BITS Pilani

Pilani | Dubai | Goa | Hyderabad

SEZG566/SSZG566

Secure Software Engineering

Threat Modelling

T V Rao

DISCLAIMERS

- *The slides presented here are obtained from the authors of the books, product documentations, and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified slides to suit the requirements of the course.*

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Threat Modelling Concepts

Securing a Computer Based System

A computer-based system has three separate but valuable components (Assets) :

- Hardware
- Software
- Data

Vulnerabilities

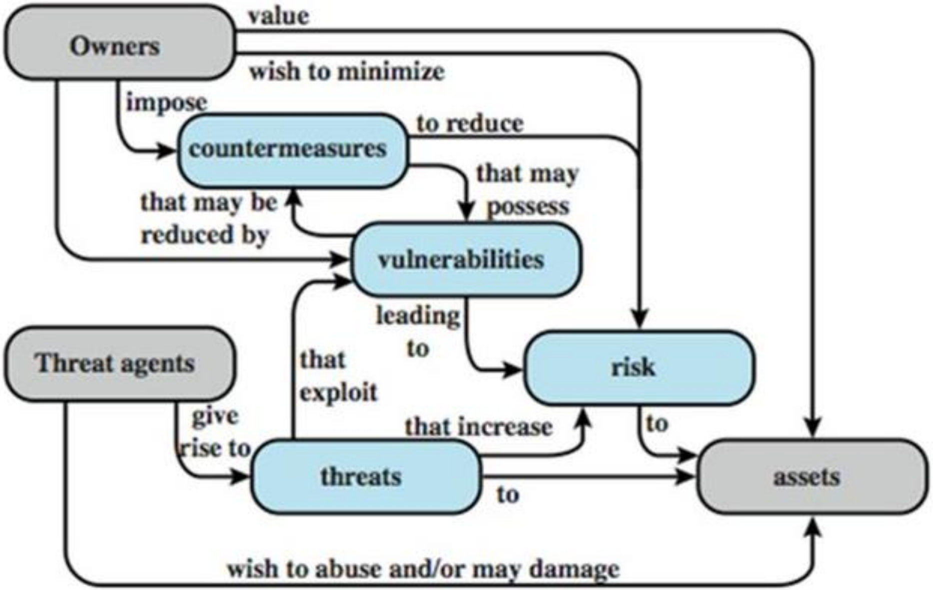
- Weaknesses in a system that may be able to be *exploited* in order to cause loss or harm
    - e.g., a file server that doesn't authenticate its users

Threats

- A loss or harm that might befall a system
    - e.g., users' personal files may be revealed to the public

Security Concepts and Relationships

Review



Characteristics of Computer Intrusion

By computing system, we include

- Hardware
- Software
- Storage media
- Data and
- People

A system is most vulnerable at its weakest point

- A robber will not attempt to penetrate a 2-inch-thick metal door if a window gives easy access

Principle of Easiest Penetration

An intruder must be expected to use any available means of penetration. The penetration may not necessarily be by the most obvious means, nor is it necessarily the one against which the most solid defense has been installed. And it certainly does not have to be the way we want the attacker to behave.

Threat Model

When designing a system, we need to state the threat model

Threat Model

- Set of threats we are undertaking to defend against
- Whom do we want to prevent from doing what?

Attack

- An action which exploits a vulnerability to execute a threat
- e.g., telling the file server you are a different user in an attempt to read or modify their files

Threats to Assets

According to Pfleeger, the threats are

- **Interruption** – an asset is destroyed, unavailable or unusable (*availability*)
- **Interception** – unauthorized party gains access to an asset (*confidentiality*)
- **Modification** – unauthorized party tampers with asset (*integrity*)
- **Fabrication** – unauthorized party inserts counterfeit object into the system (*authenticity*)

Threat Consequences (IETF RFC 4949)

Threat Action (Attack)	Threat Consequence
Exposure: Sensitive data are directly released to an unauthorized entity. **Interception:** An unauthorized entity directly accesses sensitive data traveling between authorized sources and destinations. **Inference:** A threat action whereby an unauthorized entity indirectly accesses sensitive data (but not necessarily the data contained in the communication) by reasoning from characteristics or by-products of communications. **Intrusion:** An unauthorized entity gains access to sensitive data by circumventing a system's security protections.	**Unauthorized Disclosure** A circumstance or event whereby an entity gains access to data for which the entity is not authorized
Masquerade: An unauthorized entity gains access to a system or performs a malicious act by posing as an authorized entity. **Falsification:** False data deceive an authorized entity. **Repudiation:** An entity deceives another by falsely denying responsibility for an act	**Deception** A circumstance or event that may result in an authorized entity receiving false data and believing it to be true
Incapacitation: Prevents or interrupts system operation by disabling a system component. **Corruption:** Undesirably alters system operation by adversely modifying system functions or data. **Obstruction:** A threat action that interrupts delivery of system services by hindering system operation	**Disruption** A circumstance or event that interrupts or prevents the correct operation of system services and functions
Misappropriation: An entity assumes unauthorized logical or physical control of a system resource. **Misuse:** Causes a system component to perform a function or service that is detrimental to system security	**Usurpation** A circumstance or event that results in control of system services or functions by an unauthorized entity

