Threat Modeling Using Microsoft Threat Analysis & Modeling v2.1

Microsoft Application Security

Contents

Microsoft Threat Analysis & Modeling v2.0

Why build a threat model?

The reason for building a threat model is simple: to identify potential threats so that you can build a solid security strategy to guard against them. You cannot feasibly build a secure system until you understand the potential threats against it. It is important to realize that threats do not materialize from thin air; rather, they are the by-product of your own application. This is why it is crucial to first understand your *application context*, before you begin trying to defend it.

What is the application context?

In order to understand your application context, you must understand the individual elements that, together, create it. Defining the various elements of your application individually enables each element to be analyzed and coupled together to define an application context. This makes it possible to identify potential threats, and then systematically build an effective security strategy.

How is this done?

To define your application context, it is necessary to first define your *application requirements*, and then define your *application architecture*. The application requirements consist of business objectives, user roles, data, and use cases, all of which are defined by business owners. The application architecture consists of components, service roles, external dependencies, and calls, and is defined by application architects.

Overview

The core function of the Threat Analysis & Modeling tool is to identify threats, while facilitating the process of defining a security strategy. Even if you are not a security subject-matter expert, you now have the ability to consistently and objectively identify threats to your software application.

Creating a threat model using the Microsoft Application Security Threat Analysis & Modeling tool is a three-phase process. First, you define your application context. Second, you model your threats on top of your application context. Third, you measure the risk that is associated with each threat. Once you have completed these phases, you can assimilate your threat models through analytics, visualizations, and reports.

The Threat Analysis & Modeling tool automatically generates potential threats to your software application, based solely on known information that you provide. The Threat Analysis & Modeling tool also has the capability to assimilate the information you provide to build security artifacts such as access control matrices, data flow and trust flow diagrams, and focused, customizable reports.

Getting Started

Familiarize Yourself with the Interface

In order to successfully use the Threat Analysis & Modeling tool, take a moment to familiarize yourself with the basic functionality of the user interface.

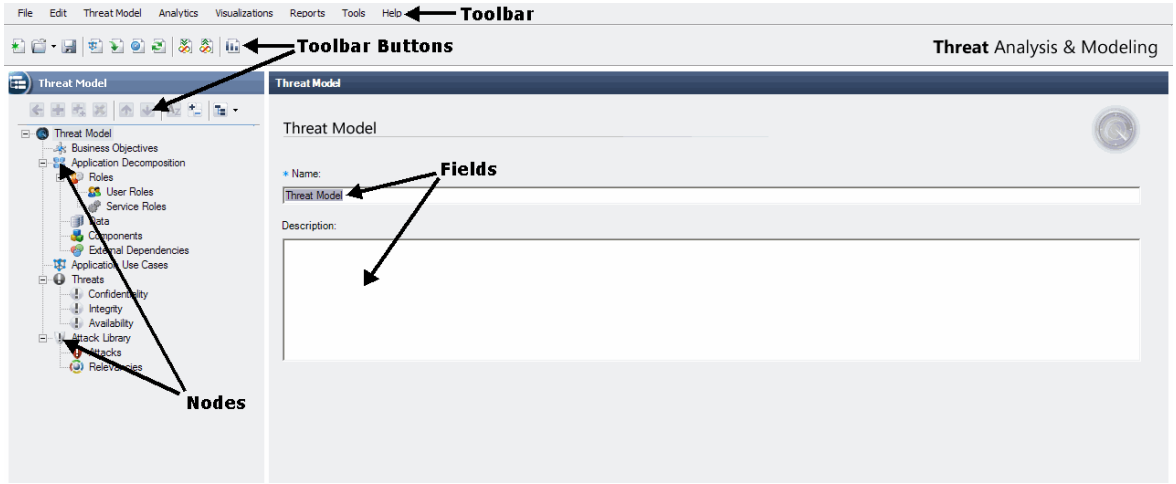
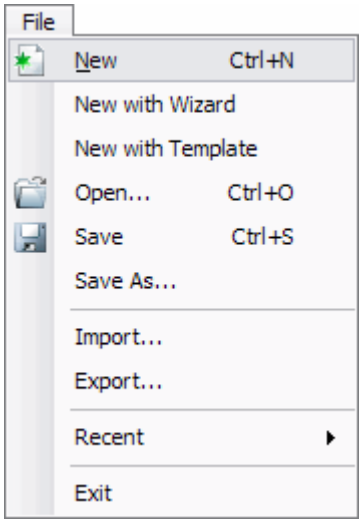


Figure 1. Threat Analysis & Modeling tool user interface

- **Menu options** enable you to navigate through the tool, access graphical representations of your data, generate reports, and import and export your attack libraries.
- **Toolbar buttons** provide shortcuts. Hover your mouse pointer over each button to display a short description of its function.
- **Nodes** represent the structure of your threat model.
- **Fields** are areas where you will type information or make a selection.

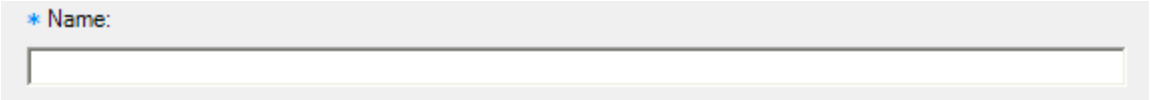
Create a New Threat Model

1. On the **File** menu, click **New**; or click the **Create New Threat Model** button on the toolbar.



- or - 

1. In the **Name** field, type a name for your threat model.



1. (Optional) In the **Description** field, type a description of this threat model.

Define Application Requirements

Application requirements consist of your business objectives, user roles, data and use cases. Before you can begin to define the application requirements, you must first create a new threat model.

Business Objectives

Business objectives are your goals - that is, the reason for creating your software application. Applications are developed to fulfill specific business needs, or to solve some specific business problem. These needs or problems are your objectives that need to be fulfilled by your application in order to benefit your business.

Example: Increase business tempo or increase online revenue.

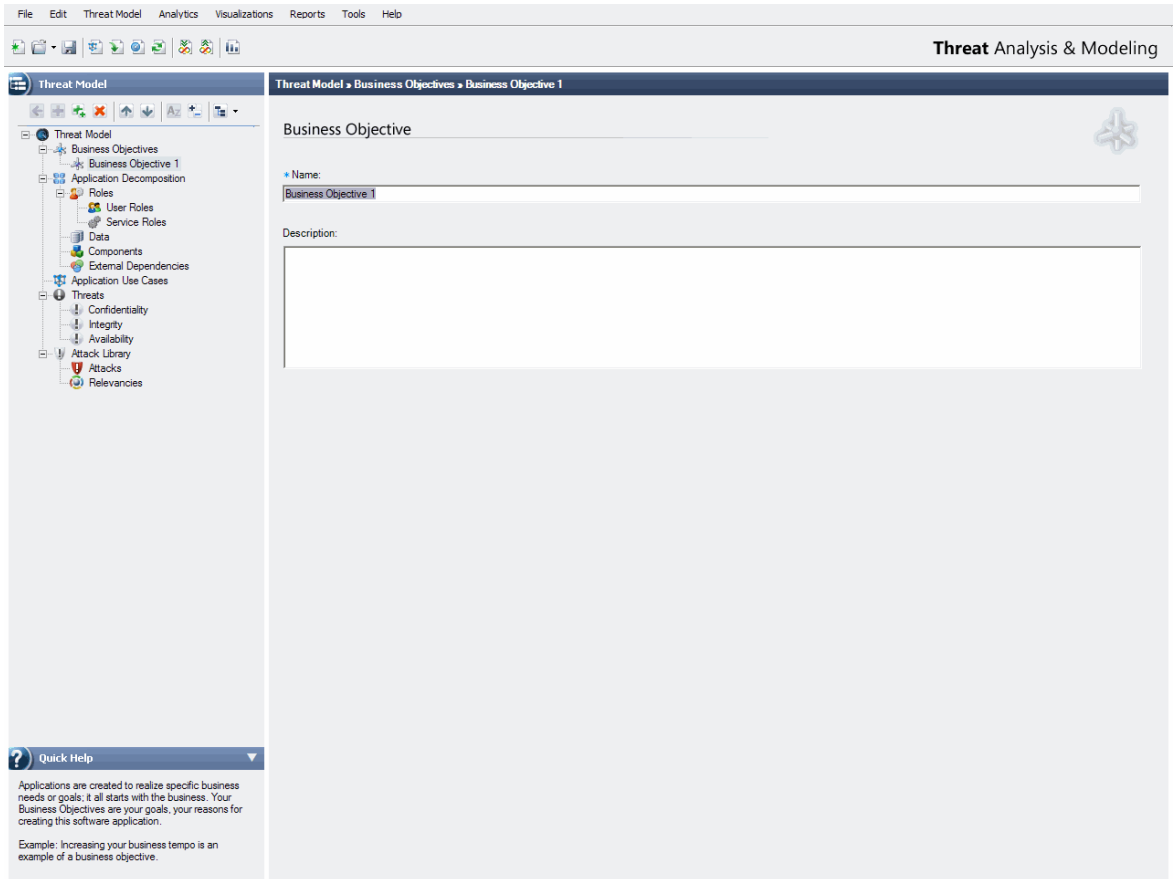


Figure 2. Business Objectives node and fields

To add a new business objective:

1. Select the **Business Objectives** node, and then click the **Add Child Item** button.

1. In the **Name** field, type your business objective.

1. (Optional) In the **Description** field, type a description of this business objective.

User Roles

Roles define the trusts of your software application, and they are primarily used to make authorization decisions. Any user who will be interacting with the application must be assigned a *user role*.

Example: Registered users, unregistered users, and administrators.

