

CYBR3000

Introduction to Threat modeling

Dr Dan Kim

Associate Professor in Cybersecurity,

School of EECS

University of Queensland, Australia

Homepage: <https://researchers.uq.edu.au/researcher/23703>

Learning objectives

- At the end of this lecture, you will be able to understand/explain
 - Why Threat modeling?
 - Secure Software Design
 - How to do threat modeling
 - STRIDE threat modeling

Why Threat modeling?

- Almost all software systems today face a variety of threats, and the number of threats grows as technology changes.
- Threats can come from outside or within organizations, and they can have devastating consequences.
- To prevent threats from taking advantage of system flaws, administrators can use threat-modeling methods to inform defensive measures.

Threat, threat model, & modeling

- A threat
 - is a **potential** or actual undesirable event that may be malicious (such as DoS attack) or incidental (failure of a Storage Device).
- A threat model
 - is essentially a **structured representation** of all the information that affects the security of an application.
 - is a view of the application and its environment through security glasses.
- Threat modeling
 - is a **process** for capturing, organizing and analyzing all of this information.
 - enables informed decision-making about application security risk.
 - produces a prioritized list of security improvements to the concept, requirements, design, or implementation.

Threat Modeling Across the Lifecycle

- Threat modeling is best applied continuously throughout a software development project.
- Ideally, a high-level threat model should be defined in the concept or planning phase, and then refined throughout the lifecycle.
- As more details are added to the system, new attack vectors are created and exposed.
- The ongoing threat modeling process should examine, diagnose, and address these threats.

Threat modeling? How?

- Threat-modeling methods are used to create
 - an abstraction of the system
 - profiles of potential attackers, including their goals and methods
 - a catalog of potential threats that may arise

Threat modelling: three main approaches

- **Attacker**-centric approach

- Intel's TARA (Threat Agent Risk Assessment)
- **LM's Cyber Kill Chain (CKC)**
- **MITRE's ATT&CK**
- Cyber OODA (Observe, Orient, Decide, Act)

- **Asset**-centric approach

- PASTA (The Process for Attack Simulation and Threat Analysis)
- OCTAVE (The Operationally Critical Threat, Asset, and Vulnerability Evaluation)
- ETSI's TVRA (Threat Vulnerability and Risk Analysis)

- **Software**-centric approach

- DREAD
- STRIDE

Software-centric approach: DREAD

- DREAD (Damage, Reproducibility, Exploitability, Affected users, Discoverability)
 - **Damage potential:** How great is the damage if the vulnerability is exploited?
 - **Reproducibility:** How easy is it to reproduce the attack?
 - **Exploitability:** How easy is it to launch an attack?
 - **Affected users:** As a rough percentage, how many users are affected?
 - **Discoverability:** How easy is it to find the vulnerability?

Software-centric approach: DREAD - example

- Damage potential:
- The organization measures the amount of damage that a threat actor can cause as the damage potential on the following scale:
 - 0 – Indicates no damage caused to the organization
 - 5 – Information disclosure said to have occurred
 - 8 – Non-sensitive user data has been compromised
 - 9 – Non-sensitive administrative data has been compromised
 - 10 – The entire information system has been destroyed. All data and applications are inaccessible

Software-centric approach: DREAD - example

- Reproducibility:
- Reproducibility indicates if it's simple to replicate an attack. These are again plotted on a scale of 0 – 10:
 - 0 – Difficult or impossible to replicate the attack
 - 5 – Complex to replicate the attack
 - 7.5 – Easy to replicate the attack
 - 10 – Very easy to replicate the attack

Software-centric approach: DREAD - example

- Exploitability:
- Different vulnerabilities in an organization can be exploited by using different tools and skills, as indicated by their ratings. They are rated as follows:
 - 2.5 – Indicates that advanced programming and networking skills needed to exploit the vulnerability
 - 5 – Available attack tools needed to exploit the vulnerability
 - 9 – Web application proxies are needed to exploit the vulnerability
 - 10 – Indicates the requirement of a web browser needed to exploit the vulnerability

