User-Story Driven Threat Modeling

Boston Code Camp 30
November 17, 2018
Robert Hurlbut
@RobertHurlbut

Who am I?

Robert Hurlbut
SVP, Threat Modeling Architect / Lead
Cyber Security Technology
Bank of America
Boston Code Camp 30 - Thanks to our Sponsors!

Platinum

Gold

Silver

In-Kind Donations

Agenda

What is Threat Modeling?

Threat Modeling Process

Threat Modeling in Agile / DevOps?

Modern Approaches
What is Threat Modeling?

What is threat modeling?

You probably (hopefully!) already do these in your security strategy:
  Penetration testing
  Vulnerability assessments
  DAST / SAST tools
  Other automated tools ...

But, if not threat modeling – you are missing a lot!
What is threat modeling, continued?

Something we all do in our personal lives ...
... when we lock our doors to our house
... when we lock the windows
... when we lock the doors to our car

What is threat modeling, continued?

When we ...
think ahead on what could go wrong
(\textit{i.e. the “what if” questions}),
weigh the risks,
and act accordingly ...

... we are “\textbf{threat modeling}”
What is threat modeling, continued?

**Threat modeling** is:
Process of understanding your system and potential threats against your system

i.e. *Critical Thinking* about Security

Approaches to Threat Modeling

**Asset-centric**
- Assets, Attack trees

**Software-centric**
- Secure design, DFDs

**Attacker-centric**
- Profile, patterns
Threat Modeling your House

Asset-centric
   Family, irreplaceable photos, valuable artwork

Software-centric
   Physical features (pool or front porch)

Attacker-centric
   Who might break in, current security system

What is threat modeling?

**Threat model** includes:
   understanding of system,
   identified threat(s),
   proposed mitigation(s),
   priorities by risk
Threat Modeling Process

1. Diagram / understand your system and data flows
2. Identify threats through answers to questions
3. Determine mitigations and risks
4. Follow through

When? Make threat modeling first priority

In SDLC – Requirements and Design phase(s):
Requirements > Design > Development > Test > Deployment

Threat modeling -> new requirements

Incremental threat modeling ->
Agile / DevOps
(User Stories, Abuser / Attacker Stories)
Threat Modeling Process

Understand the system

DFD – Data Flow Diagrams (MS SDL)

- External Entity
- Process
- Multi-Process
- Data Store
- Dataflow
- Trust Boundary
Understand the system

How do the interactors, processes and data stores connect?
Connect the info points with the data flow arrows.

Where are the trust boundaries?
For example:
• Browser (interactor) sends / receives data (data flow) with a web service (process) which saves / reads data (data flow) using a SQL Database (data store)
• Trust boundaries indicate where trust changes — authenticate / authorize / validate

Identify threats – Many Ways

STRIDE
Attack Trees
Bruce Schneier - Slide deck

Threat Libraries
CAPEC, ATT&CK, OWASP Top 10, SANS Top 25

Checklists
OWASP ASVS, OWASP Proactive Controls

Card Games
OWASP Cornucopia, Elevation of Privilege

Use Cases / Abuse Cases
STRIDE Framework – Data Flow

<table>
<thead>
<tr>
<th>Threat</th>
<th>Examples</th>
<th>Property we want</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spoofing</strong></td>
<td>Pretending to be someone else</td>
<td>Identity Assurance</td>
</tr>
<tr>
<td><strong>Tampering</strong></td>
<td>Modifying data that should not be modifiable</td>
<td>Integrity</td>
</tr>
<tr>
<td><strong>Repudiation</strong></td>
<td>Claiming someone didn’t do something</td>
<td>Non-repudiation</td>
</tr>
<tr>
<td><strong>Information Disclosure</strong></td>
<td>Exposing information</td>
<td>Confidentiality</td>
</tr>
<tr>
<td><strong>Denial of Service</strong></td>
<td>Preventing a system from providing service</td>
<td>Availability</td>
</tr>
<tr>
<td><strong>Elevation of Privilege</strong></td>
<td>Doing things that one isn’t suppose to do</td>
<td>Least Privilege</td>
</tr>
</tbody>
</table>