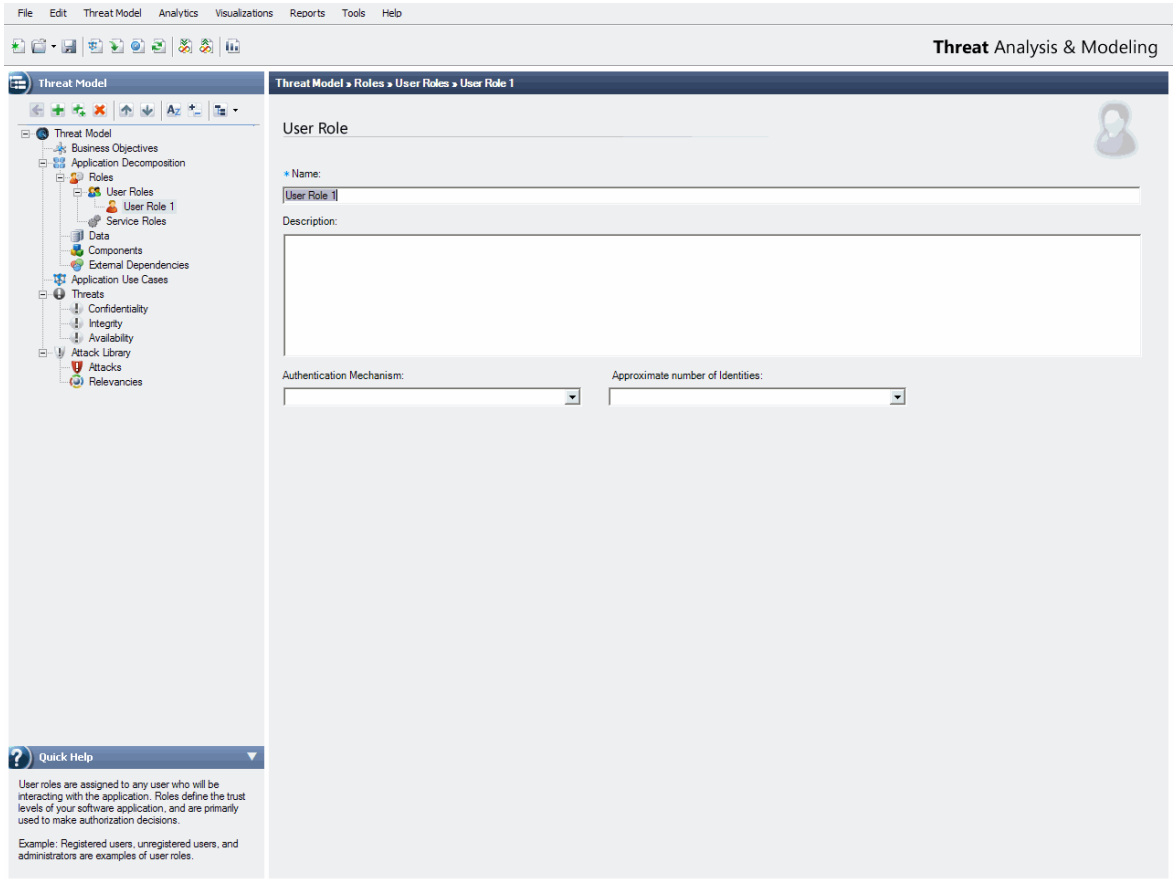


Figure 3. User Roles node and fields

To add a new user role:

1. Select the **User Roles** node, and then click the **Add Child Item** button.

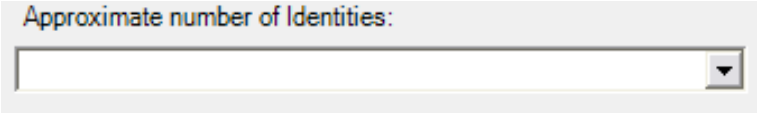
1. In the **Name** field, type the name of this user role.

Example: **Administrator** or **Registered User**.

1. (Optional) In the **Description** field, type a description of this user role.

1. (Optional) In the **Authentication Mechanism** drop-down list, select the authentication mechanism with which identities belonging to this role are authenticated.

1. (Optional) In the **Approximate number of identities** drop-down list, select the approximate number of identities that will be assigned to this user role.



Data

Data defines the information type that is maintained or processed by your software application. Optionally, data types can be broken down into specific elements that, together, make up that specific data.

> *Example:* User profile data consists of the following elements: salutation, user's first name, user's middle initial, user's last name, street address, and so on.

In order to identify the data types within a software application, data elements are grouped into logical sets, which can then be classified.

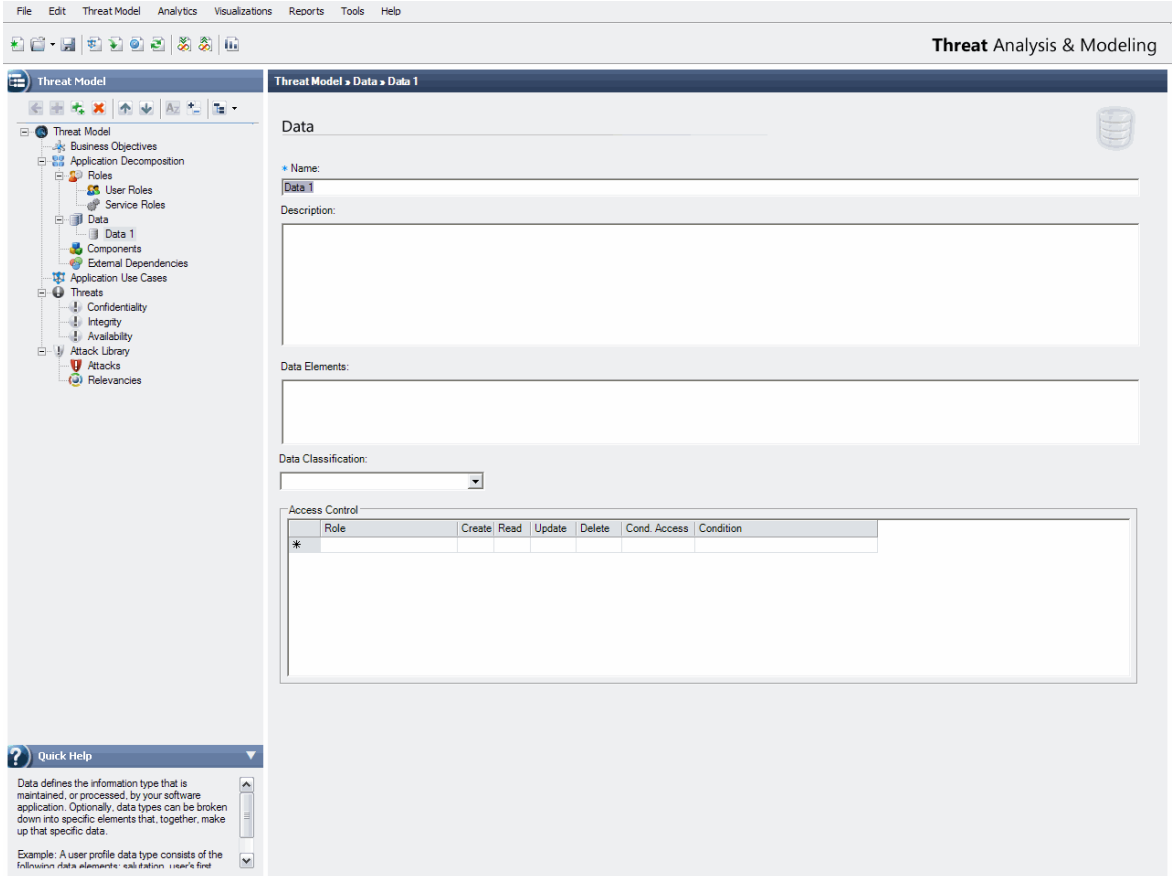
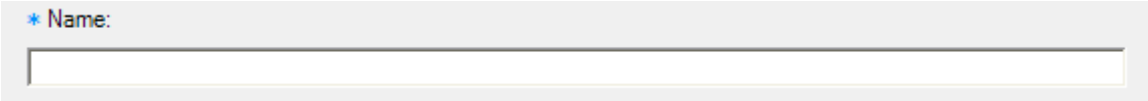


Figure 4. Data node and fields

To add a new data element:

1. Select the **Data** node, and then click **Add Child Item** button.

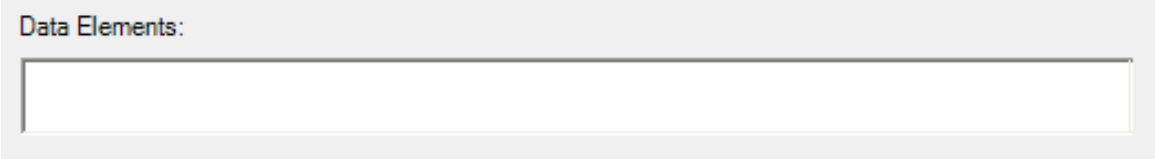
1. In the **Name** field, type a name for your data type.



Example: **Credit card Information** or **Customer Account Information**.

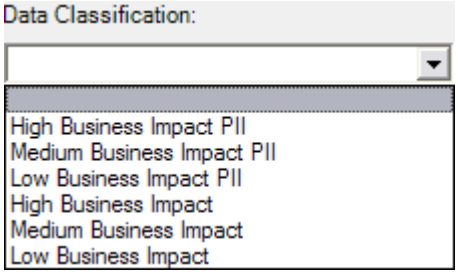
1. (Optional) In the **Description** field, type a description of your data.

1. (Optional) In the **Data Elements** field, enter the elements that make up your data.



Example: If your data is credit card information, elements would include customer name, credit card number, and expiration date.

1. (Optional) In the **Data Classification** drop-down list, select the appropriate data classification.



Note: The data classification list is extensible and can be pre-populated with any classification scheme.

1. In the **Access Control** field, you will set role permissions for each data type.

   Example: Administrators can create, read, and update credit card information.

Access Control

	Role	Create	Read	Update	Delete	Cond. Access	Condition
▶	Admins	☑	☑	☑	☐	☐	
	Registered Users	☑	☐	☐	☐	☐	
	Registered Users	☐	☑	☑	☐	☑	User is creator of Customer CC record
	Website Role	☑	☑	☑	☐	☐	
	Webservice Role	☑	☑	☑	☐	☐	
*							

Note: The available permissions include: Create, Read, Update and Delete. This group of permissions is commonly referred to by its acronym, CRUD. CRUD can be used to define any action.

Example 1: A copy operation is a combination of a Read and Create.

Example 2: A move operation is a combination of Read, Create, and Delete.

To set your access control:

a. Click in the blank cell below the **Role** heading, to enable the drop-down list.

b. In the drop-down list, select the role for which you want to set permissions.

c. Select the check box under each action that you want to allow the role to perform.

You can also set conditional permissions for specific roles.

To set conditional permissions:

d. Select the check box under **Cond. Access**.

e. In the **Condition** column, type in the conditional parameter.

   Example: A user might be able to create an instance of a credit card data type, however, to read, update or delete it requires conditional access. Thus, a user has the ability to read, update, and delete credit card information as long as it belongs them.

Application Use Cases

Once you have defined your roles, data and access control, you must define use cases that, at the very least, define the scenarios that are used to realize specific subsets of the defined access control.

A *Use Case* is an ordered sequence of actions that are used to realize an effect. This effect, known as the *Net Data Effect*, is a specific subset of the access control matrix that is realized at the end of the use case.

> *Example:* Reading product information, or updating credit card records.

When you run the data access control matrix, you will see all the access controls that are maintained by your application. These access controls are realized with use cases.

> *Example:* Suppose you have a use case called *Creating Customer Account*. If you define a user role with the access control to create credit card information, your use case must support this.

Use cases can be analyzed, to determine what elements are required in order to accomplish them. as well as how they interact with other objects.

> *Example:* A user role, website component, and database component are some of the elements required for the creating customer account use case.

In short, a use case defines what needs to happen.