Who are Attackers

One approach to prevention is to understand who carries out attacks and why

Amateurs

- Ordinary computer professionals or users who, while doing their jobs, discover they have access to something valuable. Amateurs may be disgruntled employees who vow to get even with management

Crackers or Malicious Hackers

- Attempt to access computing facilities for which they have not been authorized (often students who see it as victimless crime). Some carry out for curiosity, personal gain or self-satisfaction

Who are Attackers

Career Criminals

- Understands the targets of computer crime; often begin as computer professionals, then shift to crime finding payoff. “They don’t want to write a worm that destroys your hardware. They want to assimilate your computers and use them to make money”

Terrorists

- They use computers in three ways
  - Targets of attack – denial of service, web site defacement attacks are popular to attract attention to the cause and bring undesired negative attention to the targets of attack
  - Propaganda vehicles – inexpensive way to get a message to many
  - Methods of attack – use computers to launch attacks

Method, Opportunity, Motive (M-O-M triad)

A malicious attacker must have three things

- Method : the skills, knowledge, tools, and other things with which to be able to pull off the attack
- Opportunity : the time and access to accomplish the attack
- Motive : a reason to want to perform this attack against this system

Deny any of those three things and the attack will not occur.

Methods of defense

How can we defend against a threat?

- Prevent it: prevent the attack
- Deter it: make the attack harder or more expensive
- Deflect it: make yourself less attractive to attacker
- Detect it: notice that attack is occurring (or has occurred)
- Recover from it: mitigate the effects of the attack

Methods of defense

Threat: your car may get stolen

How to defend?

- Prevent: Immobilizer? Is it possible to absolutely prevent?
- Deter: Store your car in a secure parking facility
- Deflect: Have sticker mentioning car alarm, keep valuables out of sight
- Detect: Car alarms
- Recover: Insurance

Structured Approach to Threat Modeling

According to Adam Shostack, you begin threat modeling by focusing on four key questions

- What are you building?
- What can go wrong?
- What should you do about those things that can go wrong?
- Did you do a decent job of analysis (retrospect)

Structured Approach to Threat Modeling

People often use an approach centered on

- Models of their assets (Valuable things they have),
- Models of attackers (People who might go after assets), or
- Models of their software (Common way to attack is via the deployed software)

Centering on one of these is preferable to using approaches that combine them because the combinations tend to be confusing

According to Adam Shostack, first two sets of models help in engaging with non-technical people and third type of models are important for software development

What Threat Modelling is (not)

What threat modelling is	What threat modelling is not
A team activity	An activity performed by a single team member in isolation
An activity that helps identify security vulnerabilities in a variety of software applications	Just for large software projects
An activity that should be performed for every iteration or sprint during agile development	An activity done once during the lifecycle of the project

www.microsoft.com

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

OWASP Threat Modelling Process

OWASP Threat Modeling

According to OWASP, the threat modeling process for software application can be decomposed into 3 high level steps

- Decompose the Application
- Determine and rank threats
- Determine countermeasures and mitigation

Decompose the Application (step 1)

Understanding of the application and how it interacts with external entities.

- involves creating use-cases to understand how the application is used,
- identifying entry points to see where a potential attacker could interact with the application,
- identifying assets i.e. items/areas that the attacker would be interested in, and
- identifying trust levels which represent the access rights that the application will grant to external entities.

Produce data flow diagrams (DFDs) for the application. The DFDs show the different paths through the system, highlighting the privilege boundaries.

Determine and rank threats (step 2)

A threat categorization such as STRIDE can be used,

Spoofing

Tampering

Repudiation

Information disclosure

Denial of service

Elevation of privilege

The STRIDE categorization helps to identify threats from the attacker perspective.

Determine and rank threats (step 2)

A threat categorization ASF (Application Security Framework) defines threat categories such as

Auditing & Logging,

Authentication,

Authorization,

Configuration Management,

Data Protection in Storage and Transit,

Data Validation,

Exception Management.

The ASF categorization helps to identify threats from the defensive perspective.