Identify Threats – Functional

Input and data validation
Authentication
Authorization
Configuration management
Data Classification
- Public, Proprietary, Confidential
Identify Threats – Functional

Session management
Cryptography
Parameter manipulation
Exception management
Auditing, logging, and monitoring

Identity Threats – Ask Questions

Who’s interested in app and data (threat agents)?
What goals (assets)?
What attack methods (how)?
Any attack surfaces (trust boundaries) exposed?
Any input/output (data flows) missing?
Determine mitigations and risks

Mitigation Options:
- Leave as-is
- Remove from product
- Remedy with technology countermeasure
- Warn user

Make the mitigations part of your Security acceptance criteria

What is the risk associated with the vulnerability and threat identified?

Risk Rating

Risk is product of two factors:
- Ease of exploitation
- Business impact
Follow through

Document findings and decisions

File bugs or new requirements (as stories)

Verify bugs fixed / new requirements (stories) implemented

Did we miss anything? Review again

Anything new? Review again

---

Threat Modeling in Agile / DevOps?
## Threat Modeling approaches – Waterfall vs Agile*

<table>
<thead>
<tr>
<th>Waterfall: Threat Model Documents</th>
<th>Agile: Bugs and conversations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Model</strong></td>
<td></td>
</tr>
<tr>
<td>• Big complex scope</td>
<td>• Scope tiny: this sprint’s change</td>
</tr>
<tr>
<td>• System diagrams and essays</td>
<td>• Big picture as security debt</td>
</tr>
<tr>
<td>• Gates, dependencies</td>
<td></td>
</tr>
<tr>
<td><strong>Finding Threats</strong></td>
<td></td>
</tr>
<tr>
<td>• Brainstorm</td>
<td>• Same, aim at in-sprint code</td>
</tr>
<tr>
<td>• STRIDE</td>
<td></td>
</tr>
<tr>
<td>• Kill Chain</td>
<td></td>
</tr>
<tr>
<td><strong>Fixes</strong></td>
<td></td>
</tr>
<tr>
<td>• Controls</td>
<td>• Spikes to understand</td>
</tr>
<tr>
<td>• Mitigations</td>
<td>• Security-focused stories in sprint, backlog, or epic</td>
</tr>
<tr>
<td>• Test Cases</td>
<td>• Security acceptance criteria</td>
</tr>
<tr>
<td><strong>Quality</strong></td>
<td></td>
</tr>
<tr>
<td>• Test plans</td>
<td>• Test automation</td>
</tr>
</tbody>
</table>

*Adapted from Adam Shostack’s talk at BlackHat 2018 on Threat Modeling in 2018

### When?

There are many out-of-band activities (as opposed to inline activities such as coding, etc.)

- Sprint planning
- Spikes

Add Threat Modeling as another out-of-band activity and/or

In addition to when you create User Stories (or Abuser Stories) you determine Threat Model to help you determine Tasks and Acceptance Criteria / Security Tests

---

*Adapted from Adam Shostack’s talk at BlackHat 2018 on Threat Modeling in 2018*
User stories

User stories written typically like this:

As a <type of user>, I want <some goal> so that <some reason>

Examples:
• As a user, I want to backup my entire hard drive so that I can have a redundant copy of my stuff.

• As a power user, I want to specify files or folders to backup based on file size, date created, and date modified so that I can improve the backup time.

• As a user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved.

Security User stories

Security user stories are similar to regular user stories, but are sometimes more difficult to manage – there may be too many of them.

Examples:
• As a user, I want to log into the application so that I can conduct transactions.

• As a user, I want to be able to see my account information and not other users’ information so that I know my account information is safe.