Creating Use Cases

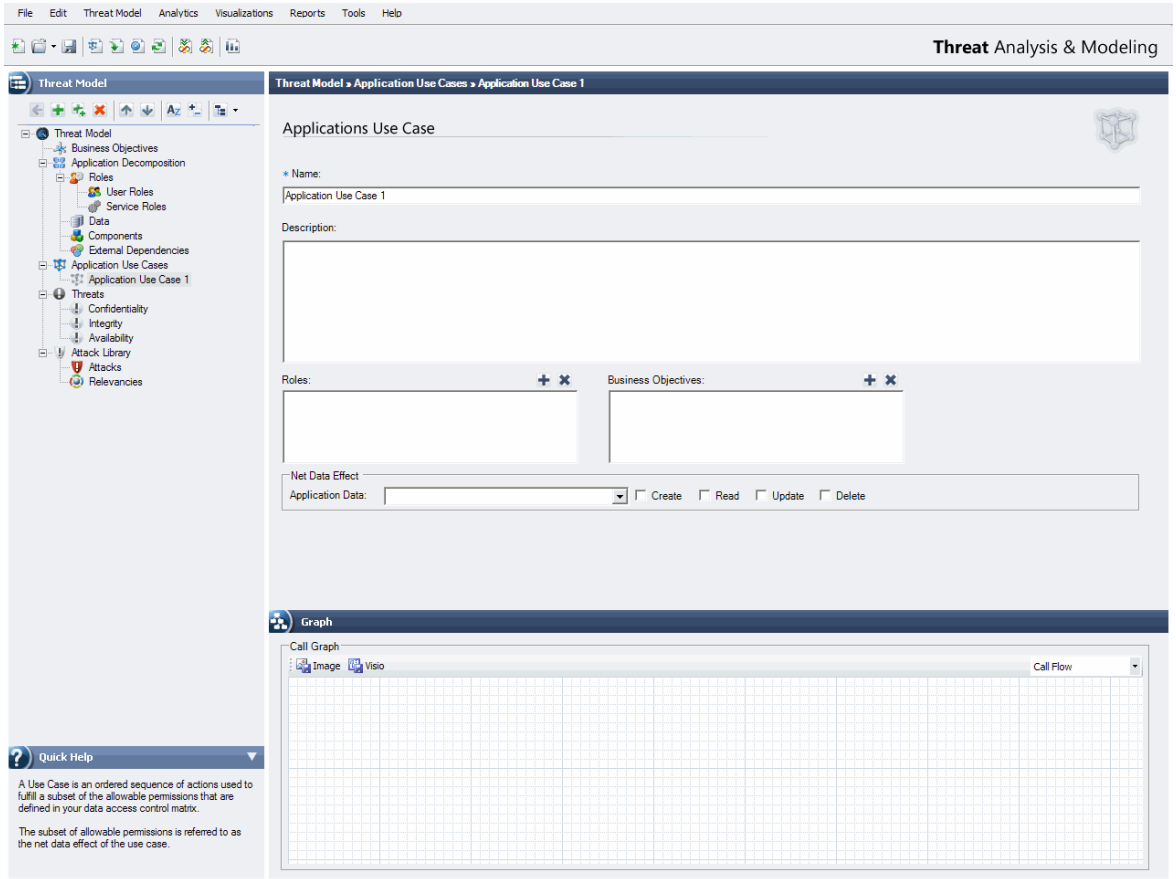
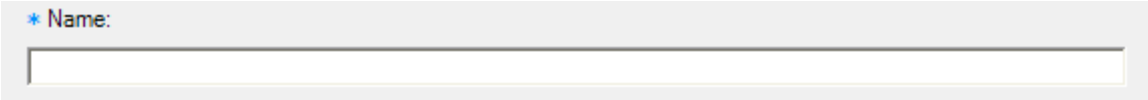


Figure 5. Application Use Cases node and fields

To create a new use case:

1. Select the **Application Use Cases** node, and then click the **Add Child Item** button.

1. In the **Name** field, type the name of your use case.



Example: **Browse Product Catalog** or **Add New Products To Catalog**.

1. (Optional) In the **Description** field, type a description of your use case.

1. In the **Business Objectives** field, click the **Add** button to select your business objectives that are being supported by this use case.

1. In the **Roles** field, click the **Add** button to select the roles for this use case. These are essentially the actors that will be interacting with your application through this defined use case.

1. The *net data effect* is the effect on the data that must be achieved through the execution of this use case.

   In the **Net Data Effect** field, select the check box for each action that is achieved at the end of this use case for *all* roles that you entered in the previous step.

   Example: The use case *Register New Users* achieves the effect of allowing certain roles the ability to create user accounts.

   Note: The *net data effect* is the subset of allowable permissions for each role, as defined in the Data Access Control Matrix.

1. In the **Application Data** drop-down list, select the application data that you are permitting the roles to access through the defined CRUD action specified in the previous step.

1. In the **Call Graph** field, the Threat Analysis & Modeling tool automatically generates a call flow graph, based on the information that you have entered. The graph will not appear until after your architects have defined the application architecture.

   A *call flow graph* is a visual presentation of how your elements (roles, data, and components) interact. The following figure shows an example of a call graph:

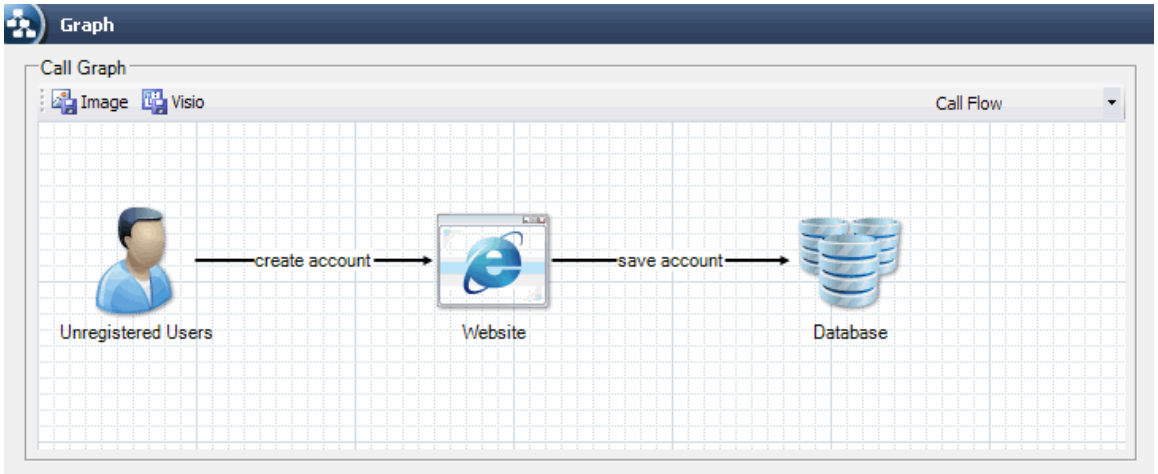


   Figure 6. Call Flow graph

1. You are now ready to move onto the next phase where you will define your application architecture.

Define Application Architecture

Your *application architecture* is comprised of your components, service roles, external dependencies and calls. You begin defining your application architecture by defining your components.

Components

Components are the building blocks of a software application that define an instance of a technology type such as databases and web services.

> *Example:* A database in which a user's credit card information is stored, web services, websites, and thick clients.

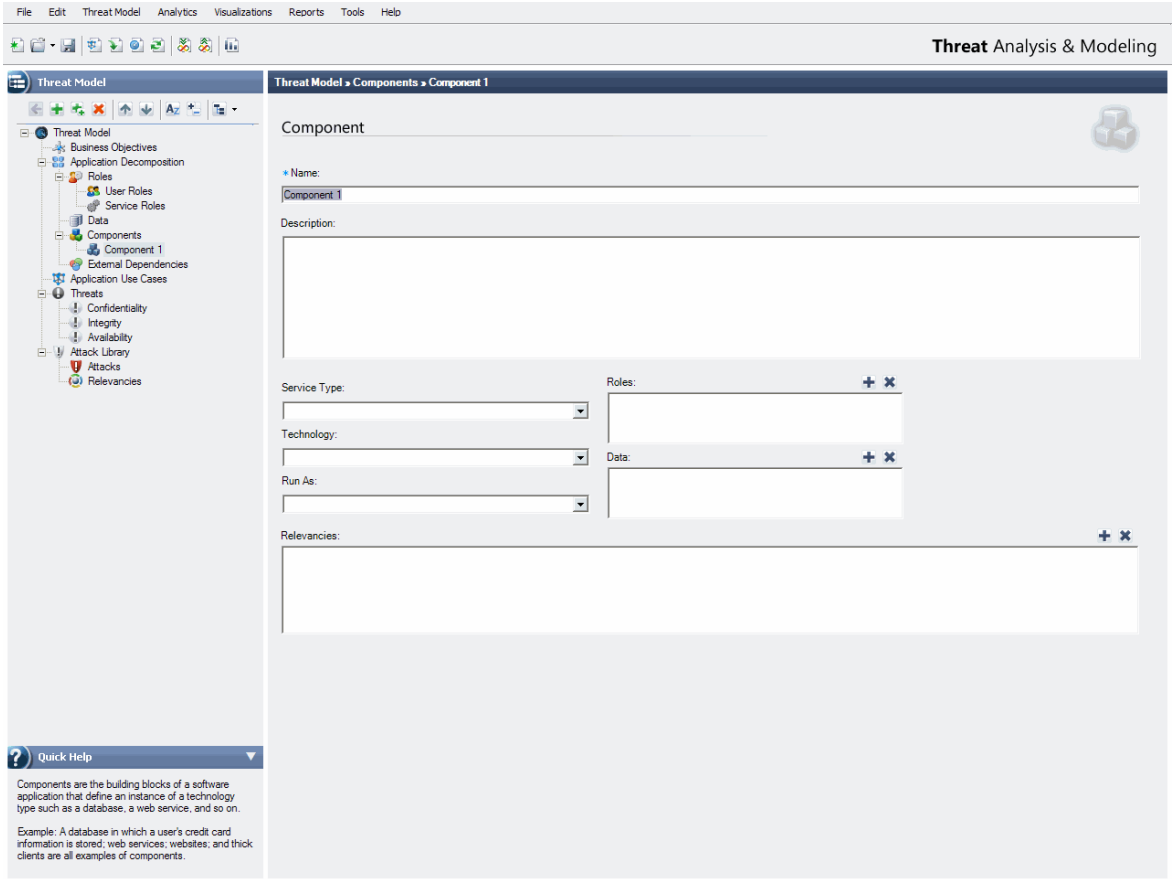
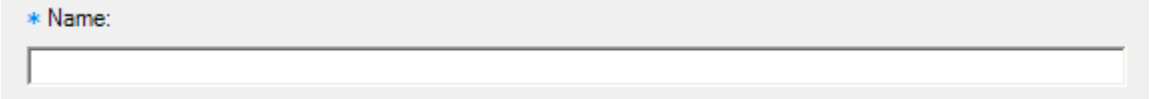


Figure 7. Components node and fields

To add a new component:

1. Select the **Components** node, and then click the **Add Child Item** button.

1. In the **Name** field, type the name of your component.

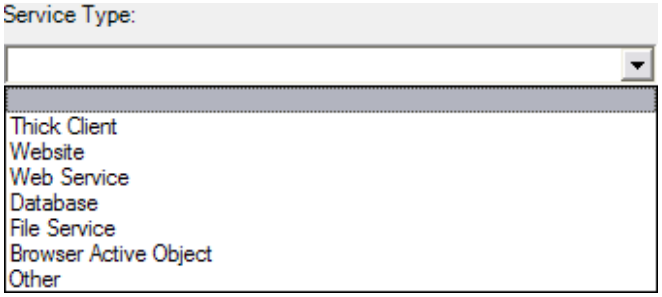


Example: **Website** or **Database**.

