Practical Threat Modeling

SecAppDev 2018
Material

tinyurl.com/secappdev2018
Sebastien Deleersnyder

- 5 years developer experience
- 15+ years information security experience
- Application security consultant Toreon

- Belgian OWASP chapter founder
- OWASP volunteer
- www.owasp.org

- Co-founder www.BruCON.org

Whiteboard Hacking - Toreon 2018
Threat modeling introduction

- Threat modeling in a secure development lifecycle
- What is threat modelling?
- Why threat modeling?
- Threat modeling stages
- Diagrams
- Identify threats
- Addressing threats
- Document a threat model
- Tools
You can’t use network layer protection (firewall, SSL, IDS, hardening) to stop or detect application layer attacks.

Your security “perimeter” has huge holes at the application layer.
Secure development lifecycle

Web/mobile application project (acquisition/development)

- **Design**: Coding guidelines, Source code review (static)
- **Build**: Threat modelling
- **Test**: Security testing (dynamic)
- **Production**: Configuration guidelines, WAF tuning

SDLC assistance

Training
Flaws versus bugs

Security Design Flaws

- Introduced because of lack of security requirements, errors in design, lack of secure design knowledge, lack of architecture design review
- Cannot be identified by tools since lack contextual knowledge of the application
- Can be identified with threat modeling/secure architecture reviews

Security Coding Bugs

- Coding errors that result in vulnerabilities
- Can be identified with source code analysis and tools
- Requires developers understanding secure coding and following secure coding standards

Whiteboard Hacking - Toreon 2018
Threat modeling

• Threat modelling is the activity of identifying and managing application risks

• Threat modelling is also known as Architectural Risk Analysis
Why threat modeling?

• Prevent security design flaws when there’s time to fix them
• Select mitigation strategy and techniques based on identified, documented and rated threats.
• Identify & address greatest risks
• Ability to prioritize development efforts within a project team based on risk weighting
• Increased risk awareness and understanding
• Mechanism for reaching consensus and better trade-off decisions
• Means for communicating results
• Cost justification and support for needed controls
• Artifacts to document due diligence for each software project
Threat modelling stages

Step 1

Diagram

What are we building?
Threat modelling stages

Step 1
Diagram

What are we building?

Step 2
Identify threats

What can go wrong?
Threat modelling stages

- **Step 1**: Diagram
  - What are we building?

- **Step 2**: Identify threats
  - What can go wrong?

- **Step 3**: Mitigate
  - What are we doing to defend against threats?
Threat modelling stages

Step 1
Diagram
What are we building?

Step 2
Identify threats
What can go wrong?

Step 3
Mitigate
What are we doing to defend against threats?

Step 4
Validate
Validate steps 1-3
Report
Diagrams

• Define scope
• Good understanding context / objectives
• Understand how the software works
• Who interacts with the software?
• With Data Flow Diagrams, Sequence Diagrams, State diagrams ...
• Identify attack surfaces
• Foundation for threat analysis
Diagramming

• Use DFDs (Data Flow Diagrams)
  – Include processes, data stores, data flows
  – Include **trust boundaries**
  – Diagrams per scenario may be helpful
• Update diagrams as web application changes
• Enumerate assumptions, dependencies
• Number everything (if manual)
# DFD Basics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>External Entity</strong></td>
<td>• Represents entities outside the application that interact with the application via an entry point</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>• Represents tasks that handle data within the application; tasks may process data or perform actions based on the data</td>
</tr>
<tr>
<td><strong>Data Store</strong></td>
<td>• Represents locations where data is stored; data stores do not modify data, they only store it.</td>
</tr>
<tr>
<td><strong>Data Flow</strong></td>
<td>• Represents data movement within applications; the arrow tells the direction of data movement</td>
</tr>
<tr>
<td><strong>Trust Boundary</strong></td>
<td>• Represents the change of trust levels as data flows through the application</td>
</tr>
</tbody>
</table>
Context diagram

![Context Diagram](image)

- Analysis instructions
- Administrator
- Integrity App
- Resource integrity information

Whiteboard Hacking - Toreon 2018
Level 1 Diagram
Identify threats

- Based on diagrams
- STRIDE analysis
- Focus on identifying threats
STRIDE

- **Spoofing**
  - Can an attacker gain access using a false identity?