• As an admin, I want to have access to configuration settings in the application so that I can update configuration settings.
Abuser stories

Abuser / attacker stories do this differently:

As <someone with malicious intent>, I want to <do some bad thing>

Examples:
• As a hacker, I want to read the application log files so that I can gather data to prepare for an attack.
• As an insider, I want to access a customer’s account information so that I can steal their identity.
• As a disgruntled employee, I want to change pricing for some products so that I can undermine company sales.

See OWASP Abuse Case Cheat Sheet for help in creating these: https://www.owasp.org/index.php/Abuse_Case_Cheat_Sheet

User Story / Abuser Story / Threat Model – Example

User Story: As an application, I want to write out application log files so that I may track what’s happening with the application.

Abuser Story: As a hacker, I want to read the application log files so that I can gather data to prepare for an attack.

Threat Model:

Identified Threat(s): Information Disclosure, Repudiation

Mitigation / Control(s): Prevent reading of log files by the wrong person / entity by applying access control.
Log and monitor who / what reads the log files.

Abuser Story Tasks:
• Apply strong ACLs on the application log files (read-only and write-only per certain user accounts).
• Apply logging / monitoring of who / what accesses log files.

Acceptance Criteria / Security Tests (Given … When … Then):
Given: An arbitrary hacker When he / she tries to read application log file Then the log file will not be read and the attempt at reading the log file will be monitored.
Typical Threat Modeling Session (Agile / DevOps version)

In Sprint Planning:
• Team
• Focused scope to set of stories
• Understand requirements, keep business / technical goals in mind

**Important:** Be honest, leave ego at the door, no blaming!

Prioritize issues in the backlog

Work through your user stories / abuser stories – determine threats and mitigations as you go

As you find issues, write these to the backlog

Prioritize based on risk
Modern Approaches

Incremental Threat Modeling
   Agile approaches – Irene Michlin (@IreneMichlin)

Lateral Movement
   “The Industrial Revolution for Lateral Movement”
   BlackHat 2017

   Think STRIDE + LM

Privacy by Design (addressing GDPR, etc.)
   STRIPED + LM
Kill Chain as Alternative to STRIDE

Kill Chain – useful for operational threat models


Threat Modeling as Code – applying “Spec” based systems

- ThreatPlaybook (@abhaybhargav)

Providing a way to combine User / Abuser stories, threat scenarios, and automated security testing.
Threat Modeling as Code – applying “Spec” based systems

- ThreatPlaybook (@abhaybhargav)

Threat Spec - Have developers and security engineers write threat specifications alongside code, then dynamically generate reports and data-flow diagrams from the code.
Threat Modeling as Code – applying “Spec” based systems

• ThreatSpec @ThreatSpec
• Fraser Scott @zeroXten

Threat Model & CHILL

Resources - Books

Threat Modeling: Designing for Security
Adam Shostack

Brook S.E. Schoenfield

Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis
Marco Morana and Tony UcedaVelez

Measuring and Managing Information Risk: A FAIR Approach
Jack Jones and Jack Freund
Resources - Tools

Microsoft Threat Modeling Tool

ThreatModeler – Web Based (in-house) Tool
http://myappsecurity.com

IriusRisk Software Risk Manager
https://iriusrisk.continuumsecurity.net

OWASP Threat Dragon
https://www.owasp.org/index.php/OWASP_Threat_Dragon

Resources - Tools

Attack Trees – Bruce Schneier on Security

Elevation of Privilege (EoP) Game

OWASP Cornucopia
https://www.owasp.org/index.php/OWASP_Cornucopia

OWASP Application Security Verification Standard (ASVS)

OWASP Top 10 Proactive Controls 2018
https://www.owasp.org/index.php/OWASP_Proactive_Controls
Resources - Tools

ThreatPlaybook  
https://we45.gitbook.io/threatplaybook

ThreatSpec  
https://threatspec.org/

Questions?

@RobertHurlbut
Thank you!

Slides: https://roberthurlbut.com/r/BCC30TM