Determine and rank threats (step 2)

DFDs produced in step 1 help to identify the potential threat targets from the attacker's perspective, viz.

data sources,

processes,

data flows, and

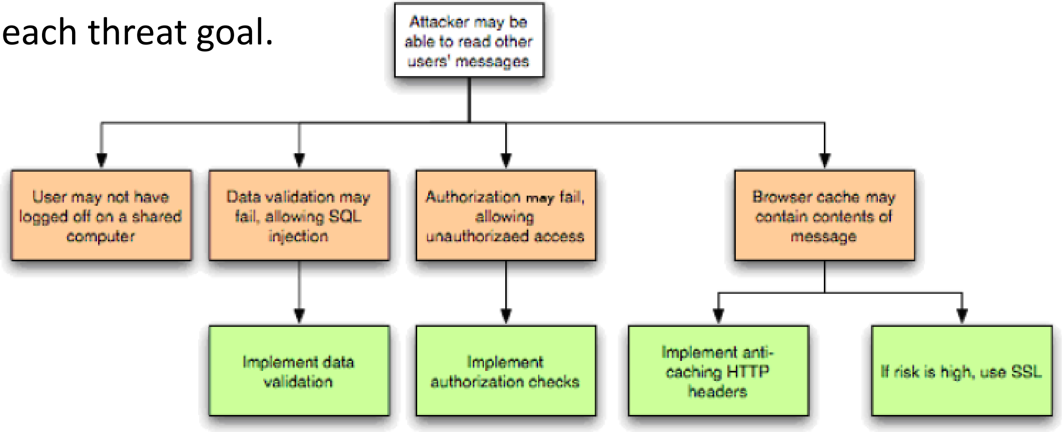
interactions with users.

Use and abuse cases can illustrate how existing protective measures could be bypassed, or where a lack of such protection exists.

Determine and rank threats (step 2)

These threats can be identified further as the roots for threat trees;

- one tree for each threat goal.



Attacker may be able to read other users' messages
User may not have logged off on a shared computer
Data validation may fail, allowing SQL injection
Authorization may fail, allowing unauthorizaed access
Browser cache may contain contents of message
Implement data validation
Implement authorization checks
Implement anti-caching HTTP headers
If risk is high, use SSL

From the defensive perspective, ASF categorization helps to identify the threats as weaknesses of security controls for such threats.

The determination of the security risk for each threat can be determined using a value-based risk model such as DREAD

Determine countermeasures and mitigation

A lack of protection against a threat might indicate a vulnerability whose risk exposure could be mitigated with the implementation of a countermeasure.

Countermeasures can be identified using threat-countermeasure mapping lists.

Based on risk ranking assigned to the threats, it is possible to sort threats from the highest to the lowest risk, and prioritize the mitigation effort, such as by responding to such threats by applying the identified countermeasures

OWASP Threat Modeling Example

Application Description:

The college library website is the first implementation of a website to provide librarians and library patrons (students and college staff) with online services. As this is the first implementation of the website, the functionality will be limited. There will be three users of the application:

1. Students
2. Staff
3. Librarians

Staff and students will be able to log in and search for books, and staff members can request books. Librarians will be able to log in, add books, add users, and search for books.

External Dependencies

External dependencies are items external to the code of the application that may pose a threat to the application. These items are typically still within the control of the organization, but possibly not within the control of the development team

ID	Description
1	The database server will be MySQL and it will run on a Linux server. This server will be hardened as per the college's server hardening standard. This will include the application of the latest operating system and application security patches.
2	The connection between the Web Server and the database server will be over a private network.

Trust Levels

Trust levels represent the access rights that the application will grant to external entities.

The trust levels are cross referenced with the entry points and assets.

This allows us to define the access rights or privileges required at each entry point, and those required to interact with each asset

Trust Levels

ID	Name	Description
1	Anonymous Web User	A user who has connected to the college library website but has not provided valid credentials.
2	User with Valid Login Credentials	A user who has connected to the college library website and has logged in using valid login credentials.
3	User with Invalid Login Credentials	A user who has connected to the college library website and is attempting to log in using invalid login credentials.
4	Librarian	The librarian can create users on the library website and view their personal information.
5	Database Server Administrator	The database server administrator has read and write access to the database that is used by the college library website.
6	Website Administrator	The Website administrator can configure the college library website.
7	Web Server User Process	This is the process/user that the web server executes code as and authenticates itself against the database server as.
8	Database Read User	The database user account used to access the database for read access.
9	Database Read/Write User	The database user account used to access the database for read and write access.

Entry Points

Entry points define the interfaces through which potential attackers can interact with the application or supply it with data. In order for a potential attacker to attack an application, entry points must exist.