1. (Optional) In the **Description** field, type a description of this component.

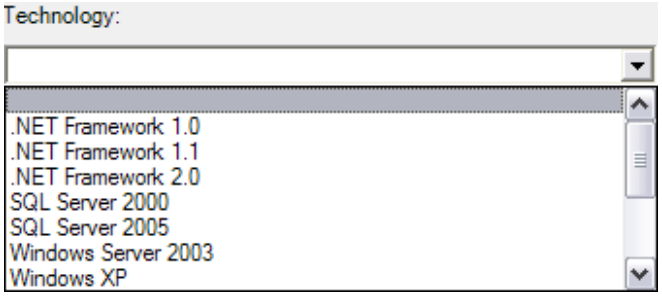
Note: Any description that you enter is for your own personal reference.

1. (Optional) In the **Service Type** drop-down list, select the type for this component.



Note: The service type list is extensible and can be pre-populated with any defined list of types.

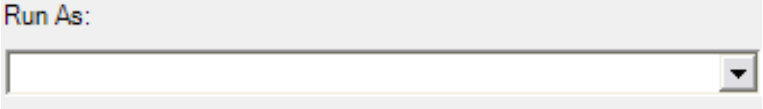
1. (Optional) In the **Technology** drop-down list, select the primary technology used to build this component.



Note: The technology list is extensible and can be pre-populated with any defined list of technologies.

1. In the **Roles** field, click the **Add** button to select the roles that will interact with this component. This defines the access control on the component where only the listed roles in this list box are permitted to interact with this component.

1. In the **Run As** drop-down list, select the service role identity that this component will impersonate by default.



1. In the **Data** field, click the **Add** button to select the defined data types that will be persisted in this component.

Note: A component that persists one or more data types is considered a *data store*.

1. In the **Relevancies** field, click the **Add** button to select the attributes that are relevant to this component.

Note: These relevancies are provided as part of a pre-defined attack library and help bind components to specific attacks.

To import an attack library:

a. On the **Tools** menu, select **Attack Library**.

b. Click on **Import**.

c. Select the attack library you want to import.

Service Roles

Service roles are trust levels that contain specific identities that define the context of various components running in your software application.

Example: Website roles and database roles for your website and database components, respectively.

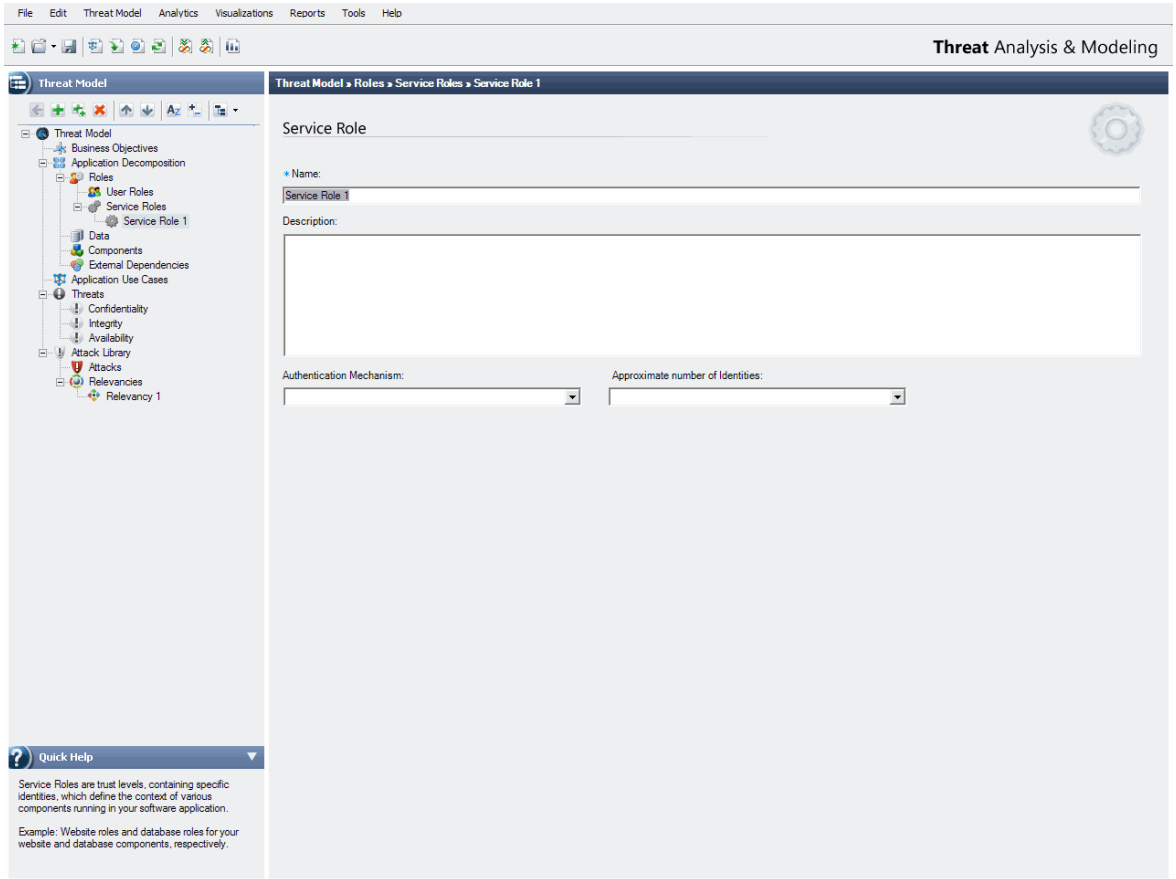
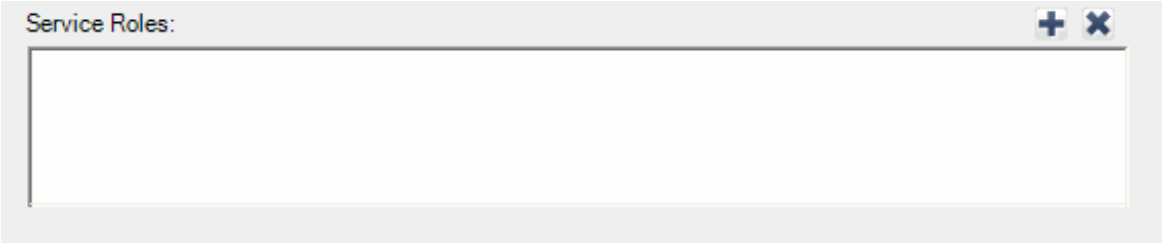


Figure 8. Service Roles node and fields

To add a new service role:

1. Select the **Service Roles** node, and then click **Add Child Item** button.



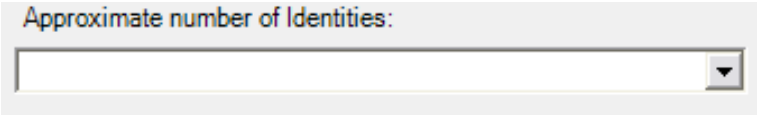
1. In the **Name** field, type the name of the service role.

Example: **Website Role** or **Database Role**.

1. (Optional) In the **Description** field, type a description of this service role.

1. (Optional) In the **Authentication Mechanism** drop-down list, select the authentication mechanism with which identities belonging to this role are authenticated.

1. (Optional) In the **Approximate number of identities** drop-down list, select the approximate number of identities that will be assigned to this service role.



External Dependencies

External dependencies are components with which your application will interact, and over which you have no control.

Example: .NET Passport is an external dependency of www.hotmail.com.

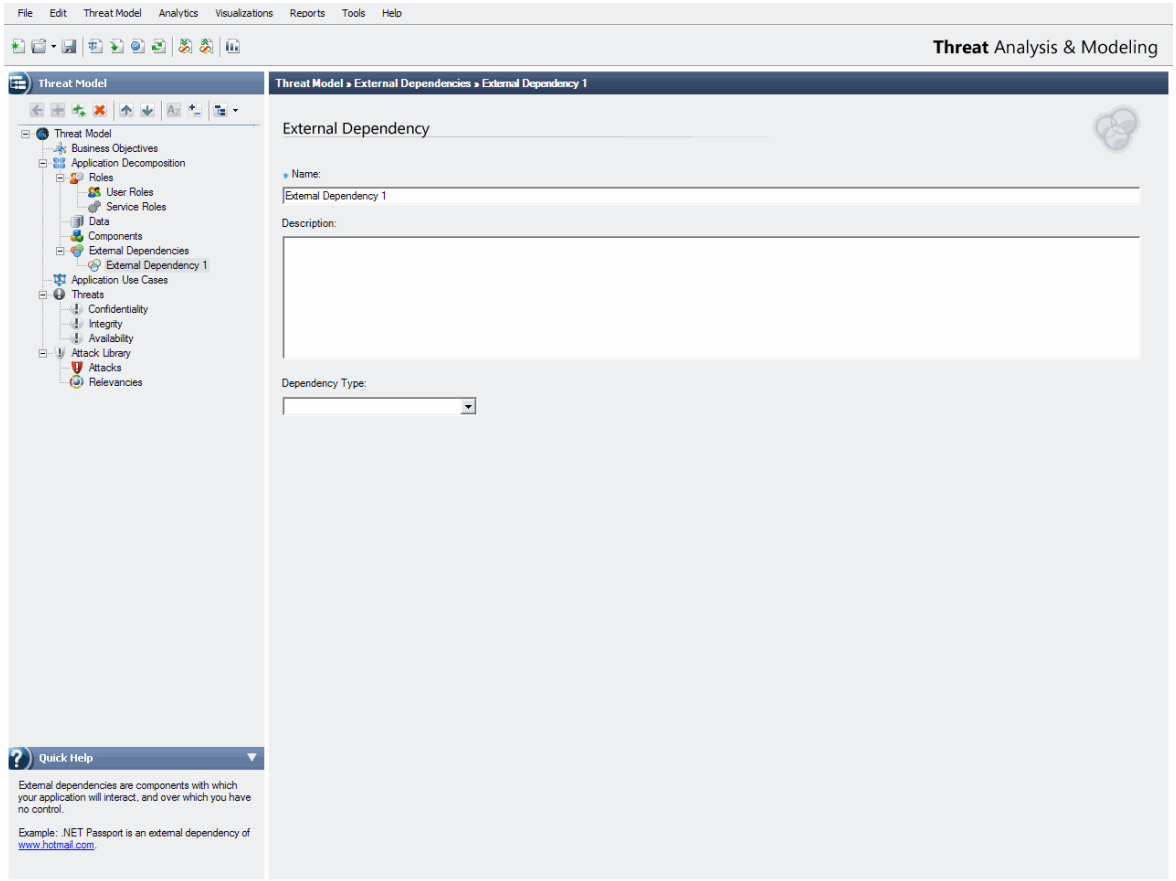


Figure 9. External Dependencies node and fields

To add a new external dependency:

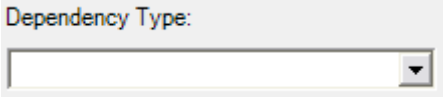
1. Select the **External Dependencies** node, and then click the **Add Child Item** button.

