# Threat Modeling

Threat modeling and analysis provides a complete view about the security of a system. It is performed by a systematic and strategic way for identifying and enumerating threats to a system.

### 1. Some Common Definition (RFC 2828)

**Vulnerability:**
" A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy"

**Threat:**
"A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm... a threat is a possible danger that might exploit a vulnerability"

**Attack:**
"An assault on system security that derives from an intelligent threat, to evade security services and violate the security policy of a system."

### 2. Modeling Phases

We are currently following an iterative step, starting from the higher level – identifying major components and identification of threats from overall perspective. In the second part, we are performing threat analysis for each of the earlier identified components then passing data through those components from identified major use case. Finally, we merge the analysis report to one.

Questions: component breakdown or use case breakdown is better option ?

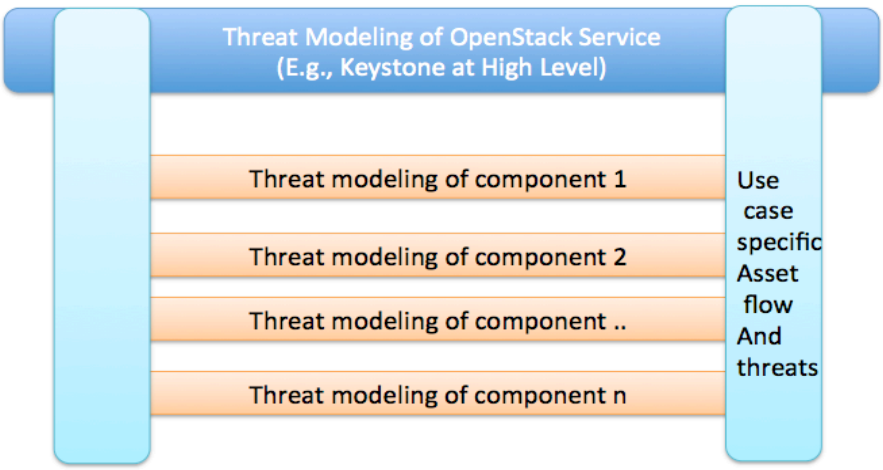
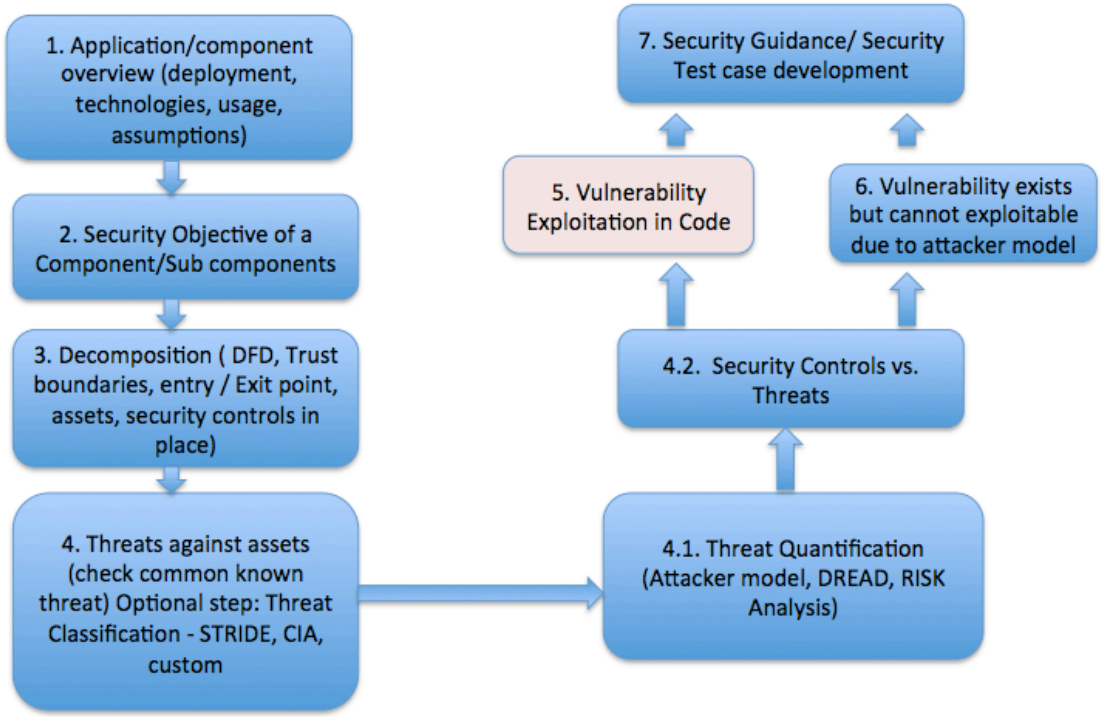



Figure 1: An iterative process for threat identification

Currently, we follow a simplified steps for modeling(Do not try to reinvent the wheel here,, Modified the OWSAP steps)