ID	Name	Description	Trust Levels
1.1	Library Main Page	The splash page for the college library website is the entry point for all users.	(1) Anonymous Web User (2) User with Valid Login Credentials (3) User with Invalid Login Credentials (4) Librarian
1.3	Search Entry Page	The page used to enter a search query.	(2) User with Valid Login Credentials (4) Librarian

Assets

Attacker is interested in the system because it has Assets

Assets can be

- Physical – Private Data, List of customers etc.
- Privilege – System has ability to update/process data
- Abstract – Reputation of the organization

Assets are documented in the threat model as follows:

- ID,
- Name,
- Description
- Trust Level (required for access)

Assets

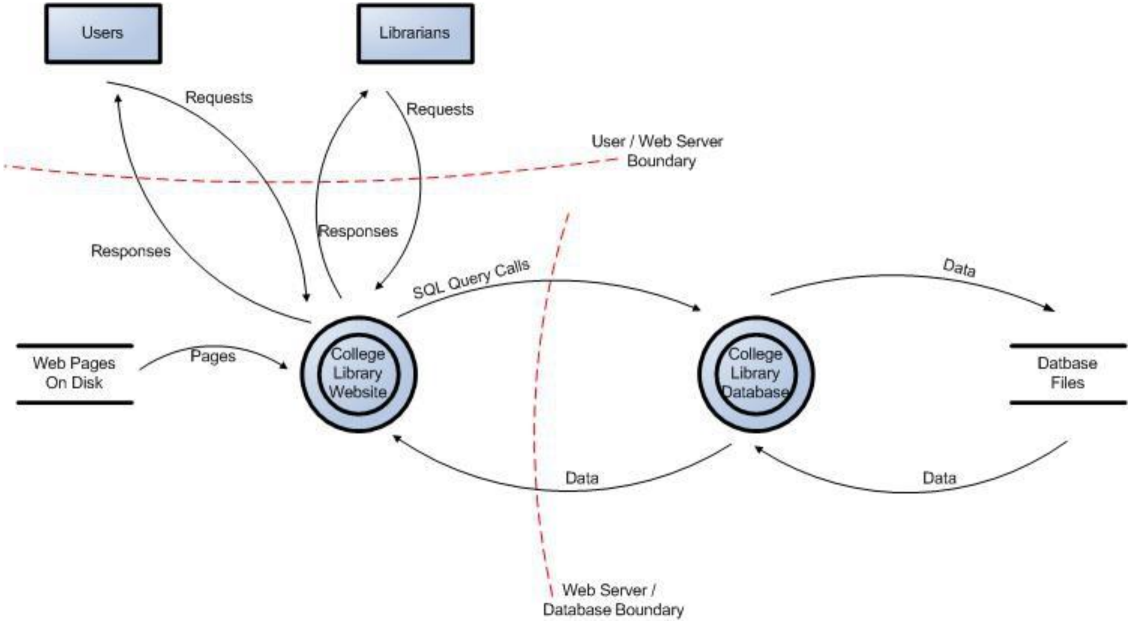
ID	Name	Description	Trust Levels
1	Library Users and Librarian	Assets relating to students, faculty members, and librarians.	
1.1	User Login Details	The login credentials that a student or a faculty member will use to log into the College Library website.	(2) User with Valid Login Credentials, (4) Librarian, (5) Database Server Administrator, (7) Web Server User Process, (8) Database Read User, (9) Database Read/Write User
1.3	Personal Data	The College Library website will store personal information relating to the students, faculty members, and librarians.	(4) Librarian, (5) Database Server Administrator, (6) Website Administrator, (7) Web Server User Process, (8) Database Read User, (9) Database Read/Write User
2	System	Assets relating to the underlying system.	
2.2	Ability to Execute Code as a Web Server User	This is the ability to execute source code on the web server as a web server user.	(6) Website Administrator, (7) Web Server User Process
2.4	Ability to Execute SQL as a Database Read/Write User	This is the ability to execute SQL. Select, insert, and update queries on the database and thus have read and write access to any information stored within the College Library database.	(5) Database Server Administrator, (9) Database Read/Write User
3	Website	Assets relating to the College Library website.	
3.1	Login Session	This is the login session of a user to the College Library website. This user could be a student, a member of the college faculty, or a Librarian.	(2) User with Valid Login Credentials, (4) Librarian
3.3	Ability to Create Users	The ability to create users would allow an individual to create new users on the system. These could be student users, faculty member users, and librarian users.	(4) Librarian, (6) Website Administrator