1. In the **Name** field, type the name of your external dependency.

   Example: **Microsoft Passport** or **External Datastore Feed**.

1. (Optional) In the **Description** field, type a description of this external dependency.

1. (Optional) In the **Dependency Type** drop-down list, select a type for this external dependency.



   Note: The dependency list is extensible and can be pre-populated with any defined list of technologies.

Calls

The *call* captures the connection between two pre-defined elements (user roles, service roles, components, external dependencies) in the form of a caller (the element that is invoking the action) acting on a component (the element effected) through a specified action. The data transfer between the coupled elements for the specified action is also captured in the call and the transfer could be from the caller to the component (data sent) or from the component back to the caller (data received). A collection of calls for a specified action allows you to define how that use case is realized in the context of your application.

Recall that use cases define the set of actions or features that need to be supported by your application, and these allotted actions can be executed by specific roles in order to achieve a net data effect. Put simply, use cases define what needs to happen, and *calls* define how it happens.

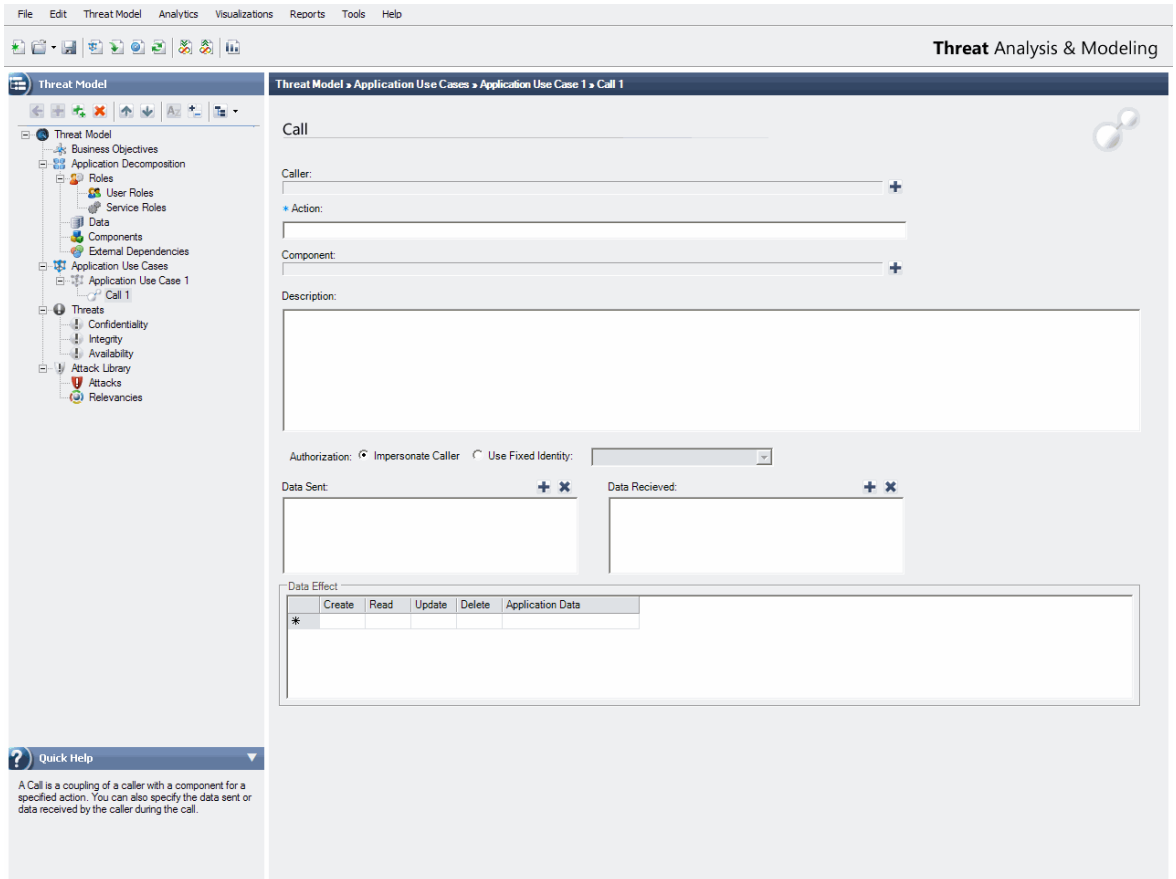


Figure 10. Call node and fields

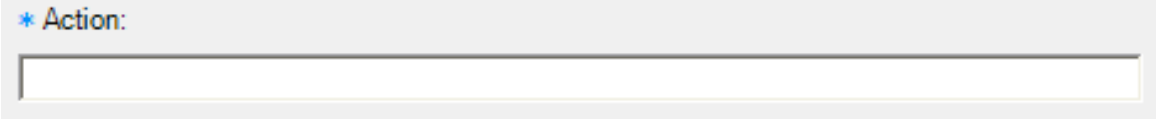
To create a new call:

1. Select a defined Use Case node under **Application Use Cases** group, and then click the **Add Child Item** button.

1. In the **Caller** field, click the **Add** button to select the role or component that is initiating this call.

Example: **Registered users**, **Admins**, or **Websites**.

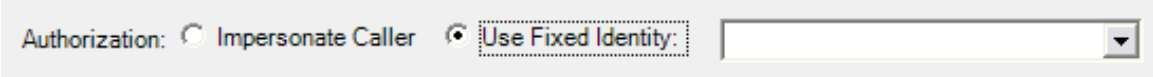
1. In the **Action** field, type in the allowed action for the specified caller.



1. In the **Component** field, click the **Add** button to select the component for this call.

1. (Optional) In the **Description** field, type a description of your call.

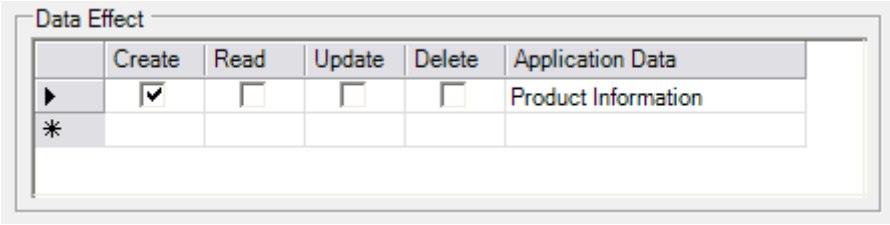
1. In the **Authorization** field, select either the **Impersonate Caller** or **Use Fixed Identity** radio button, depending on whether you want to delegate a caller or impersonate a service role.



1. In the **Data Sent** field, click the **Add** button to select the data that the caller will send to the component during this transaction.

1. In the **Data Received** field, click the **Add** button to select the data that the caller will receive from the component during this transaction.

1. If a component is defined as a data store—if it stores one more data types—select the appropriate boxes in the **Data Effect** field to indicate the effect achieved on the data involved in this transaction.



To select the data effect:

a. Click in the cell below the **Application Data** heading, to activate the drop-down list.

b. In the drop-down list, select the data type to be effected.

c. Check the box under each data effect (Create, Read, Update, or Delete) you would like to set.

Model

Threat modeling is meant to be an iterative process in which the threat model evolves through the many stages that information is consolidated from different members of your application team.

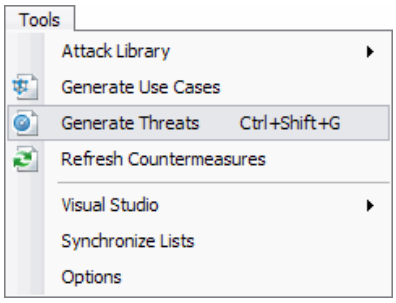
Generating Threats

Once you have defined your application context, the Threat Analysis & Modeling tool can be used to automatically generate threats. Threats are generated by systematically corrupting the allowable actions (defined calls) of your application. They are then classified into the following three threat categories: confidentiality threats, integrity threats, and availability threats.

> **Note:** Through the use of the attack library the Threat Analysis & Modeling tool will automatically define potential attacks that can be used to realize given threats, and then propose effective countermeasures.

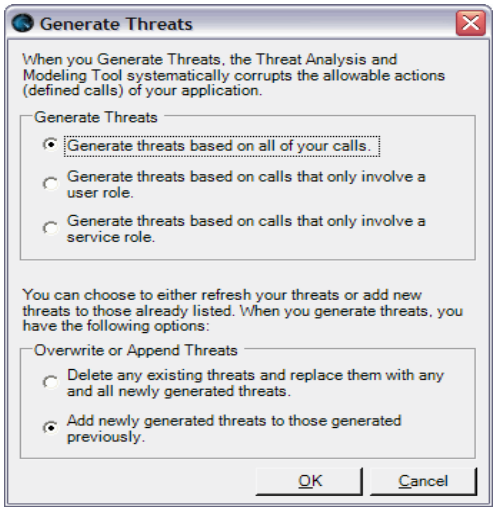
To generate threats:

1. On the **Tools** menu, click **Generate Threats**; or click the **Generate Threats** button on the toolbar.



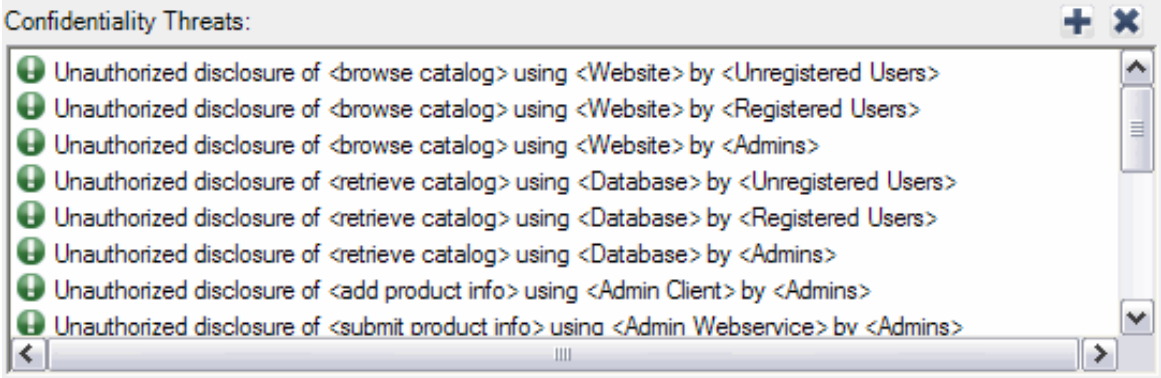
- or - 

1. In the **Generate Threats** dialog box that appears, select the threat types that you want to generate, and then click **OK**.