Software-centric approach: DREAD - example

- Affected Users:
- Calculate the number of users who will be affected by an attack to determine the potential impact of the attack. This is again rated on a scale of 1 – 10.
 - 0 – no users affected
 - 2.5 – Indicates chances of fewer individual users affected
 - 6 – Few users affected
 - 8 – Administrative users affected
 - 10 – All users affected

Software-centric approach: DREAD - example

- Discoverability:
- On a scale of 1 – 10, this factor rates the discoverability of a vulnerability.
 - 0 – Indicates it's hard to discover the vulnerability
 - 5 – HTTP requests can uncover the vulnerability
 - 8 – Vulnerability found in the public domain
 - 10 – Vulnerability found in web address bar or form

An Example of DREAD

- **Attackers may be able to social engineer the helpdesk in order to gain access to organization user accounts. Some of these accounts may be administrative accounts.**

DREAD Criterion	Score	Comments
Damage	10	Gaining access to user accounts, or administrative user accounts would allow an attacker to perform further attacks, including destruction of applications and systems.
Reproducibility	5	Social engineering requires skill, and the helpdesk employees have been trained to look out for social engineering attacks. There are also procedures in place regarding user account actions (like resetting accounts, etc.).
Exploitability	5	Exploiting the accounts would still require access to the organization network.
Affected users	8	Administrative accounts may be targeted.
Discoverability	8	The helpdesk is widely known within the organization, and anyone (inside or outside) can reach the helpdesk.

An Example of DREAD

- Continuing with the example discussed above, the following table shows how a designer might assess the hypothetical denial-of-service attack:

DREAD Criterion	Score	Comments
Damage	8	Disrupts work temporarily but causes no permanent damage or data loss.
Reproducibility	10	Causes the device to fail every time.
Exploitability	7	Requires a focused effort to determine the command sequence.
Affected users	10	Affects every model of this device on the market.
Discoverability	10	Assumes that every potential threat will be discovered.
Total:	9.0	Mitigating this problem is high priority.

Software-centric approach - STRIDE

▪ **STRIDE** and Associated Derivations

- **S**poofing identity,
- **T**ampering with data,
- **R**epudiation threats,
- **I**nformation disclosure,
- **D**enial of service and
- **E**levation of privileges

STRIDE Threat categories

Threat	Property violated	Threat definition
Spoofing	Authentication	Pretending to be something or someone other than yourself
Tampering	Integrity	Modifying something on disk, network, memory, or elsewhere
Repudiation	Nonrepudiation	Claiming that you didn't do something or were not responsible; can be honest or false
Information Disclosure	Confidentiality	Providing information to someone not authorized to access it
Denial of Service	Availability	Exhausting resources needed to provide service
Elevation of Privilege	Authorization	Allowing someone to do something they are not authorized to do

Standard Mitigations

- public key infrastructure (PKI)
- Secure Sockets Layer (SSL)/ Transport Layer Security (TLS)
- Access control lists (ACLs)