Data Flow Diagrams

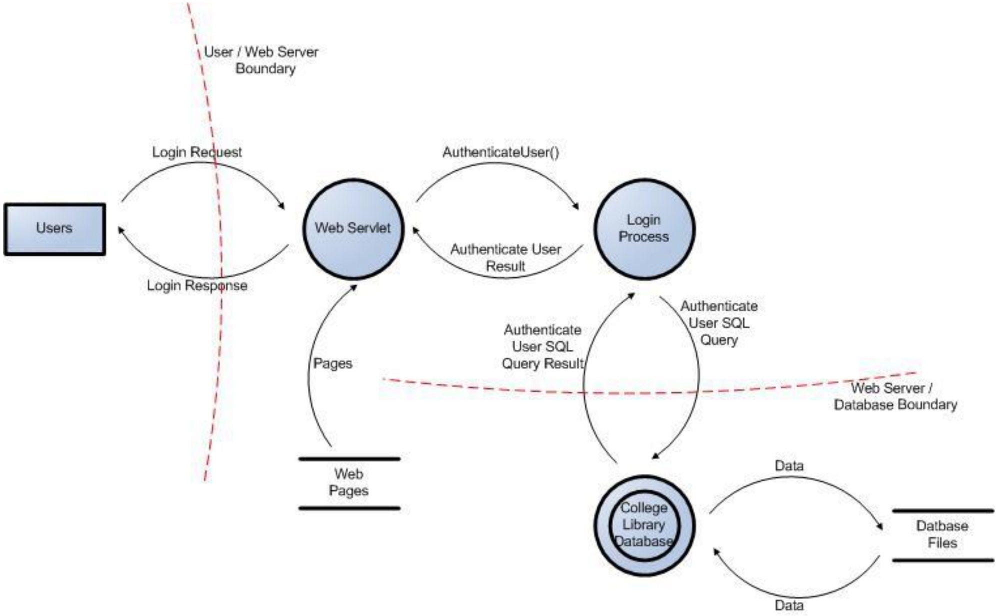
Knowledge of Assets, Entry points, etc. help in creating DFDs.

- The DFDs will allow us to gain a better understanding of the application by providing a visual representation of how the application processes data
- DFDs focus on how data moves through the application and what happens to the data as it moves
- DFDs are hierarchical in structure, so they can be used to decompose the application into subsystems and lower-level subsystems

DFD for the example



Partially Expanded DFD



Trust Boundaries in DFD

Add trust boundaries that intersect data flows

- Points/surfaces where an attacker can interject
  - Machine boundaries, privilege boundaries, integrity boundaries are examples of trust boundaries
  - Threads in a native process are often inside a trust boundary, because they share the same privileges, rights, identifiers and access
- Processes talking across a network always have a trust boundary
  - They may create a secure channel, but they’re still distinct entities
  - Encrypting network traffic doesn’t address tampering or spoofing

Iterate over processes, data stores, and see where they need to be broken down

Threat Perspectives

Threat categorization helps to identify threats.

- Threat categorization to help identify threats from the attacker
  - (STRIDE)

- Threat categorization to help identify threats from the defensive perspective
  - Application Security Framework(ASF)

Example Countermeasures (ASF)

Threat Type	Countermeasure
Authentication	1.Credentials and authentication tokens are protected with encryption in storage and transit 2.Protocols are resistant to brute force, dictionary, and replay attacks
Authorization	1.Strong ACLs are used for enforcing authorized access to resources 2.Role-based access controls are used to restrict access to specific operations
Configuration Management	1.Least privileged processes are used and service accounts with no administration capability 2.Auditing and logging of all administration activities is enabled
Data Protection in Storage and Transit	1.Standard encryption algorithms and correct key sizes are being used 2.Hashed message authentication codes (HMACs) are used to protect data integrity
Data Validation / Parameter Validation	1.Data type, format, length, and range checks are enforced 2.No security decision is based upon parameters (e.g. URL parameters) that can be manipulated
Error Handling and Exception Management	1.All exceptions are handled in a structured manner 2.Error messages are scrubbed so that no sensitive information is revealed to the attacker
User and Session Management	1.No sensitive information is stored in clear text in the cookie 2.Cookies are configured to expire
Auditing and Logging	1.Sensitive information (e.g. passwords, PII) is not logged 2.Integrity controls (e.g. signatures) are enforced on log files to provide non-repudiation

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

SDL Threat Modeling

Objectives

Produce software that's secure by design

- Improve designs the same way we've improved code

Because attackers think differently

- Creator blindness/new perspective

Allow you to predictably and effectively find security problems early in the process

Responsibilities

Building a threat model (at Microsoft)

- –Program Manager (PM) owns overall process
- –Testers
  - •Identify threats in analyze phase
  - •Use threat models to drive test plans
- –Developers create diagrams