- **Tampering**
  - Can an attacker modify data as it flows through the application?

- **Repudiation**
  - If an attacker denies doing something, can we prove he did it?

- **Information Disclosure**
  - Can an attacker gain access to private or potentially injurious data?

- **Denial of Service**
  - Can an attacker crash or reduce the availability of the system?

- **Elevation of Privilege**
  - Can an attacker assume the identity of a privileged user?
Apply STRIDE Threats to Each Element

Apply the relevant parts of STRIDE to each item on the diagram

- External Entity – S, T
- Process – S, T, R, I, D, E
- Data store, data flow – T, I, D
  - Data stores that are logs – T, I, D, and R

This is why you number things
# Example

<table>
<thead>
<tr>
<th>Admin</th>
<th>Mitigations</th>
<th>Vulnerabilities</th>
<th>Mitigations</th>
<th>Vulnerabilities</th>
<th>Mitigations</th>
<th>Vulnerabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>User/PW</td>
<td>SSL Cert</td>
<td></td>
<td>No audit log</td>
<td>No Audit log</td>
<td></td>
<td>SSL</td>
</tr>
<tr>
<td>SSL</td>
<td></td>
<td></td>
<td>SSL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No access control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Whiteboard Hacking - Toreon 2018
Addressing threats

- Cover all threats
- Identify controls already in place
- Handle threats not (completely) covered
Addressing each threat

Mitigation patterns

- **Authentication**
  - Mitigating spoofing

- **Integrity**
  - Mitigating tampering

- **Non-repudiation**
  - Mitigating repudiation

- **Confidentiality**
  - Mitigating information disclosure

- **Availability**
  - Mitigating denial of service

- **Authorisation**
  - Mitigating elevation of privilege
Mitigation patterns

• Apply appropriate secure design strategies
• Leverage proven best practices
• Reuse organisation security services, e.g.,
  – Single-Sign-On, Log Server
• Do not reinvent the wheel
For threats not (completely) covered

- Redesign to eliminate
- Apply standard mitigations
- Create new mitigations
- Accept vulnerability in design
Risk-based Threat Management

“The only truly secure system is one that is powered off, cast in a block of concrete, and sealed in a lead-lined room with armed guards - and even then I have my doubts.”

Prof Gene Spafford

Source: [http://spaf.cerias.purdue.edu/quotes.html](http://spaf.cerias.purdue.edu/quotes.html)
OWASP risk rating

![Diagram showing OWASP risk rating process]

### Injection Example

<table>
<thead>
<tr>
<th>Threat Agent</th>
<th>Attack Vector</th>
<th>Weakness Prevalence</th>
<th>Weakness Detectability</th>
<th>Technical Impact</th>
<th>Business Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Easy</td>
<td>Widespread</td>
<td>Easy</td>
<td>Severe</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Average</td>
<td>Common</td>
<td>Average</td>
<td>Moderate</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Difficult</td>
<td>Uncommon</td>
<td>Difficult</td>
<td>Minor</td>
<td></td>
</tr>
</tbody>
</table>

**7 weighted risk rating**

2.33

Whiteboard Hacking - Toreon 2018
## Example

<table>
<thead>
<tr>
<th>Threat</th>
<th>Description</th>
<th>Vector</th>
<th>Prevalence</th>
<th>Detectability</th>
<th>Impact</th>
<th>Rating</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>TH – 01</td>
<td>Credentials can be brute forced</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>7.00</td>
<td>High</td>
</tr>
<tr>
<td>TH – 02</td>
<td>No security rules on password</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>6.00</td>
<td>Medium</td>
</tr>
<tr>
<td>TH – 03</td>
<td>No SSL for Android App</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4.67</td>
<td>Medium</td>
</tr>
<tr>
<td>TH – 04</td>
<td>No SSL active for admin module</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4.00</td>
<td>Medium</td>
</tr>
<tr>
<td>TH – 05</td>
<td>No accountability of Drupal updates</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2.33</td>
<td>Low</td>
</tr>
<tr>
<td>TH – 06</td>
<td>API calls can be tampered with</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2.00</td>
<td>Low</td>
</tr>
<tr>
<td>TH – 07</td>
<td>Fake IDs can be used</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2.00</td>
<td>Low</td>
</tr>
</tbody>
</table>