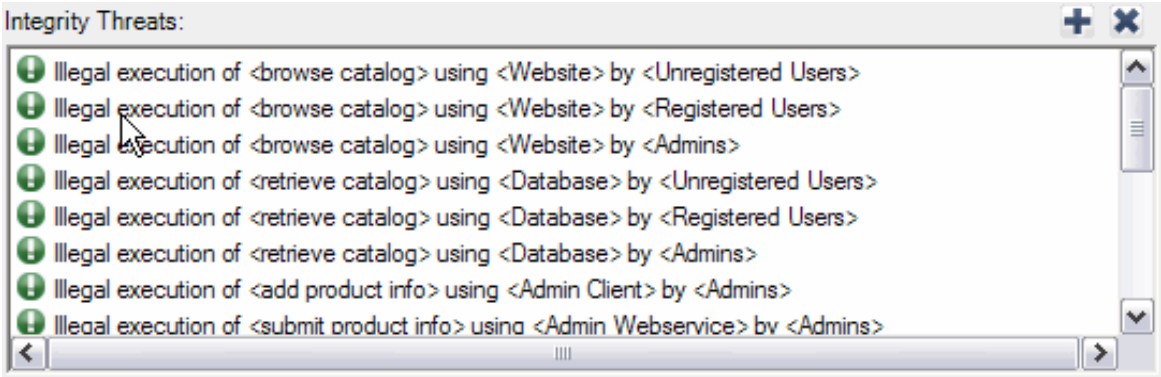


1. Your tree view will automatically repopulate itself with a list of threats, classified under the **Confidentiality**, **Integrity**, and **Availability** nodes. See the following examples:

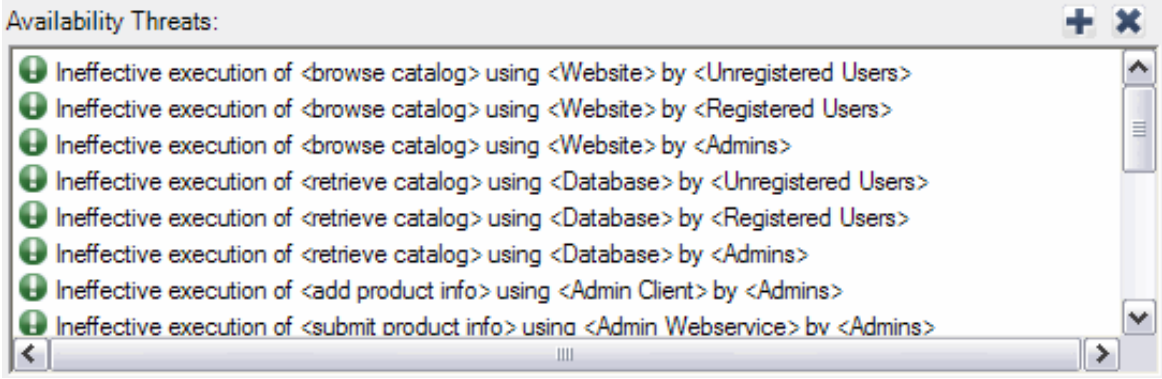
   Confidentiality threats:



   Integrity threats:



   Availability threats:



Primary Threat Factors

After your threats have been generated you are able to select the primary threat factors for each individual threat. These threat factors help provide a better context to aid in the process of analyzing your threats, which is essential to building your security strategy.

Confidentiality Threat

There are two primary threat factors that fall into the confidentiality category. Given an allowable action on a component by a role, its confidentiality can primarily be compromised through the unauthorized disclosure of the executing identity or the data.

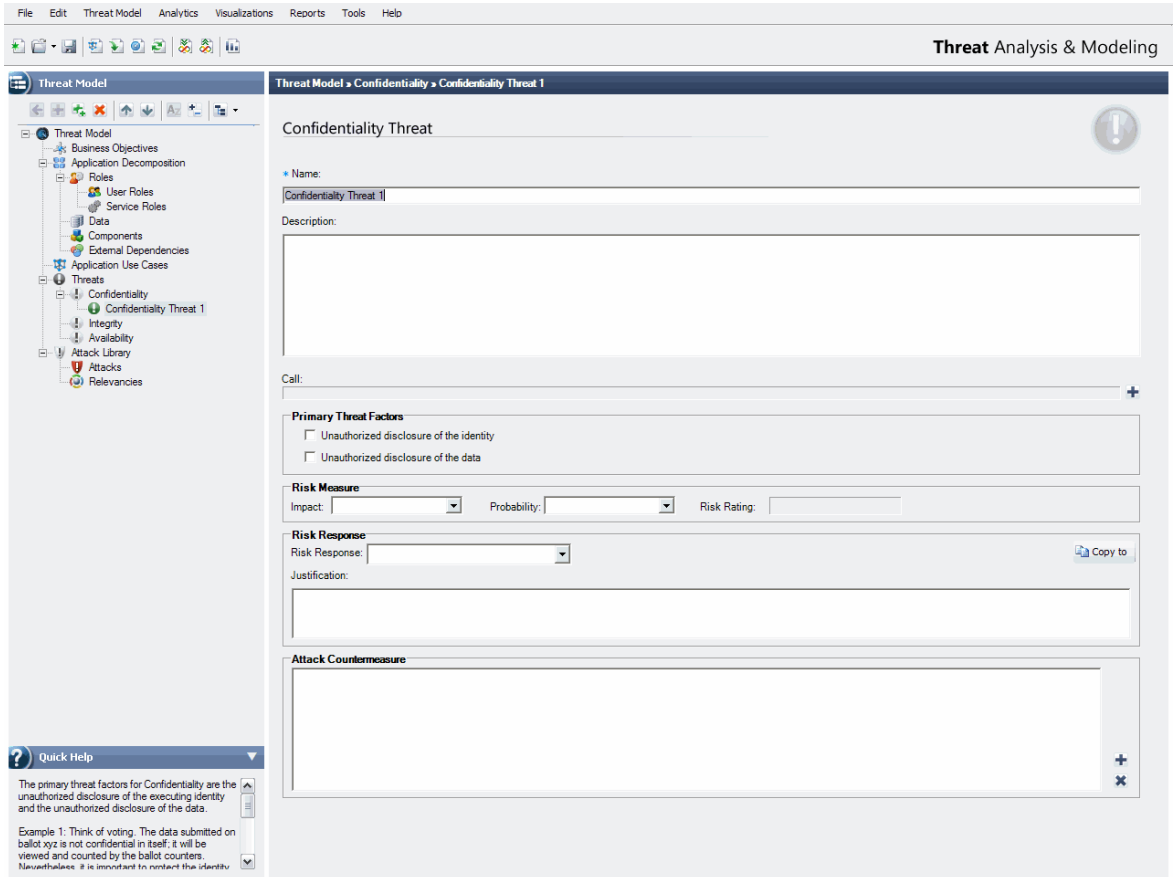
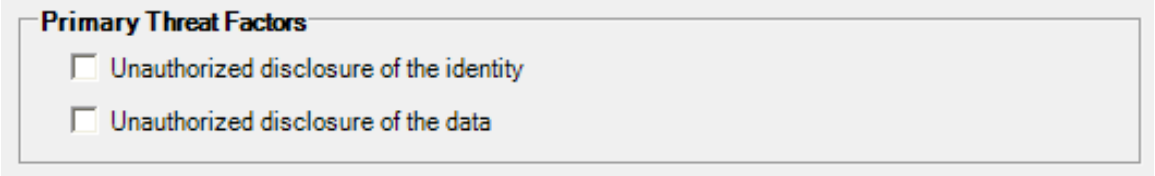


Figure 11. Confidentiality node and fields

To select your primary confidentiality threat factors:

1. Select one of the threats generated under the **Confidentiality** node.
2. In the **Primary Threat Factors** field, check the box next to the option that best defines the threat.



Unauthorized disclosure of the identity

Example: Consider voting. It may not be important to protect the result of the vote (the data), but it is certainly an issue if the vote (the data) along with the voters identity is disclosed to anyone (unauthorized disclosure).

Unauthorized disclosure of the data

Example: Consider a patient's visit to the doctor. Protecting the patient's identity from unauthorized disclosure in regards to this action (visiting a doctor) is not generally important. However, the information the patient discusses with the doctor (the data of the action) is almost certainly of sensitive nature and needs to be kept confidential.

Integrity Threat

There are three primary threat factors that fall into the integrity category. The primary factors are the violation of access control, business rule(s), or data integrity.

