Threat Modelling Strategies

Roy Harrow
BCS DevSecOps Group
21st March 2023
Roy Harrow

- Applications development
- Methods and Tools
- Standards and procedures
- Change and configuration management
  - Secretary of BCS CMSG
- Design and Architecture
- Security
  - Web Security and IAM
  - Infrastructure / Designed DIY monitoring system
  - Consultancy and architecture
  - Solution design
  - Product assurance

- Sectors
  - Financial services
  - Central and Local Government
  - Communications, Health Care, Transport, Retail + many others

- IBM Security
  - 2008-2022 (14 years)
- Sainsbury’s Information Security
  - September 2022+

- Chair of BCS DevSecOps group
Huge range of business applications
- eCommerce: groceries online, general merchandise
- Store systems, point of sale, warehouse, delivery and logistics
- Contact centre, corporate services many more

Wide range of technologies
- On premise: mainframe, midrange and specialised technologies
- Cloud hosted services and many SaaS applications

Large number of engineering teams
- Linked to product managers
- Using modern agile practices and CI/CD pipelines integrated with various security processes and tooling

https://sainsburys.jobs/roles/digital-tech-data/
Agenda

• What is Threat Modelling?
• Why do it?
• When to perform it?
• How to do threat modelling?
• Who should do threat modelling?
• Threat modelling tools
• Summary & Conclusions

These are my personal views and don’t represent policies and processes from my current or previous employers
What is Threat Modelling?

A process to attempt to identify security weaknesses in an application

• *Before someone else does*

Aims to help to improve the security of IT applications

• Ideally *before they are built*

The focus tends to be on thinking about deliberate attempts to circumvent an application’s security controls - aka “threats”

• But also needs to consider accidents
• Deliberate attempts could be targeted or just “random” / opportunistic
“Traditional” Approach

- A focus on functional requirements
  - To drive the design, build and testing of applications

- Some consideration for non-functional requirements, in particular
  - Usability / User Experience
  - Performance and availability
  - Depending on type of business
    - Regulatory or industry specific compliance requirements e.g. healthcare or financial services

- Security requirements/controls
  - May be added as application evolves through agile development iterations
  - Some driven by compliance
  - Some derived from organisational “baseline” security requirements

- Penetration testing at the end
  - With mixed results

- Vulnerability scanning and patching in the live environment
  - With mixed results
Why do Threat Modelling?

- **To identify potential vulnerabilities early**
  - Ideally during design stages
  - To be able to influence design and build before it is too late

- **To include input from all stakeholders**
  - To ensure all “angles” are considered, both technical and non-technical

- **To drive security controls based on business priorities**
  - By taking inputs from product owners and business representatives

- **To encourage a “security mindset”**
  - To influence the selection and design of future IT services
Threat Modelling – Main Steps

- We need to understand the system or application
  - Its business purpose and information being processed
- Then need to consider what could go wrong
- What can we do about it?
- And finally
  - How did we do?
Threat Modelling – Key Activities

• We need to understand the system or application
  • Business purpose and information being processed

• Then need to consider what could go wrong

• What can we do about it?

• And finally....
  • How did we do?

• Scope + Context (Business + Technical)
  • A sprint or a component
  • A new release
  • Diagrams are common

• Brainstorm possible threats or attacks
  • Application profiling questions
  • Common threat/attack models

• Identify or design countermeasures
  • to reduce risk

• “Fit for purpose” given context?
  • Coverage
  • Lessons learned?
Clarifying Scope and Business Context

• Essential – to drive thinking about relevant risks
• Diagrams are useful – but not essential
  • DFD or Process Flow
• Can be iterative or evolving during a project
• Could use an application profiling questionnaire
  • If combined with some “best practice” or domain specific guidance, such as
    • OWASP Top 10 Proactive Security Controls
    • PCI-DSS Guidance
Threat Modelling Simplified #1