Customers / Work Products

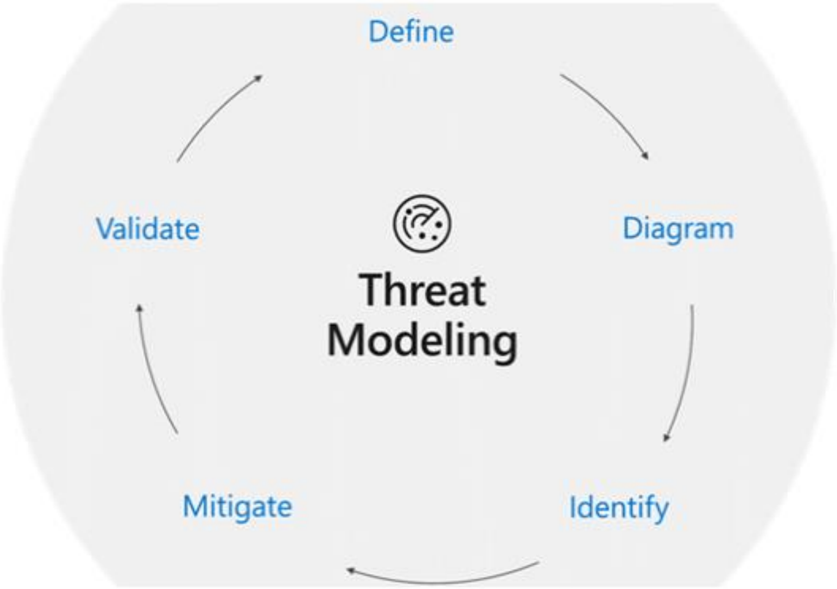
Customers for threat models

- Your team
- Other features, product teams
- Customers, via user education
- “External” quality assurance resources, such as pen testers

Threat model documentation

- The product as a whole
- The security-relevant features
- The attack surfaces

The Process in a Nutshell



- Defining security requirements.
- Creating an application diagram.
- Identifying threats.
- Mitigating threats.
- Validating that threats have been mitigated.

Define Security Requirements

Security requirements may come from

- industry standards,
- applicable laws, and
- history of past vulnerabilities

Security requirements provide a foundation of vetted security functionality for an application

Allow developers to reuse the definition of security controls and best practices

Prevent the repeat of past security failures

Diagramming

Use DFDs (Data Flow Diagrams)

- Include processes, data stores, data flows
- Include *trust boundaries*
- Diagrams per scenario may be helpful

Update diagrams as product changes

Enumerate assumptions, dependencies

Number everything (if manual)

Effective Threat Modeling Meetings

Develop draft threat model before the meeting

- Use the meeting to discuss

Start with a DFD walkthrough

Identify most interesting elements

- Assets (if you identify any)
- Entry points/trust boundaries

Walk through STRIDE against those elements

Threats that cross elements/recur

- Consider library, redesigns

Validating Threat Models

Validate the whole threat model

- Does diagram match final code?
- Are threats enumerated?
- Minimum: STRIDE per element that touches a trust boundary
- Has Test / QA reviewed the model?
  - Tester approach often finds issues with threat model or details
- Is each threat mitigated?
- Are mitigations done right?

Did you check these before Final Security Review?

- Shipping will be more predictable

Threats (STRIDE)

Type	Examples
Spoofing	Threat action aimed to illegally access and use another user's credentials, such as username and password.
Tampering	Threat action aimed to maliciously change/modify persistent data, such as persistent data in a database, and the alteration of data in transit between two computers over an open network, such as the Internet.
Repudiation	Threat action aimed to perform illegal operations in a system that lacks the ability to trace the prohibited operations.
Information disclosure	Threat action to read a file that one was not granted access to, or to read data in transit.
Denial of service	Threat aimed to deny access to valid users, such as by making a web server temporarily unavailable or unusable.
Elevation of privilege	Threat aimed to gain privileged access to resources for gaining unauthorized access to information or to compromise a system.

Threat: Spoofing

Threat	**S**poofing
Property	Authentication
Definition	Impersonating something or someone else
Example	Pretending to be any of billg, microsoft.com, or ntdll.dll

Threat: Tampering

Threat	**T**ampering
Property	Integrity
Definition	Modifying data or code
Example	Modifying a DLL on disk or DVD, or a packet as it traverses the LAN

Threat: Repudiation

Threat	**R**epudiation
Property	Non-Repudiation
Definition	Claiming to have not performed an action
Example	“I didn’t send that email,” “I didn’t modify that file,” “I didn’t visit that web site”

Threat: Information Disclosure

Threat	Information Disclosure
Property	Confidentiality
Definition	Exposing information to someone not authorized to see it
Example	Allowing someone to read the Windows source code; publishing a list of customers to a Web site