Figures 2: Simplified steps for threat modeling

**Steps:**

1. Application/component overview (deployment, technologies, usage, assumptions)

2. Security Objective of a Component/Sub components

3. Decomposition (Data Flow Diagrams - DFD, Trust boundaries, entry/exit point, assets, security controls in place)

4. Threats against assets (check known threats here.). STRIDE can be used for threat identification and classification.

4.1. Threat Quantification (Attacker model, DREAD, RISK Analysis)

4.2 Security Controls vs. Threats.

5. Vulnerability Exploitation in Code

6. Vulnerability exists but cannot be exploitable due to attacker model

7. Security Guidance/ Security Test case developments

## 3. Related Assumptions for the Threat Modeling:

### Threat Agents:

We consider three types of threat agent (Attackers).

| ID | Name | Details |
|---|---|---|
| IA-U | Internet Attacker– Unauthorized | |
| IA-A | Internet Attacker – Authorized | |
| IN-I | Internal Attacker - Insider | |

### Threat Categorization

We can follow STRIDE model

| **ID** | **Details** |
|---|---|
| SPOOFING | |
| TAMPERING | |
| REPUDIATION | |
| INFORMATION DISCLOSURE | |
| DEINIAL OF SERVICE | |
| ELEVATION OF PRIVILEDGE | |

### Threat Agent Capabilities

| Level | Name |
|---|---|
| 0 | Script Kiddies |
| 1 | Motivated individuals |
| 2 | Highly Capable Individuals |
| 3 | Serious Organized crime |
| 4 | Intelligence services |

### Likelihood of a Threat

| Level | Definition |
|---|---|
| Low | |
| Medium | |
| High | |

**Actors and Trust Level**

| Trust Level | Actors | Details |
|---|---|---|
| 1 | Anonymous | |
| 2 | Tenant User | |
| 3 | Tenant admin | |
| 4 | Keystone Identity Admin | Can control any operation on keystone through exposed API. |
| 5 | External Identity admin | Used in case authentication is consumed from external sources |
| 6 | Cloud Service | Cloud service account to verify user auth token and role info |
| 7 | System admin | Access to system process and databases |
| 8 | DB user | User access the database |
| 9 | User | Tenant user, tenant admin, Keystone identity admin user in authenticated form (any or combination of them in authenticated form) |
| 10 | Dashboard Admin | |
| 11 | Keystone process user | |

**Threat DB**

**Link to Threat DB ( A generic database to search for possible threats)**

## 4. Identification of Threats

In final analysis, we can follow a simplified process as exemplified in figure 3. In this picture, top row elements are generated from the step 1 to 3 from figure 2. These are dynamic parameters. The bottom row is the result. The middle rows are static elements – defined in section 3.

Based on this information, we identify threats for each asset accessed or modified by the proxies (operations) in the data flows diagram.

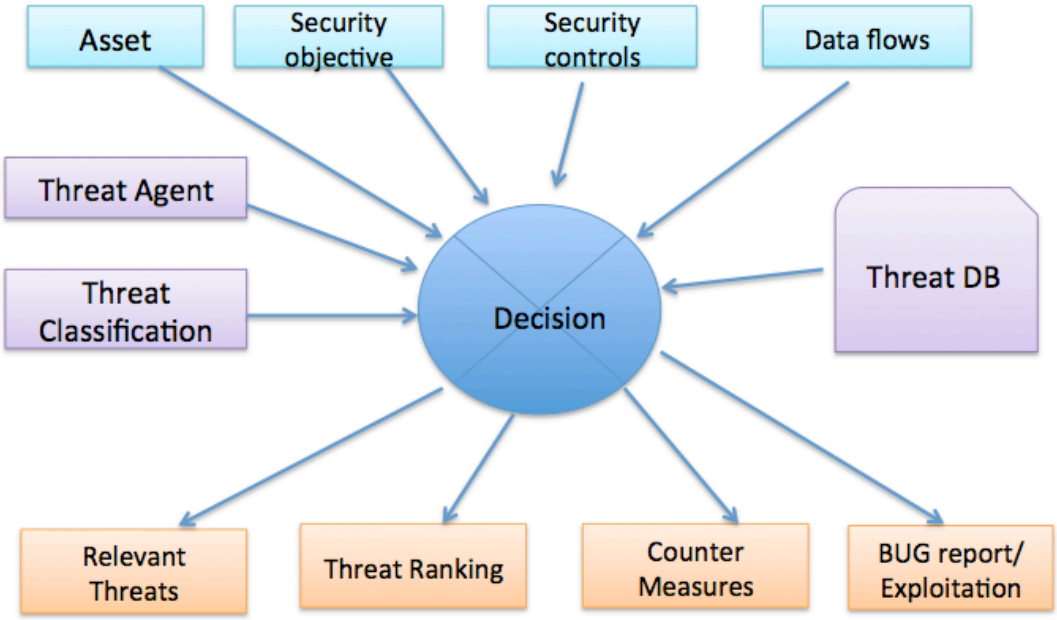



Figure 3: Threat Analysis - simplified

In other words,

```
# Logic to find related threats

for each asset_A in Asset:
   for each operations in DataFlows:
       modified_Asset_A = operation(asset_A)
       for each threat_group in STRIDE:
            attack_vectors = Find attack_vectors related to
            threat_group from Threat_DB

            possible_Threat  = Threats exploitable  by using
            attack_vectors on asset_A to perform an operation

           ranked_Threat = Ranking(possible_Threat)
           return ranked_Threat
```

**5. Example Application**

**Link to Keystone Threat Modeling**

**Link to Keystone Token Provider Threat Modeling**