Example Application Profiling Questions

- **Business purpose**
  - Importance of the application to the organisation

- **Information being handled**
  - Customer data
  - Product information

- **Sensitivity of data being handled**
  - PCI-DSS cardholder information
  - Personal sensitive such as healthcare
  - Company secrets
  - State secrets

- **Mode of access**
  - Internal from trusted devices and networks
  - External via Internet

- **Users of the application**
  - Internal
  - External
  - Trusted third-parties

- **Modes of access**
  - Browser
  - Fat-client
  - APIs

- **Main interfaces – ideally with a diagram**
  - Inputs
  - Outputs

- **Hosting Arrangements**
  - On premise
  - Cloud
  - SaaS

Led by Security Expert
Threat Modelling Simplified #2

Using Checklists for Security Controls

Using the information from the application profiling questionnaire

- Select relevant security controls from a framework
- Create a tailored set of controls

Examples of general security checklists you could use are:

- OWASP Top 10 Proactive Controls 2018
- OWASP Cheat Sheets - includes “Secure Product Design”
- UK National Cyber Security Centre (NCSC) – 14 Cloud Security Principles

Domain specific checklists and compliance frameworks include

- PCI-DSS for the protection of payment card information
- NHS Data Security and Protection Toolkit (DSPT)
Review proposed new applications as early as possible with the development team

- Ideally before detailed design is completed
- Review and revise the responses as the project progresses
- Review for major changes to the application

Best led by a security expert

- But the questions need input from the application team

Pros

- Relatively “light touch” for project team
- Can be combined with other more formal methods for identifying threats

Cons

- Takes time to develop questionnaire
- “Light touch” involvement from project team
  - May not encourage a true collaborative approach + ownership of security requirements
  - May struggle to scale, given amount of input required from security expert
• C1: Define Security Requirements
• C2: Leverage Security Frameworks and Libraries
• C3: Secure Database Access
• C4: Encode and Escape Data
• C5: Validate All Inputs
• C6: Implement Digital Identity
• C7: Enforce Access Controls
• C8: Protect Data Everywhere
• C9: Implement Security Logging and Monitoring
• C10: Handle All Errors and Exceptions
Examples of OWASP Cheat Sheets

• Topical advice for Developers
  • Authentication
  • Authorisation
  • Cryptographic Storage
    • Encryption of data at rest
  • Database Security
  • Docker and Kubernetes Security
  • Input Validation
  • Secrets Management

• Advice on defending against common vulnerabilities
  • Clickjacking Defence
  • Cross Site Scripting Prevention
  • Denial of Service Protection
Beyond Application Profiling

Importance of collaborative approach
- Input from all stakeholders

Needs to be able to scale
- Given low number of “security experts” vs. developers

Requires
- Deeper understanding of the context – to identify more subtle threats
- Techniques and sources of information to help identify threats
- More structure – to have confidence in coverage
- Standardised processes that can be repeated and adapted for many projects
Threat Modelling Manifesto Four Key Questions

1. What are we working on?
   • Typically supported using a diagram – such as a DFD
   • With a clear boundary to define the scope of the application – decomposing large and complex applications
   • Identifies key “assets”

2. What can go wrong?
   • Threats that could impact the security or privacy of the application
   • List of potential weaknesses in the design or implementation

3. What are we going to do about it?
   • Actions to mitigate the impact of the threats identified
   • Countermeasures or additional security controls
   • Prioritisation of actions.

4. Did we do a good enough job?
   • Threats identified?
   • Risks reduced – through effective countermeasures
   • Lessons learned – e.g. new recommended “standard” security controls for the organisation

https://www.threatmodelingmanifesto.org/
Thinking like an Attacker

What can we learn from TV detectives?
## Criminal Investigation Techniques

### Crime Scene
- **Demarcation**
- Gather evidence
- Background and context

### Assessment of suspects
- **Motivation**
  - e.g. financial gain or revenge
- **Means / Method**
  - Tools, skills,
- **Opportunity**
  - e.g. access to crime scene
- **Relationship** with victim
  - May be important to understand motivation