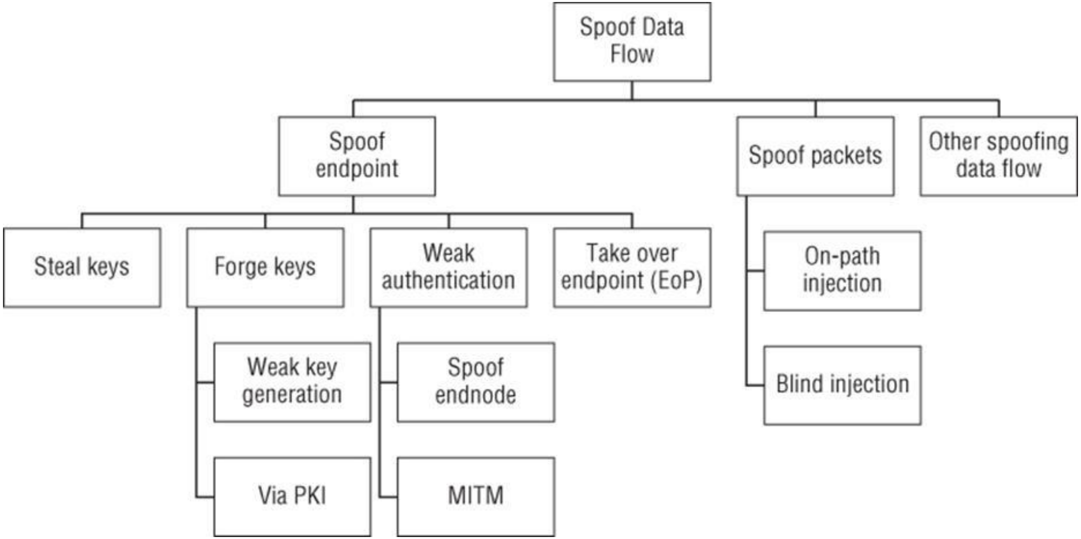
Threat: Denial of Service

Threat	**D**enial of Service
Property	Availability
Definition	Deny or degrade service to users
Example	Crashing Windows or a Web site, sending a packet and absorbing seconds of CPU time, or routing packets into a black hole

Threat: Elevation of Privilege

Threat	**E**levation of Privilege (EoP)
Property	Authorization
Definition	Gain capabilities without proper authorization
Example	Allowing a remote Internet user to run commands is the classic example, but going from a “Limited User” to “Admin” is also EoP

Threat Tree (Spoofing data flow)



How to Mitigate

Address each threat

Four ways to address threats

1. Redesign to eliminate
2. Apply standard mitigations
   - What have similar software packages done and how has that worked out for them?
3. Invent new mitigations (riskier)
4. Accept vulnerability in design

Standard Mitigations

Spoofing	Authentication	To authenticate principals: • Cookie authentication • Kerberos authentication To authenticate code or data: • Digital signatures
Tampering	Integrity	•ACLs • Digital signatures
Repudiation	Non Repudiation	• Secure logging and auditing • Digital Signatures
Information Disclosure	Confidentiality	• Encryption • ACLS
Denial of Service	Availability	• ACLs • Filtering • Quotas
Elevation of Privilege	Authorization	• ACLs • Group or role membership • Privilege ownership • Input validation

DREAD

In the Microsoft DREAD threat-risk ranking model, the technical risk factors for impact are Damage and Affected Users, while the ease of exploitation factors are Reproducibility, Exploitability and Discoverability. This risk factorization allows the assignment of values to the different influencing factors of a threat. To determine the ranking of a threat, the threat analyst has to answer basic questions for each factor of risk

- For Damage: How big would the damage be if the attack succeeded?
- For Reproducibility: How easy is it to reproduce an attack to work?
- For Exploitability: How much time, effort, and expertise is needed to exploit the threat?
- For Affected Users: If a threat were exploited, what percentage of users would be affected?
- For Discoverability: How easy is it for an attacker to discover this threat?

DREAD Example

The college library website use case:

Threat: Malicious user views confidential information of students, faculty members and librarians.

