THREAT ANALYSIS – AN INTRODUCTION
THREAT ANALYSIS OVERVIEW

Threat Agent/Attacker

Threat exploits vulnerabilities

Successful attack has Threat consequences

Target System

Counter Measure

Vulnerability
RECAP – 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.”
THREAT ANALYSIS PROCESS

Simple Approach

Threat Modeling – Design centric

Exploiting the vulnerabilities related to Threats using an attacker model
MODELING STEPS

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

2. Security Objective of a Component/Sub components

3. Decomposition (DFD, Trust boundaries, entry/Exit point, assets, security controls in place)

4. Threats against assets, elements from the Decomposed system (check common known threat)
   Optional step: Threat Classification - STRIDE, CIA

5. Related Vulnerabilities

6. Threat Quantification (Attacker model, DREAD)

7. Existing Counter measures Vs. Threats

8. Vulnerability exists but cannot exploitable due to attacker model

9. Vulnerability Exploitation in Code

10. Security Guidance/Security Test case development
A SIMPLE EXAMPLE
ELECTRONIC LOCK IN HOTEL

A Simple Lock you see in Hotel Room
1. APPLICATION OVERVIEW

• Locks are used to control access to Hotel room.
• A guest card is used to open the lock.
• Guest card are programmable by some master.
• Each entry is logged in the door lock system.
1. SUB COMPONENTS

- **Central Encoder:** For making guest keycard, controlling individual lock programmer.
- **Lock Programmer:** A tool in the lock, to communicate with the card and to open the lock.
- **Lock:** Consists of a large magnet which operates with electric power to open and lock the door.
- **Card:** Guest card or master card used to open the lock.
2. SECURITY OBJECTIVES

- Allow guest to access the room
- Allow service personnel to access a room
- Prevent unauthorized access to your hotel room
- Every entry should be logged
3. WORK FLOW

- A guest arrives
- Hotel staff makes a new guest card using central encoder. The card contains room, lock no, duration for which the card is valid. The data is encrypted using a common key shared with the lock programmer.
- The Guest touches the card in front of Lock programmer
- The Lock programmer reads the data, decrypts the data and if data is valid sends electric signal to the lock to open the door.
3. TRUST BOUNDARIES

- Central encoder in a secure location
- The lock encoder
- The interface between lock and lock encoder
- The card
3. SECURITY CONTROLS

• The central encoder is only accessible to hotel staff
• The lock encoder is physically hard to modify
• The cards data are encrypted
• The locks programmer data is encrypted
3. ATTACKER TARGET ASSETS

• The encoding master keys (stored securely in the lock programmer and central encoder)
• The card itself
• The lock programmer
• The encoder
• The lock
4. THREATS

**Integrity:**
1. Someone steals your guest card and use it to open the door
2. Sneaks into the room when the door is open
3. Creating a duplicate guest card by reading the lock encryption key from the lock programmer
4. Changing the audit log of the lock programmer by physically accessing Lock log store.
5. Use weakness of crypto/key size to derive the key material
6. A malicious RFID reader reads the data and use it to create a duplicate card
7. Break the lock

**Confidentiality:**
1. Exposure of audit log by physically accessing the log in the Lock programmer

**Availability:**
1. The central encoder out of order, no way to make a new guest card
2. The power to the lock is lost, no way to open a door
5. VULNERABILITIES

- Psychological manipulation
- Accessible encryption keys in lock programmer
- Crypto algorithm/key size weakness
- Guest card easy to copy
- Lock physical strength weakness
6. THREAT QUANTIFICATION (SIMPLE EXAMPLE)

<table>
<thead>
<tr>
<th>Threats</th>
<th>Threat consequences</th>
<th>Probability of Threat</th>
<th>Damage of Threat</th>
<th>Attacker level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steal the Guest card</td>
<td>Unauthorized access</td>
<td>Medium</td>
<td>Medium</td>
<td>Loner</td>
</tr>
<tr>
<td>Sneaks into room when door is open</td>
<td>Unauthorized access</td>
<td>Low</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>malicious RFID reader reads the data and use it to create a duplicate card</td>
<td>Unauthorized access</td>
<td>Medium</td>
<td>Medium</td>
<td>Motivated individual</td>
</tr>
<tr>
<td>Changing the audit log of the lock programmer by physically accessing the log storage</td>
<td>Log integrity violation</td>
<td>Low</td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>Use weakness of crypto/key size to derive the key material</td>
<td>Access and integrity violation</td>
<td>Low</td>
<td>High</td>
<td>Organized crime group</td>
</tr>
<tr>
<td>The central encoder out of order, no way to make new guest card</td>
<td>Availability violation</td>
<td>Low</td>
<td>High</td>
<td></td>
</tr>
</tbody>
</table>
9. VERIFY SOME OF THESE THREATS IN REAL WORLD
RECAP

- Threat analysis deals with possible threats to a system and its consequences.
- Threats are inevitable for any system, careful threat domain analysis can save future trouble.
- OpenStack – as the next cloud platform – requires proactive threat analysis.