### Digital forensics
- **Scope** of system under attack
  - Boundaries
  - Business purpose, context
- **Motivation**
  - Financial gain, digital harm
  - Access to confidential information
- **Means/Method**
  - Tools, Techniques, Procedures (TTP)
- **Opportunity**
  - System access requirements
- **Relationship**
  - May be important to understand

Crime Prevention

Aims to understand

- Motivations for crime - for example financial reward without being detected
- Characteristics of a target

Aims to

- Increase likelihood of detection
- Minimise the reward

Situational crime prevention

- Attempts to reduce opportunities to commit crimes
- Make it more difficult to break the law in everyday situations.
- It looks at
  - the types of offences people commit,
  - the places where they offend, and
  - aims to prevent them at the point of their intersection.
- A pre-emptive strategy.

Possible Drivers

Risk Reduction

Countermeasures

Threat Intelligence
Threat Modelling Diagrams

• Useful for Scoping and Identifying Potential Targets for Attack
  • Data Flow Diagram (DFD)
  • Process Flow Diagram (PFD)
  • C4 Model - architectural diagrams
    • Context, Container, Component and Code

• Attack Tree Diagrams
  • Explains the steps of an attack
    • Bruce Schneier, 1999
    • Synopsis, 2015
Data or Process Flow Diagrams

• Helps define scope
• Aids understanding of data flows
• Provides structure for assessing risks
• Data Flow Diagrams (DFD) are the most common, for example
Attack Trees

- Commonly combined with other techniques such as STRIDE.
- Show attacks on a system in tree form.
- The tree root is the goal for the attack, and the leaves are ways to achieve that goal.
- Each attack goal is represented as a separate tree.
Threat Modelling Techniques

Threat Identification

- Q. What might cause us to breach....?
- CIA - Confidentiality, Integrity and Availability
- Compliance framework

Threat Classification

- Q. Could we be vulnerable to certain types of attack?
  - STRIDE
    - Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege
  - OWASP lists of top 10 types of common security vulnerabilities
    - Browser-based web applications
    - Mobile applications
    - API services
CIA approach to Threat Modelling

Confidentiality
- Will we be storing or handling any sensitive information?
- How will we be protecting it?

Integrity
- What are the consequences of an accidental or deliberate data corruption or unauthorised change?
- Why might someone want to change some data?
- What controls exist to prevent or detect unauthorised changes?

Availability
- How long could the business operate without the system?
- Have we planned any controls to help ensure availability?
<table>
<thead>
<tr>
<th>Threat</th>
<th>Property Violated</th>
<th>Threat Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Spoofing Identity</td>
<td>Authenticity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pretending to be something or someone other than yourself</td>
</tr>
<tr>
<td>T</td>
<td>Tampering with data</td>
<td>Integrity</td>
</tr>
<tr>
<td>R</td>
<td>Repudiation</td>
<td>Non-repudiation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Denying that you did something</td>
</tr>
<tr>
<td>I</td>
<td>Information disclosure</td>
<td>Confidentiality</td>
</tr>
<tr>
<td>D</td>
<td>Denial of service</td>
<td>Availability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exhausting computing resources needed to support the service</td>
</tr>
<tr>
<td>E</td>
<td>Elevation of privilege</td>
<td>Authorisation</td>
</tr>
</tbody>
</table>

Threat Modelling Using Lists of Common Vulnerabilities

- OWASP lists of top 10 types of common security vulnerabilities
  - Browser-based web applications
  - Mobile applications
  - API services

- Cloud Security Alliance
  - Generic cloud security weaknesses

- Common Attack Pattern Enumeration and Classification (CAPEC)
  - Similar to OWASP Top 10

- Common Vulnerability Scoring System (CVSS)
  - Can be used to prioritise vulnerabilities
  - e.g. NIST National Vulnerability Database (NVD)
Common Attack Pattern Enumeration and Classification (CAPEC)