- **Damage potential:** Threat to reputation as well as financial and legal liability:8
- **Reproducibility:** Fully reproducible:10
- **Exploitability:** Require to be on the same subnet or have compromised a router:7
- **Affected users:** Affects all users:10
- **Discoverability:** Can be found out easily:10

Overall DREAD score: (8+10+7+10+10) / 5 = 9

In this case having 9 on a 10 point scale is certainly a high risk threat

Cyber security and resilience for Smart Hospitals by ENISA

https://www.enisa.europa.eu/publications/cyber-security-and-resilience-for-smart-hospitals

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Privacy Threat Modelling
LINDDUN

LINDDUN

LINDDUN was created at KU Research University, Belgium

It is Systematic elicitation and mitigation of privacy. LINDDUN Stands for

- Linkability,
- Identifiability,
- Non-Repudiation,
- Detectability,
- Disclosure of Information,
- Unawareness,
- Non-Compliance

LINDDUN

- Linkability - Being able to sufficiently distinguish whether 2 IOI (items of interest) are linked or not, even without knowing the actual identity of the subject of the linkable IOI
- Identifiability - Being able to sufficiently identify the subject within a set of subjects (i.e. the anonymity set) Or Not being able to hide the link between the identity and the IOI (an action or piece of information).
- Non-Repudiation – works differently in the context of privacy
- Detectability - An attacker can sufficiently distinguish whether an item of interest (IOI) exists or not
- Disclosure of Information - Exposing information to someone not authorized to see it.
- Unawareness - Not understanding the consequences of sharing personal information in the past, present, or future.
- Non-Compliance - Not following the (data protection) legislation, the advertised policies or the existing user consents

Non-Repudiation in Privacy context

- In the context of privacy, non-repudiation works differently from a financial transaction perspective.

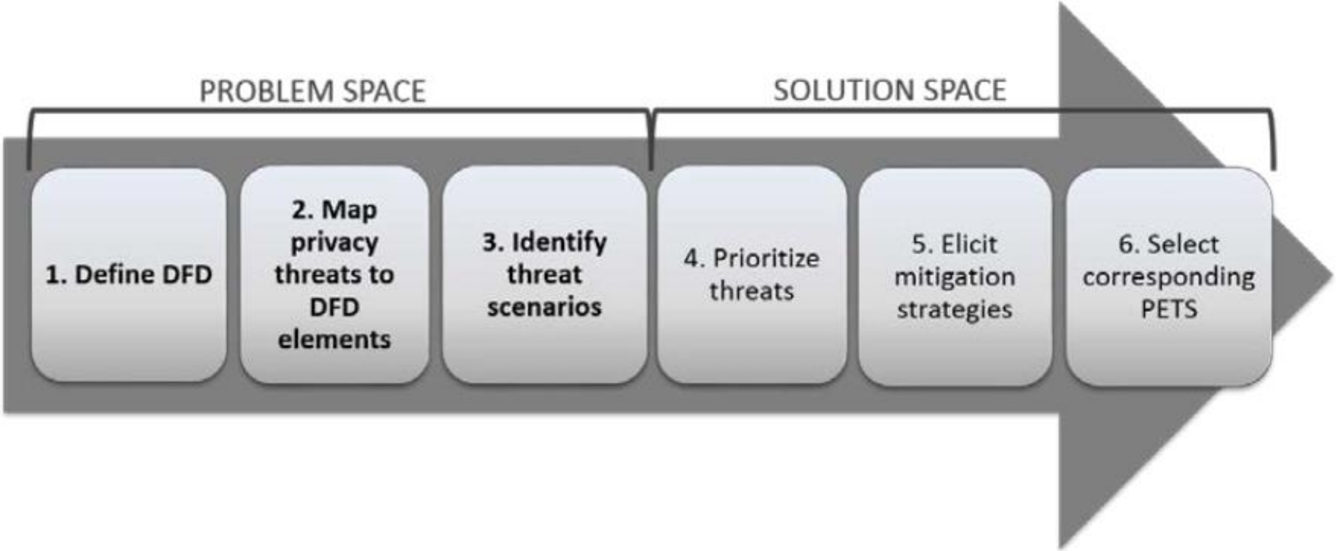
- Privacy context expects that attacker can not prove
  - Typical non-repudiation examples exist e.g.
    - Anonymous online voting systems, and
    - Whistleblowing systems

    where plausible deniability is required

- Non-repudiation is actually a security goal, for many systems (e.g. systems where payments are involved)

LINDDUN

LINDDUN provides a systematic approach to privacy assessment, consisting of 6 steps


PROBLEM SPACE
SOLUTION SPACE
1. Define DFD
2. Map privacy threats to DFD elements
3. Identify threat scenarios
4. Prioritize threats
5. Elicit mitigation strategies
6. Select corresponding PETS

PET – privacy enhancing technology

More Methods of Threat Modeling

https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=524448 https://www.enisa.europa.eu/publications/cyber-security-and-resilience-for-smart-hospitals

Threat Modelling by Adam Shostack, John Wiley 2014

Security in Computing by Charles P. Pfleeger, Shari L. Pfleeger, and Deven Shah Pearson Education 2009

Computer Security: Principles and Practice by William Stallings, and Lawrie Brown Pearson, 2018.

www.owasp.com

www.microsoft.com

Thank You!