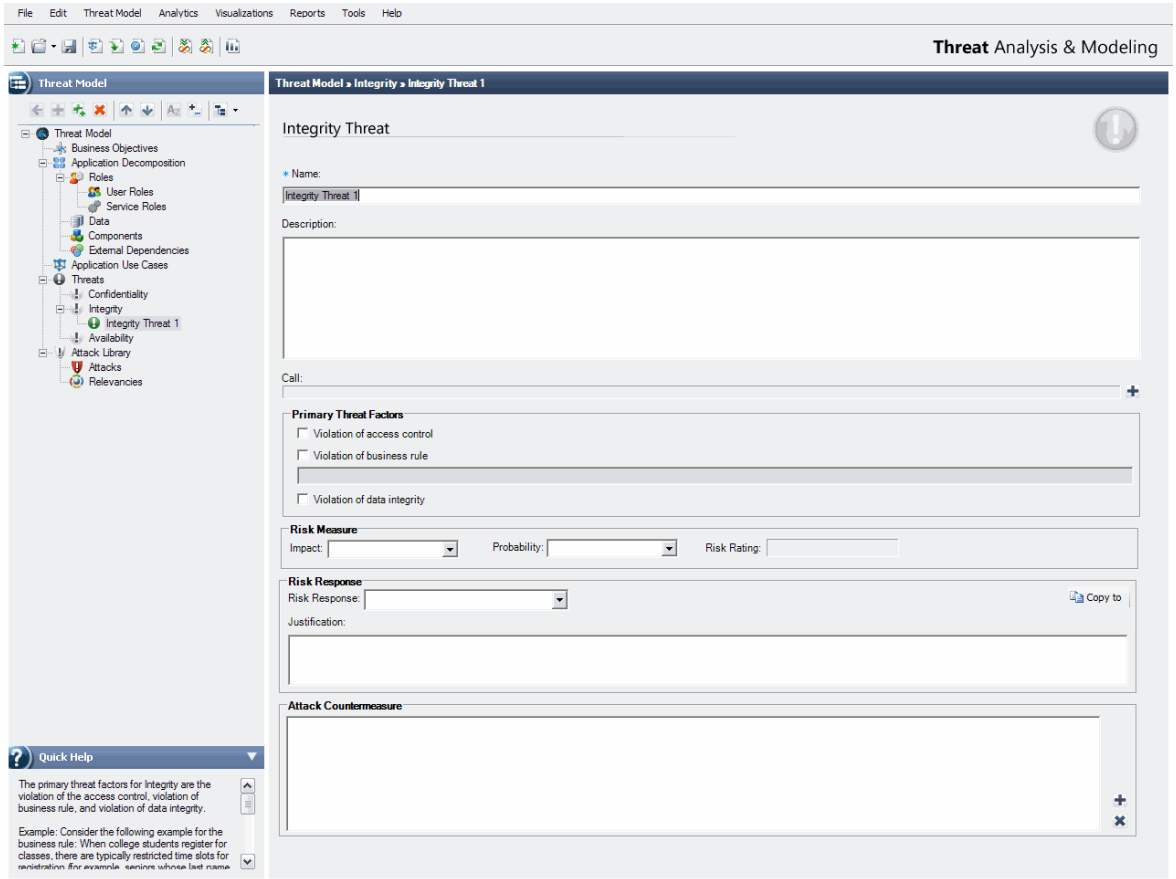
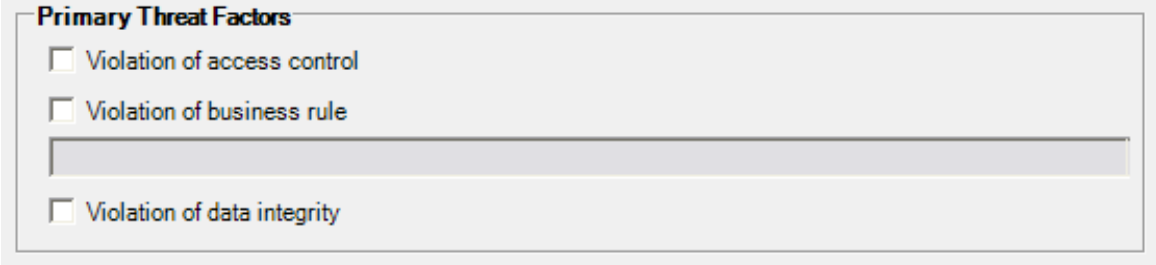


Figure 12. Integrity node and fields

To select your primary integrity threat factors:

1. Select one of the threats generated under the **Integrity** node.
2. In the **Primary Threat Factors** field, check the box next to the option that best defines the threat.



Violation of access control

Example: In your access control, you set permissions that allow administrators to create, edit, and delete product pricing information, while you set permissions that allow registered users to read product information. If your access control was violated and registered users were allowed to edit product pricing information, it would be possible for them to set all pricing to zero. This would be an enormous threat to your business.

Violation of business rule

The violation of business rule is something to consider that may be outside the realm of access control and data integrity.

Example: A banking transaction use case is made up of two calls, a withdrawal and a deposit. The business rule is that a user cannot make a withdraw without first making a deposit (i.e. there needs to be money in the users account in order for them to take any out). If a user was allowed to withdrawal money from a bank without having to make a deposit first, banks would surely be depleted quickly.

Note: If you check *Violation of business rule* as a primary threat factor, you must supply the business rule.

Violation of data integrity

Example 1: Consider a user's credit card information. It is crucial that the correct credit card number is paired with its rightful owner (user).

Example 2: Consider a case where an attack is able to compromise your application and modify the price of all your products that you sell to $1. This attack violates your data integrity by compromising your product information and pricing.

Availability Threat

There are two primary threat factors that fall into the availability category. The compromise could happen through the ineffective execution of the action or due to performance degradation.

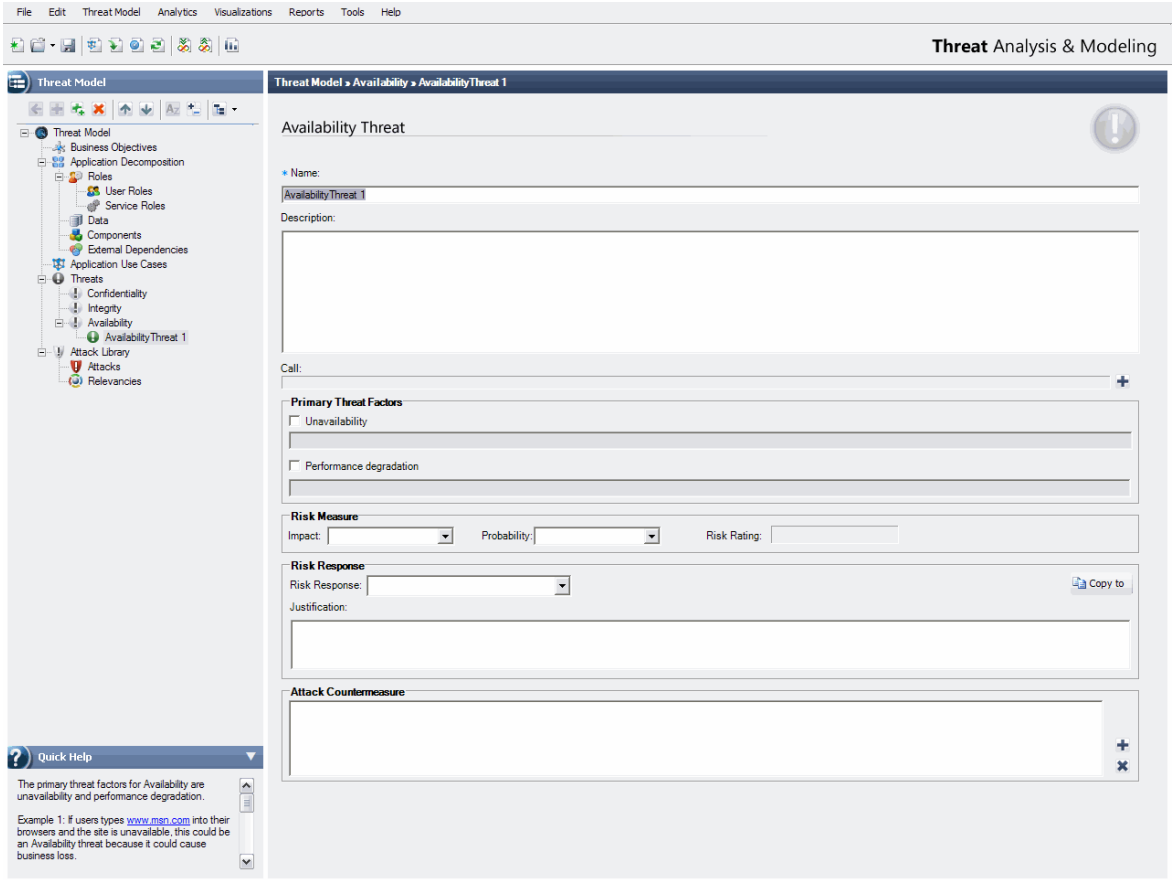
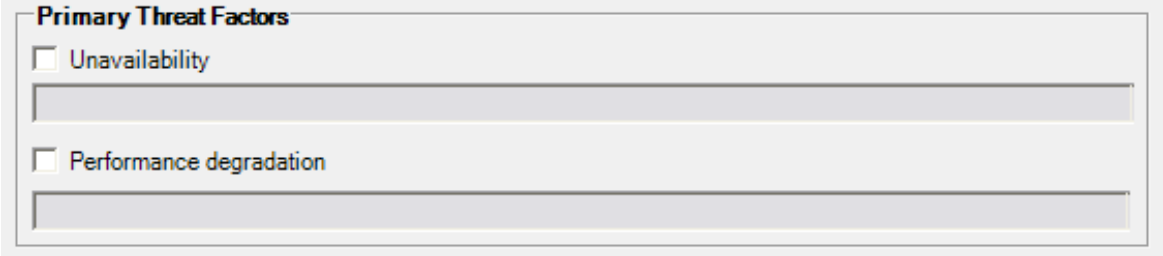


Figure 13. Availability node and fields

To select your primary availability threat factors:

1. Select one of the threats generated under the **Availability** node.
2. In the **Primary Threat Factors** field, check the box next to the option that best defines the threat.



Unavailability

Example: When a website is required to be available 99.9% of the time or a bank that needs to be open during their advertised business hours, the unavailability of either could cause significant business impacts.

Note: If you check *Unavailability* as a primary threat factor, you must supply a definition of the availability condition.

Performance degradation

Example: If an e-commerce site takes more than 5 seconds to respond to a request, this may be considered a threat, especially if this ineffective execution will result in a loss of potential business.

Note: If you check *Performance degradation* as a primary threat factor, you must supply a definition of the performance requirement.

After you have classified your threats, your next step is to define how you will respond to them.

Measure

The process of measuring risk is very subjective. Because of this subjectivity, once the Threat Analysis & Modeling tool has modeled your application context and generated threats, you are given the opportunity to measure the probability and impact of each threat in relation to your own business needs.

In the measure phase, you will attempt to quantify your threats. This is achieved through quantifying the risk associated with each threat.

> **Note:** Although risk measurement is subjective, as long as the user stays consistent in the measurement, the user is provided with a priority that can be used to appropriately allocate resources in an effort to guard against the realization of threats.

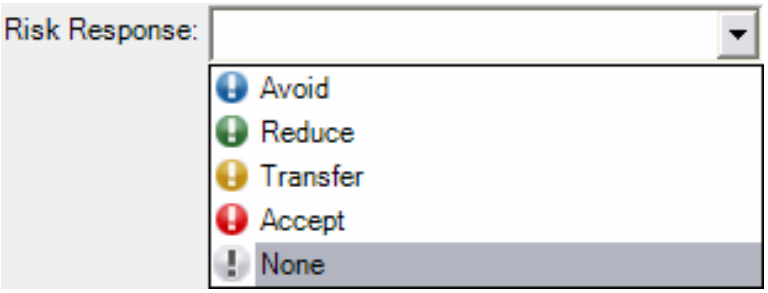
Risk Response

Along with measuring the risk associated with each threat, your threat model should also facilitate and document the responses to those identified risks. There are four responses one can have towards an identified risk. You can choose to accept, avoid, reduce, or transfer the risk.

- **Accept:** Choosing to accept a risk would be appropriate when the business supporting the software application takes full ownership of that risk, and all that it entails in terms of the negative business impact.
- **Avoid:** Risk is avoided when all supporting features for the underlying factors are removed. If, for example, the threat is concerned with the unavailability of some action, then that action needs to be removed.
- **Reduce:** Risk is reduced by applying countermeasures which lessen either the impact or the probability of the threat.
- **Transfer:** Risk is transferred when the underlying action is transferred to an external dependency. Risk is also transferred when the risk inherent in the action is illustrated to the user and the user accepts that risk in order to use that feature.