- A catalogue of common **attack patterns** that helps explain how adversaries exploit weaknesses in applications
- **Attack Patterns** are descriptions of the common attributes and approaches employed by adversaries to exploit known weaknesses in cyber-enabled capabilities
- Some well-known **attack patterns**
  - SQL Injection (CAPEC-66)
  - Cross-Site Scripting (CAPEC-63)
  - Buffer Overflow (CAPEC-100)
  - Clickjacking (CAPEC-103)
  - Cross Site Request Forgery (CAPEC-62)

[https://capec.mitre.org/](https://capec.mitre.org/)
Lockheed Martin Cyber Kill Chain

• 7 steps attackers commonly use
  • Reconnaissance
  • Weaponization
  • Delivery
  • Exploitation
  • Installation
  • Command and Control (C2)
  • Actions on Objectives

• Threat modelling would assess the potential for each of these

MITRE ATT@CK Knowledge Base

- Attack tactics and techniques by platform, such as
  - Cloud
  - Mobile
  - Operating system family: Window, Linux
  - At a lower-level than Lockheed Martin Cyber Kill Chain or STRIDE
- Classified by stage of attack, e.g.
  - Reconnaissance
  - Initial Access
  - Execution
  - Persistence etc...
- Mitigations

- Useful when a more detailed assessment is required

https://attack.mitre.org/
Threat Modelling Methods

- **OWASP Threat Modelling Method**
- **The Process for Attack Simulation and Threat Analysis (PASTA)**
  - Risk-centric modelling method
- **LINDDUN**
  - A privacy focussed method
  - Linkability, Identifiability, Nonrepudiation, Detectability, Disclosure of information, Unawareness, Noncompliance
- **NIST Data-Centric System Threat Modelling – SP 800-154**
- **Persona non Grata (PnG)**
  - Focuses on the motivations and skills of human attackers.
- **The SEI Hybrid Threat Modelling Method (hTMM)**
- **Vendor approaches**
  - e.g. **Microsoft** and **Synopsys**
  - **Visual, Agile, and Simple Threat (VAST)** – from **Threatmodeler**
OWASP Threat Modelling

Decompose
- Decompose the Application
- External Dependencies
- Entry Points and Exit Points
- Assets
- Trust Levels
- Data Flow Diagrams

Determine and Rank
- Determine and Rank Threats
- Threat Categorisation e.g. using STRIDE

Countermeasures
- Determine Countermeasures and Mitigation
- Typically uses the OWASP Application Security Framework (ASF) or
- STRIDE threat mitigations

https://owasp.org/www-community/Threat_Modeling
Process for Attack Simulation and Threat Analysis (PASTA)

• Risk-centric
  • Identification
  • Classification and prioritisation
  • Highest and most relevant
  • Not just technical issues

• Seven step process to ensure business objectives are understood

• Benefits
  • Business Context is Prime
  • Tests viability
  • Attacker perspective

• Invented in 2015 by Tony UcedaVélez

https://versprite.com/blog/what-is-pasta-threat-modeling/
LINDDUN – Privacy Threat Modelling

- Designed to help identify and mitigate privacy threats:

  ![Diagram](https://www.linndun.org/)

- Likely to be used alongside a security oriented method, such as STRIDE

https://www.linndun.org/
Threat modelling is a form of risk assessment that models aspects of the attack and defence sides of a particular logical entity, such as a piece of data, an application, a host, a system, or an environment.

Data-centric threat modelling

- Focused on protecting particular types of data within systems.

