Threat Modelling - hacking the design

Mustafa Kasmani

Senior Cyber Security Consultant, Worldpay
• 12 years at Worldpay:
  Test (payment gateway) —> AppSec (CyberSecurity Consulting)
  a division of a major bank —> FTSE 100 —> merger talks…

• Worldpay - leader in global payments, 15 billion transactions processed in 146 countries, 126 currencies, 300+ APM’s.

• Global brands, 30 years of payments history, 5000+ colleagues across 25 offices in 13 countries.

• Change is the only constant - Transformation, Innovation & culture
• New office: Fintech Hub - complementing other sites in Romania.

• Partnering with Endava - building engineering capability, including Security Specialisms

• **Open roles** - meet us at the stand to find out more.
What is Threat Modelling?

Threat modelling is a process by which potential threats can be identified, enumerated, and prioritised – all from a hypothetical attacker’s point of view.

The purpose of threat modelling is to provide defenders with a systematic analysis of the probable attacker’s profile, the most likely attack vectors, and the assets most desired by an attacker.

Threat modelling answers the questions “Where are the high-value assets?” “Where am I most vulnerable to attack?” “What are the most relevant threats?” “Is there an attack vector that might go unnoticed?”

What are you building?
• Model system —> DFD’s, sequence flows, API contracts, etc.

What can go wrong?
• Identify threats —> STRIDE threat analysis

What should be done about it?
• Address threats —> Risk analysis

Is the threat analysis correct?
• Validate analysis —> Testing of controls
Why should it be done?

• Analyse the system from an **attackers point of view**, threat actors & motives, and enumerate assets to protect.

• Find flaws in the design and remediate when **easiest & cheapest** to do so.

• Create a **common understanding** of the system design amongst the architects, designers, developers, testers & security folk.

• Culture over Process over Tools: Security Maturity & Worldpay experiences
• The more **perspectives** you get into your threat model means better protection can be designed to the system.

• Certain features can become vulnerabilities when used by people with malicious intent.

• **Balance** between security -vs- usability -vs- cost -vs- other competing resources (opportunity cost).

• Build up library of patterns for which **risks** are known, understood & accepted by the stakeholders.

• Avoid **technical debt** being built up through better understanding prior to new features being added
Who should be involved?

- Architects, Designers, Developers, Testers, Security, + Anyone who has an interest in it:
  - Different perspectives - business fraud (operational processes / external entities), not just technical threats

- Security Champions in the team: Link between Development & Security:
  - scale AppSec capabilities, understand the system, maintain risk log, point of contact.
Use-cases

• As a security architect,
  • I want to do a threat model of …
  • So that I can design effective security controls mitigate the threats identified in the threat model.

• As a security tester,
  • I want to create a library of security tests for …
  • So that I can validate that the security controls in place are mitigating the threats identified in the threat model.
When should it be done?

- As early as possible!
- Influence direction, technology choice, system design
- Iterative - can re-visit once further details are known
- “The best time to plant an oak tree was 20 years ago. The next best time is now.” — wise words
• STRIDE - Microsoft Methodology (c.1999)
  – Explore this further later on in the workshop
• PASTA - (Process for Attack Simulation and Threat Analysis)
• VAST - (Visual Agile and Simple Threat Modelling)
In reality

• use a methodology for structure,
• But focus on how to find good threats, rather than the merits of one approach over another
  – each has its own strengths & weaknesses
• appropriate to what is being built, who is building it (skill-set), the prevalent risk appetite & culture
Workshop later

• Practical exercise of threat modelling a fictitious payments web application:
  – payments page, merchant portal, administration
  – actors, assets, distributed architecture

• Objective: put theory into practice
What are you building?

- Model the system - (appropriate level of detail)
- Trust boundaries -vs- Attack surface
- Data - in transit / on disk / in memory
- Actors - benign / malicious, internal / external, employees, suppliers / customers / partners / etc.
- Assets - physical, logical, configuration, code, intellectual property, API contract (e.g. Swagger spec)
• Data-Flow Diagrams
• Sequence Interaction Diagrams
• API contracts / Swagger definitions

• Keep It Simple - easy to understand
• Complexity is the enemy of Security
What can go wrong?

- Map attack surface
- Actors -vs- Motives
- STRIDE threat analysis
- Risk analysis
- Controls testing
• **Spoofing** - pretending to be someone / something else
• **Tampering** - modifying something that should not be modified
• **Repudiation** - denial of something that was done (true or not)
• **Information disclosure** - divulge information that should not be divulged, a breach of confidentiality
• **Denial of service** - prevent a system or service from being available or fulfilling its purpose
• **Elevation of privilege** - executing something without being allowed to do so
<table>
<thead>
<tr>
<th>What should be done ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spoofing</td>
</tr>
<tr>
<td>Tampering</td>
</tr>
<tr>
<td>Repudiation</td>
</tr>
<tr>
<td>Information disclosure</td>
</tr>
<tr>
<td>Denial of service</td>
</tr>
<tr>
<td>Elevation of privilege</td>
</tr>
</tbody>
</table>
Examples

• CCleaner
• Ranking of issues - risk assessment
• SDLC - DevSecOps -> iterative on-going assessment
• keep the security culture on-going
Controls testing

• Scoping assessments, targeted testing
  – Understand the system - testers get involved earlier on in the design.
  – Later tests are more targeted in approach, validation of controls rather than find new issues
  – Security built in right from the outset rather than being bolted on at the end - saves time & money!
Experiences

• What we’ve found
  • Experience at Worldpay - culture, what works in one place may not work in another - same for different teams.
  • Iterative process - get better over time, understand what works what doesn’t
  • Resistive teams - how to deal with them: hostile, resistive, unaware, enthusiastic
  • Management
Further reading

Thank you

Any questions?
Threat Modelling a fictitious payment web application - (workshop)

Mustafa Kasmani

Senior Cyber Security Consultant,
• As an Application Security Consultant,
• assess the design of this application,
• so that the risk profile of it can be established and that mitigating action can be taken