Threat Modeling
**Vulnerability**: a software defect with security consequences

**Threat**: a potential danger to the software

**Attack**: an attempt to damage or gain access to the system

**Exploit**: a successful attack

**Trust Boundary**: where the level of trust changes for data or code
• Threats represent a potential danger to the security of one or more assets or components
  – Threats could be malicious, accidental, due to a natural event, an insider, an outsider, …
  – A single software choice can result in many threats.
  – Threats exist even if there are no vulnerabilities
    • No relaxing
    • Threats change with system changes

How can a change in software result in either or fewer threats?
• Social threats: people are the primary attack vector
• Operational threats: failures of policy and procedure
• Technological threats: technical issues with the system
• Environmental threats: from natural or physical facility factors
• The threats themselves are the same, but this is a different view
  – Threats have certain sources (Social, Operational, Technical, Environments)
  – And certain security impacts (STRIDE)
Threat Modeling Overview

• Threat Modeling is a process that helps the architecture team:
  – Accurately determine the attack surface for the application
  – Assign risk to the various threats
  – Drive the vulnerability mitigation process

• It is widely considered to be the one best method of improving the security of software

• The Microsoft approach is cumbersome
The phases of the Threat Modeling process

- Understand the security requirements
  - Use Scenarios – what are the boundaries of the security problem
  - Identify external dependencies – OS, web server, network, …
  - Define security assumptions – what can you expect with regard to security; will the DB encrypt columns? Is there a key manager? What are the limitations you are working with.

- Create an activity matrix (actor-asset-action matrix)
  - Identify assets
  - Identify roles
  - Their interaction

- Create Trust Boundaries
Threat Modeling
Overview

– Identify threats that put assets at risk
– Identify attacks that can be used to realize each threat
  • Threat Trees
  • Abuse Cases
– Determine the risk for each attack and prioritize (if needed)
– Define the conditions required for each attack to be successful

• Plan and implement your mitigations
This is abstracted from the OWASP site so that you can look at it in greater detail

- [https://www.owasp.org/index.php/Application_Threat_Modeling](https://www.owasp.org/index.php/Application_Threat_Modeling)

Moo U University is installing a new website to provide online access to patrons (students, staff) and library personnel

This starts with you determining the requirements

- What needs to be secured and what are the security requirements
- What are potential threats against the system
- What are the limitations on building the system
- ...
1. Name: Library Online Access Site Threat Model
2. Version: 1.0
3. Document Owner: Joe Security
4. Description: <as above>
5. Participants: Joe S, Bob W, Amy C (DEV), Ron R(LIB), Abby T(IT)
6. Reviewers: CISO, CSO, DM, SECTEAM
7. External Dependencies

- Server type will be Linux
- Site will have to be off-campus accessible
- MySQL database
- Database server will be the existing library server
- Private network between web server and db server
- Both servers must be behind the campus firewall
- All communications over TLS
8. Use Scenarios

- Students can search the database(s)
- Students can put holds on some items for checkout
- Staff can search the database(s)
- Staff can place some items on reserve for up to 15 weeks
- Librarians can do anything students or staff can do
- Librarians can place items on an invisible list
- Librarians can access limited account information
9. Roles (deviation from OWASP)

- Anonymous user – connected, but not yet authenticated
- Invalid user – attempted to authenticate and failed
- Student – authenticated student
- Staff – authenticated staff
- Librarian – authenticated librarian
- Site admin – authenticated site administrator with configuration privileges
- DB admin – authenticated database administrator with full db privileges
- Web server user – user/process id of web server
- Database read user – db user for accessing the database with read-only access
- Database write user – db user for accessing the database with read-write access
10. Assets

- Library users and librarian
- User credentials
- Librarian credentials
- User personal information
- Web site system
- DB system
- Availability of the web server
- Availability of the DB server
- User code execution on web site
- User DB read access
- Librarian/Admin code execution on the web site
- Librarian/Admin DB read/write access
- Ability to create users
- Ability to audit system events
### 11. Activity Matrix

- This can be messy and it best done in a spreadsheet. The results are much the same as in the OWASP example, but easier to visualize.

<table>
<thead>
<tr>
<th>Asset/Role</th>
<th>Anonymous</th>
<th>Invalid</th>
<th>Student</th>
<th>Faculty</th>
<th>Librarian</th>
<th>Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C R U D</td>
<td>C R U D</td>
<td>C R U D</td>
<td>C R U D</td>
<td>C R U D</td>
<td>C R U D</td>
</tr>
<tr>
<td>Users</td>
<td>A - - -</td>
<td>A - - -</td>
<td>- - -</td>
<td>- - -</td>
<td>X X X</td>
<td>-</td>
</tr>
<tr>
<td>Librarians</td>
<td>- - -</td>
<td>- - -</td>
<td>- - -</td>
<td>- - -</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Personal info</td>
<td>- - -</td>
<td>- - -</td>
<td>B B B</td>
<td>B B -</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Web site</td>
<td>- - -</td>
<td>- - -</td>
<td>- - -</td>
<td>- - -</td>
<td>C -</td>
<td>X X X X</td>
</tr>
</tbody>
</table>

- **A** = Create if valid name, id, pin provided
- **B** = Only for their own profile information
- **C** = Must be limited to specific files, tables. No access to web site files.
12. Trust Boundaries
Threat Modeling Example

- Login DFD
13. Threats

- Anonymous user evades the authentication system
- Anonymous user gathers information from the authentication system
- Anonymous user can forcefully browse to pages
- Librarian has access to web site pages on the server
- Student or Staff can modify privilege level
- Student or Staff can forcefully browse to restricted pages
- Any user can tamper with critical data on the client
- Student/Staff/Anonymous can inject SQL into the database
- Student/Staff/Anonymous can inject JavaScript into an HTML page
- SSL version is vulnerable or allows vulnerable algorithms
- …..
13. Threats – continued

- OWASP does this differently
  - First they talk about STRIDE, but they don't follow through with a list of threats
  - It is fine to use STRIDE and think about every place where Spoofing, Tampering, …. can be used
  - You need a very complete list, but you can combine threats that are common

- Understand the threats
  - There are tools to help: Threat Trees and Abuse Cases
Threat Modeling Example

• Threat Tree

- Attacker may be able to read other users’ messages
  - User may not have logged off on a shared computer
  - Data validation may fail, allowing SQL injection
  - Authorization may fail, allowing unauthorized access
  - Browser cache may contain contents of message
    - Implement data validation
    - Implement authorization checks
    - Implement anticaching HTTP headers
    - If risk is high, use SSL
• Abuse Case
Threat Modeling
Example

• Threat Tree

Student or staff can modify privilege level

By tampering with a URL
• Abuse Case: Student or staff can elevate privilege level
15. Plan your mitigations

- OWASP uses the following categories
  - Authentication
    - All credentialed users require user name and password required for authentication
    - All pages check authentication
    - All credentials communicated only with secure channel
    - No backdoor accounts or default accounts can be left available
  - Authorization
    - Use role-based authentication with unlimited levels, but including anonymous, user, staff, librarian, admin
    - All accesses will use least privilege and fail securely
  - Cookie Management
  - Data/Input Validation
  - Error Handling
  - Logging/Auditing
  - Cryptography
  - Secure Code Environment
  - Session Management
Threat Modeling Example

- Threat Modeling is over; continue with the remainder of the Design process