This standard defines principles for data-centric threat modelling.

https://csrc.nist.gov/publications/detail/sp/800-154/draft
Risk Treatment

Assess / prioritise risks
- Focus on impact – if it is too hard to assess likelihood

Potential treatment strategies
- Reduce – the best outcome, if feasible and affordable
- Transfer – e.g. insure or outsource
- Avoid – e.g. disable or isolate
- Accept – if within risk appetite

Identify potential countermeasures
- Security controls

Risk Acceptance
- Risk Register
# Common Mitigations or Controls by type of Risk or Threat

<table>
<thead>
<tr>
<th>Type of Risk</th>
<th>Mitigation Strategy / Security Controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spoofing</td>
<td>Strong authentication</td>
</tr>
<tr>
<td></td>
<td>Digital signatures</td>
</tr>
<tr>
<td>Tampering</td>
<td>Access controls</td>
</tr>
<tr>
<td></td>
<td>Check-sums, hash-totals and signatures on data items</td>
</tr>
<tr>
<td>Repudiation</td>
<td>Strong authentication</td>
</tr>
<tr>
<td></td>
<td>Audit logs</td>
</tr>
<tr>
<td>Information Disclosure</td>
<td>Access controls</td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
</tr>
<tr>
<td>Denial of Service</td>
<td>Quotas / throttling of transaction volumes</td>
</tr>
<tr>
<td>Elevation of Privilege</td>
<td>Access controls</td>
</tr>
<tr>
<td></td>
<td>Hardened system configuration</td>
</tr>
</tbody>
</table>
## Different Types of Threat Modelling Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attack Trees</strong></td>
<td>A graphical representation of an attack scenario, which helps identify possible threats and their relationships with each other.</td>
</tr>
<tr>
<td><strong>STRIDE Analysis</strong></td>
<td>Evaluates the threat landscape from the perspective of six common risks (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege). Allows for a more comprehensive assessment of potential risk by helping to identify any weaknesses in the system that could be exploited by attackers. Typically uses diagramming techniques such as DFD to create a model of the system.</td>
</tr>
<tr>
<td><strong>Failure Mode and Effects Analysis (FMEA)</strong></td>
<td>This type of analysis looks at what might occur if specific components of a system fail or do not function as intended. It allows developers to anticipate possible failures and establish proactive measures to prevent them from occurring.</td>
</tr>
<tr>
<td><strong>Attack Surface Analysis</strong></td>
<td>Analyses the attack surface available to attackers when attempting to compromise a system or organization. Identifies elements of an application or network where an attacker can gain access or leverage an exploit.</td>
</tr>
<tr>
<td><strong>Risk Management Process</strong></td>
<td>Designed to help organizations analyse, prioritize and respond to potential security risks they face in their environment. Typically involves assessing potential risks, determining acceptable levels of risk and then taking steps to mitigate those risks through countermeasures such as implementing security policies or deploying security tools.</td>
</tr>
</tbody>
</table>
Features of Threat Modelling Tools

- **Systems modelling**
  - Typically as a flow diagram

- **Threat intelligence**
  - To inform and prompt
  - e.g. using a scheme like MITRE’s CAPEC

- **Dashboard of vulnerabilities identified**
  - Showing severity

- **Dashboard of mitigations defined**
  - Potentially mapping mitigations top vulnerabilities and threats
  - Guidance for developers

- **Rules engine**
  - To add value by interpreting policies when applied to the system model

- **Scalability and Collaboration**

- **Integration with existing processes and tools**
  - other CI/CD pipeline tools
  - task/issue tracking tools such as Jira
  - Diagramming tools

- **Reporting and Exporting of Information e.g. in PDF and CSV formats**
Some Tools that Can Help

- **Microsoft Threat Modeling Tool** – free
- **OWASP Threat Dragon** – free
- **IriusRisk** – commercial – has a free community tier
- **SD Elements by Security Compass** – commercial
- **Elevation of Privilege (EoP) Security Cards** – Microsoft + OWASP Cornucopia
- **Cairis** – open source
- **Threagile** – open source
- **ThreatModeler** – commercial with a free tier
Summary and Conclusions
1. What are we working on?
- Typically supported using a diagram – such as a DFD
- With a clear boundary to define the scope of the application – decomposing large and complex applications
- Identifies key “assets”

2. What can go wrong?
- Threats that could impact the security or privacy of the application
- List of potential weaknesses in the design or implementation

3. What are we going to do about it?
- Actions to mitigate the impact of the threats identified
- Countermeasures or additional security controls
- Prioritisation of actions.