Low: 1-3, Medium: 4-6, High: 7-9
Communicate Your Threat Model

You cannot just “write and throw out” a security document

- Recipients often won’t understand it
Communicate Your Threat Model

To increase adoption
• Present the results to the audience, in person
• Discuss the countermeasures – cost vs. impact
• Complete the threat model with a proposed action list that you know is acceptable

Typical audience
Architects
• Should integrate the proposition to update the design

Developers
• Should benefit from the model transparently, through updated specification

Security testing team
• Now know precisely what to test!

Software editor
• If you are acquiring software, you can add the threat model to the software acceptance procedure
Update Your Threat Model

• First Threat Model during design
• Update Threat Model during technology decisions
• Review Threat Model before implementation
• Refine and verify Threat Model during security review
• Iterate
Free Tools

• Whiteboards!
• Mind-Mapping diagramming tools such as FreeMind
• Microsoft Threat Modeling Tool 2016
  https://www.youtube.com/watch?v=G2reie1skGg (demo)
• Gliffy Adds Dynamic Diagrams to Your Confluence Wiki Pages
  https://www.gliffy.com/products/confluence-plugin/
• ThreatSpec, developers and security engineers write threat specifications alongside code https://threatspec.org/
• Mozilla SeaSponge, browser-based graphical threat modeling tool
  http://mozilla.github.io/seasponge
• OWASP Threat Dragon Project
  https://www.owasp.org/index.php/OWASP_Threat_Dragon
• Trike was introduced as an open source threat modeling methodology and tool introduced in 2006
Commercial Tools (no particular order)

- Microsoft Visio (Windows)
- ConceptDraw Pro (MacOS)


- PTA Technologies http://www.ptatechnologies.com/

- Amenaza SecuriTree (Based on Attack trees vs Software centric approach) http://www.amenaza.com/

- IriusRisk by Continuum Security https://iriusrisk.continuumsecurity.net/

- Security Compass SD Elements is a Software Security Requirements Management platform that includes automated threat modeling capabilities https://www.securitycompass.com/threatmodeling/

- isograph AttackTree https://www.isograph.com/software/attacktree/
Resources

Books

• Threat Modeling (Adam Shostack, MS)
• Threat Modeling (Swiderski, Snyder) – older
• Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis (P.A.S.T.A) (Marco Morana and Tony “UV.”)
• FAIR - Measuring and Managing Information Risk: A FAIR Approach (Jack Freund and Jack Jones)

Online

• https://en.wikipedia.org/wiki/Threat_model
• https://www.owasp.org/index.php/Application_Threat_Modeling
• BruCON 0x06 - Keynote - Adam Shostack - https://www.youtube.com/watch?v=-2zvfevLnp4
OWASP – On-going

- OWASP threat model project creation
- With the summit group, go over all the outcomes
- Publish all outcomes as soon as project is started
- Continue work on the cheat sheets
- Start work on a model to compare all threat methodologies, tools and techniques

Join the discussion at

[https://owasp.slack.com/messages/C1CS3C6AF](https://owasp.slack.com/messages/C1CS3C6AF)

OWASP Threat Modeling Slack channel
Templates for this Workshop

• Template to document a threat model
• Template to calculate risk levels of identified threats
• Threat modeling Visio Stencil
That’s All Folks

You can contact me through

• Toreon seba@toreon.com
• OWASP seba@owasp.org
• Twitter @SebaDele
• OWASP TM Slack channel