To respond to a risk:

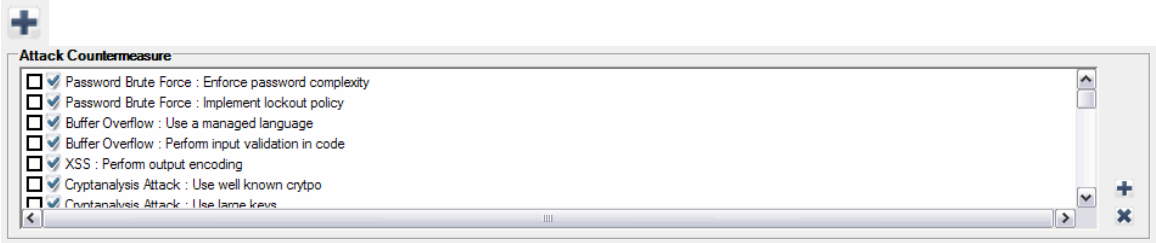
1. Select one of the threats that was generated.
2. In the **Risk Response** field, select your response from the drop-down list.



1. In the **Justification** field, type in your justification to your selected risk response.

Example: Your threat is the unauthorized disclosure of product information by unregistered users, and you choose to accept the risk. Your justification could be that the data exposed over this action is not sensitive data.

1. In the **Attack Countermeasure** field, click the **Add** button and select the appropriate countermeasures for this threat.



Risk Measure

To measure the risk associated with a threat:

1. Select one of the threats that was generated.
2. In the **Risk Measure** field, select the **Impact** and **Probability** of the threat from the drop-down list.

Note: The risk rating will automatically generate based on your impact and probability selections.

Attack Library

Overview

An *attack library* is a collection of attack types along with their relevant vulnerabilities and proposed countermeasures to those vulnerabilities. Attack libraries enable software application teams to define and adopt secure engineering techniques, gain the information necessary to detect security concerns, and create relevant security test cases.

Attack libraries provide a way to define, with absolutely minimal permission, the relationship between the exploit (*attack*), the cause (*vulnerability*), and the fix (*countermeasure*). The attack library helps ensure that various development teams understand the security assumptions and dependencies of your application.

> **Note:** Attack libraries are meant to be created by security subject-matter experts and consumed in the process of threat modeling.

Attacks

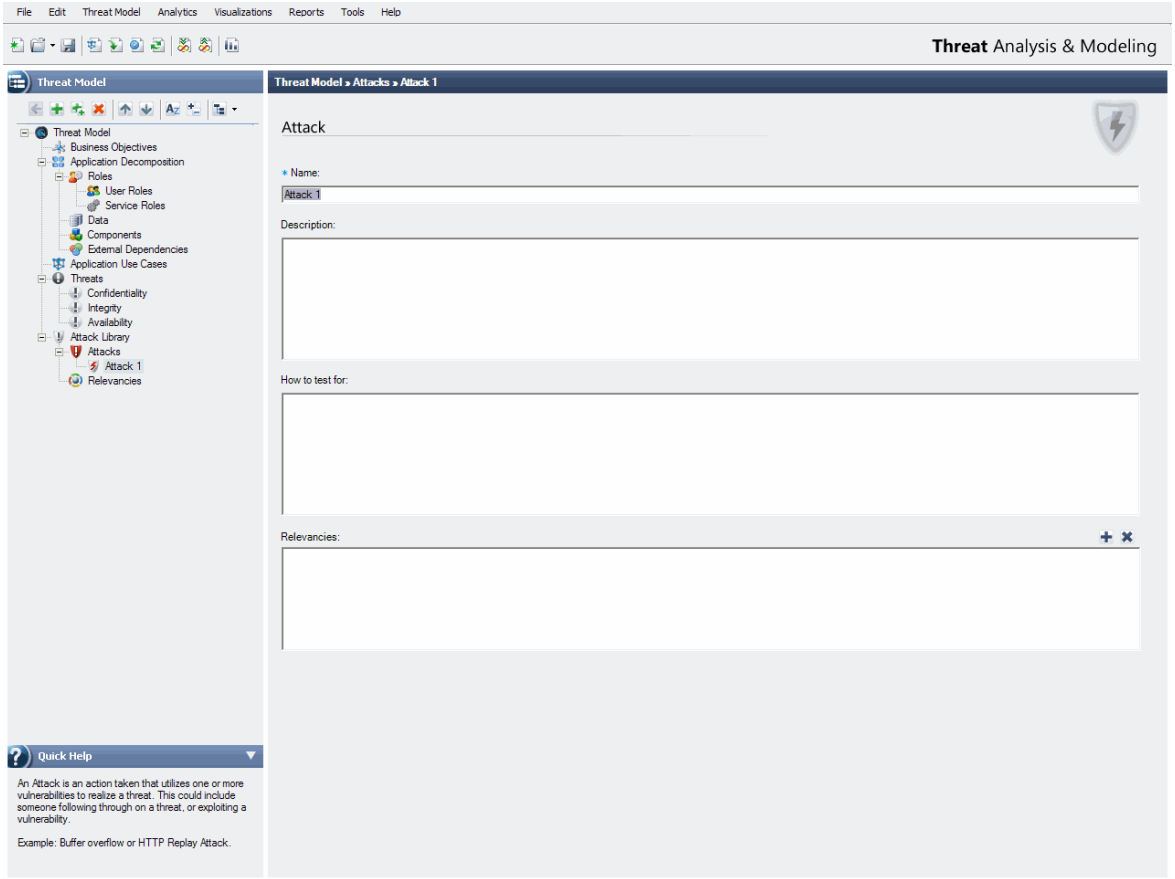
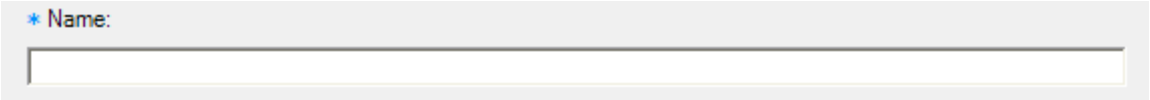


Figure 14. Attacks node and fields

To add a new attack:

1. Select the **Attacks** node, and then click the **Add Child Item** button.

1. In the **Name** field, type the name of the attack.



1. (Optional) In the **Description** field, type a description of this attack.

1. In the **How to test for** field, type an explanation of how to test for this attack.

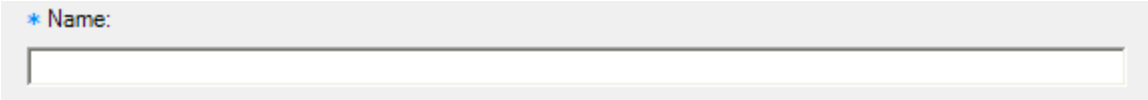
1. In the **Relevancies** field, click the **Add** button to select the relevancies for this attack.

Note: You cannot add relevancies until you define them in your attack library.

To add a new vulnerability:

1. Select the an attack, and then click the **Add Child Item** button.

1. In the **Name** field, type the name of the vulnerability.



1. (Optional) In the **Description** field, type a description of this vulnerability.

1. In the **How to recognize** field, type an explanation of how to recognize this vulnerability for this attack.

To add a new countermeasure:

1. Select a vulnerability, and then click the **Add Child Item** button.

1. In the **Name** field, type the name of the countermeasure.

1. Check the box if this is a core countermeasure.
2. (Optional) In the **Description** field, type a description of this countermeasure.

1. In the **How to implement** field, type an explanation of how to implement this countermeasure for this attack.

Relevancies

A *relevancy* is an attribute that defines a component's behavior.

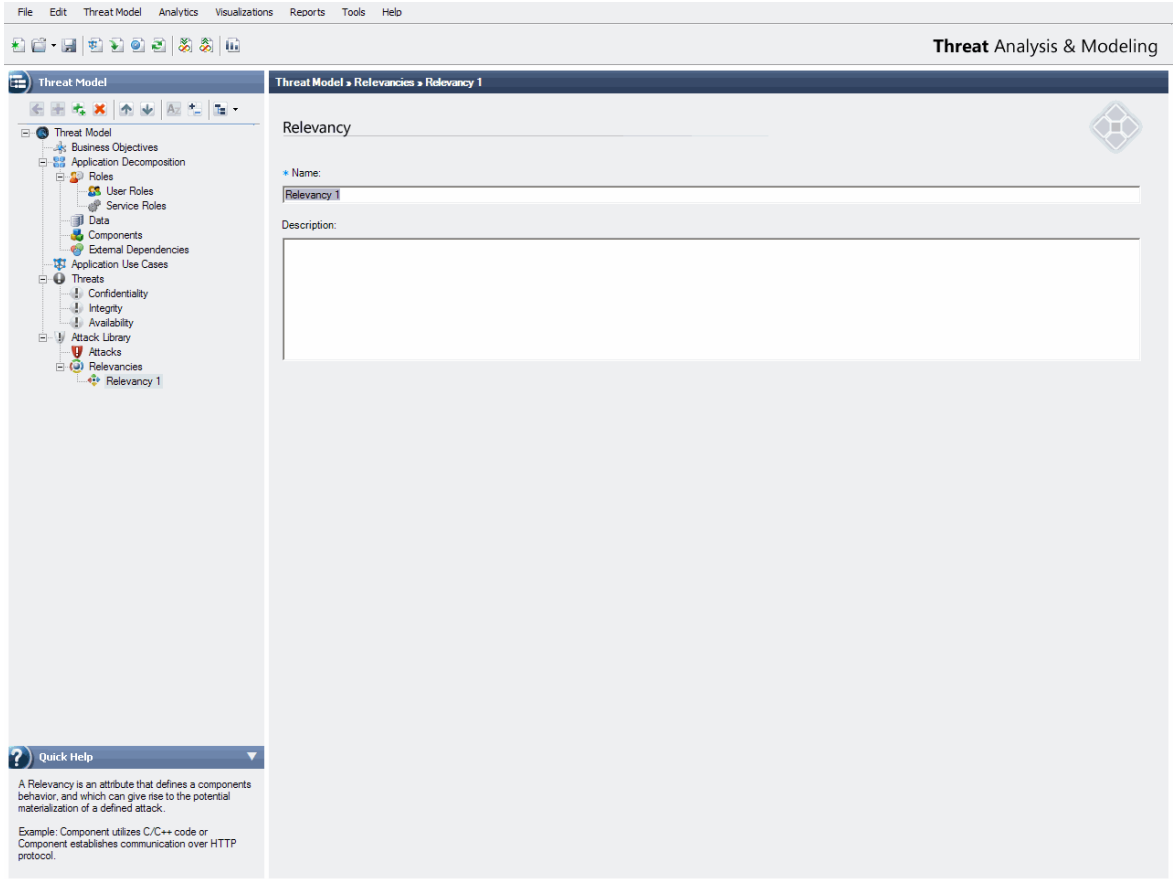


Figure 15. Relevancies node and fields

To add a new relevancy:

1. Select the **Relevancy** node, and then click the **Add Child Item** button.

1. In the **Name** field, type the name of the relevancy.

1. (Optional) In the **Description** field, type a description of this relevancy.