4. Did we do a good enough job?
- Threats identified?
- Risks reduced – through effective countermeasures
- Lessons learned – e.g. new recommended “standard” security controls for the organisation

https://www.threatmodelingmanifesto.org/
Position of Threat Modelling in the SDLC

• Gartner’s View
  • Build Security Into the Design Phase
    • Translate Security Requirements
    • **Adopt Threat Modelling Practices**
    • Distribute and Promote Secure Coding Practices
    • Automate Governance of Open-Source Software

Microsoft Secure Development Lifecycle (SDL)

• Training / Education on Security Best Practices
• Define Security Requirements – linked to Threat Modelling
• Define Metrics and Compliance Reporting – for security quality e.g. severity thresholds
• **Threat Modelling**
  • Establish standard security features / design requirements
  • Cryptographic standards and requirements
  • Management of third-party components
  • Approved tools
  • Static Analysis Security Testing (SAST)
  • Dynamic Analysis Security Testing (DAST)
  • Penetration Testing
  • Incident Response Plan and Processes

## Approaches to Finding Security Vulnerabilities

*Often left until late in the development lifecycle*

<table>
<thead>
<tr>
<th>Life Cycle Stage</th>
<th>Manual Methods</th>
<th>Automated Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime / Live</td>
<td>Bug-bounty programme&lt;br&gt;Reviewing security bulletins from vendors</td>
<td>Vulnerability scanning&lt;br&gt;Attack surface management</td>
</tr>
<tr>
<td>Deployment</td>
<td>Manual checks</td>
<td>Scanning for vulnerabilities</td>
</tr>
<tr>
<td>Testing</td>
<td>Penetration testing</td>
<td>Dynamic scanning</td>
</tr>
<tr>
<td>Build</td>
<td>Code review/pair programming</td>
<td>Static code analysis</td>
</tr>
<tr>
<td>Design</td>
<td><strong>Threat modelling</strong></td>
<td>Analysis of flow diagrams</td>
</tr>
<tr>
<td>Requirements</td>
<td>Selecting security non-functional requirements</td>
<td>Computer assisted generation of security NFRs (e.g. from profile questionnaire)</td>
</tr>
</tbody>
</table>
Benefits of Threat Modelling

• Helps identify and prioritise threats, early in the lifecycle
  • Helping to optimise resources and limited budgets
  • Reducing risk exposure
• Considers evolving threat landscape
• Helps developers design and build secure software.
• Develops security skills/mindset within project/engineering teams
• Encourages collaboration on security initiatives
BCS DevSecOps Group

https://www.bcs.org/membership-and-registrations/member-communities/devsecops-specialist-group/

Roy Harrow - chairdevsecops@bcs.org
Q + A
Some Useful Talks on YouTube

- IT-SECX 2019 | Keynote - Adam Shostack: Threat Modeling Lessons from Star Wars
  - [https://youtu.be/nd02oPnMdR4](https://youtu.be/nd02oPnMdR4)
- PASTA Threat Modeling for Cybersecurity | OWASP All Chapters 2020 Presentation
  - [https://youtu.be/8k-I3vn8C2A](https://youtu.be/8k-I3vn8C2A)
- Using the Threat Modeling Manifesto to Build an Enterprise Threat Modeling Program, 2022
  - [https://youtu.be/jeHL8PXtezc](https://youtu.be/jeHL8PXtezc)
- Threat Modeling 2.0 - Developer's flavour - Emil Kvarnhammar, DevSecOps London Meeting, 2023
  - [https://www.youtube.com/live/_4gbV7Roc_o](https://www.youtube.com/live/_4gbV7Roc_o)
- Threat Modeling using Microsoft Threat Modeling Tool, 2021
  - [https://youtu.be/Wry2get_RRc](https://youtu.be/Wry2get_RRc)
- Threat modelling with OWASP Threat Dragon, 2022
  - [https://youtu.be/mL5G8HeI8zl](https://youtu.be/mL5G8HeI8zl)