistakes and achieve the application goals in much more predictable way. This proactive approach will also greatly reduce the risk of budget overshoots due to security flaws to be mitigated at the later stages of the application.

At the enterprise level ACE has Threat Analysis and Modeling tool for enterprise (TAMe), but that is different subject all together and can be discussed in a similar paper.