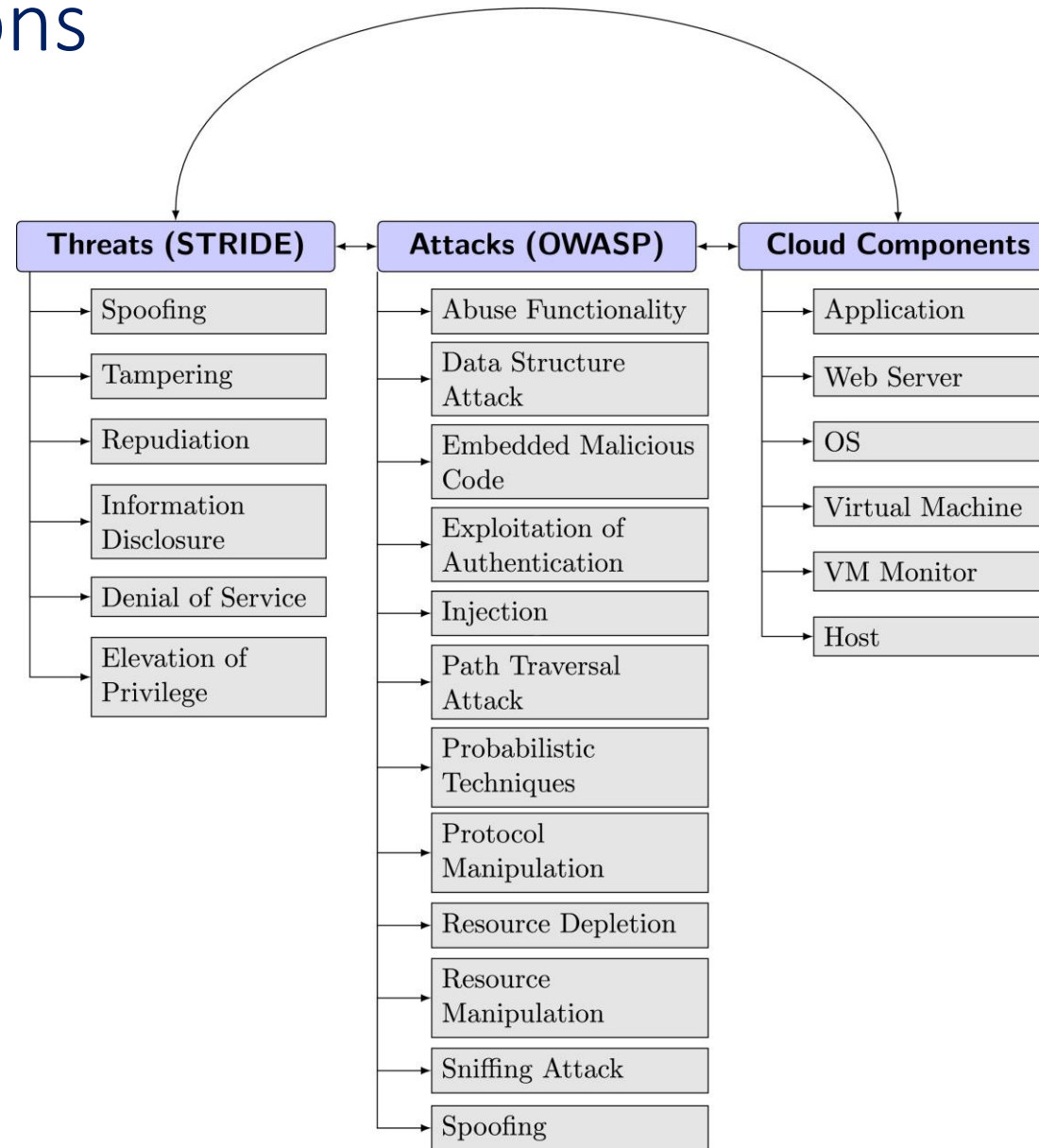
Spoofing	Authentication	<p>To authenticate principals:</p> <ul style="list-style-type: none"> • Cookie authentication • Kerberos authentication • PKI systems such as SSL/TLS and certificates <p>To authenticate code or data:</p> <ul style="list-style-type: none"> • Digital signatures
Tampering	Integrity	<ul style="list-style-type: none"> • Windows Vista Mandatory Integrity Controls • ACLs • Digital signatures
Repudiation	Non Repudiation	<ul style="list-style-type: none"> • Secure logging and auditing • Digital Signatures
Information Disclosure	Confidentiality	<ul style="list-style-type: none"> • Encryption • ACLs
Denial of Service	Availability	<ul style="list-style-type: none"> • ACLs • Filtering • Quotas
Elevation of Privilege	Authorization	<ul style="list-style-type: none"> • ACLs • Group or role membership • Privilege ownership • Input validation

An example

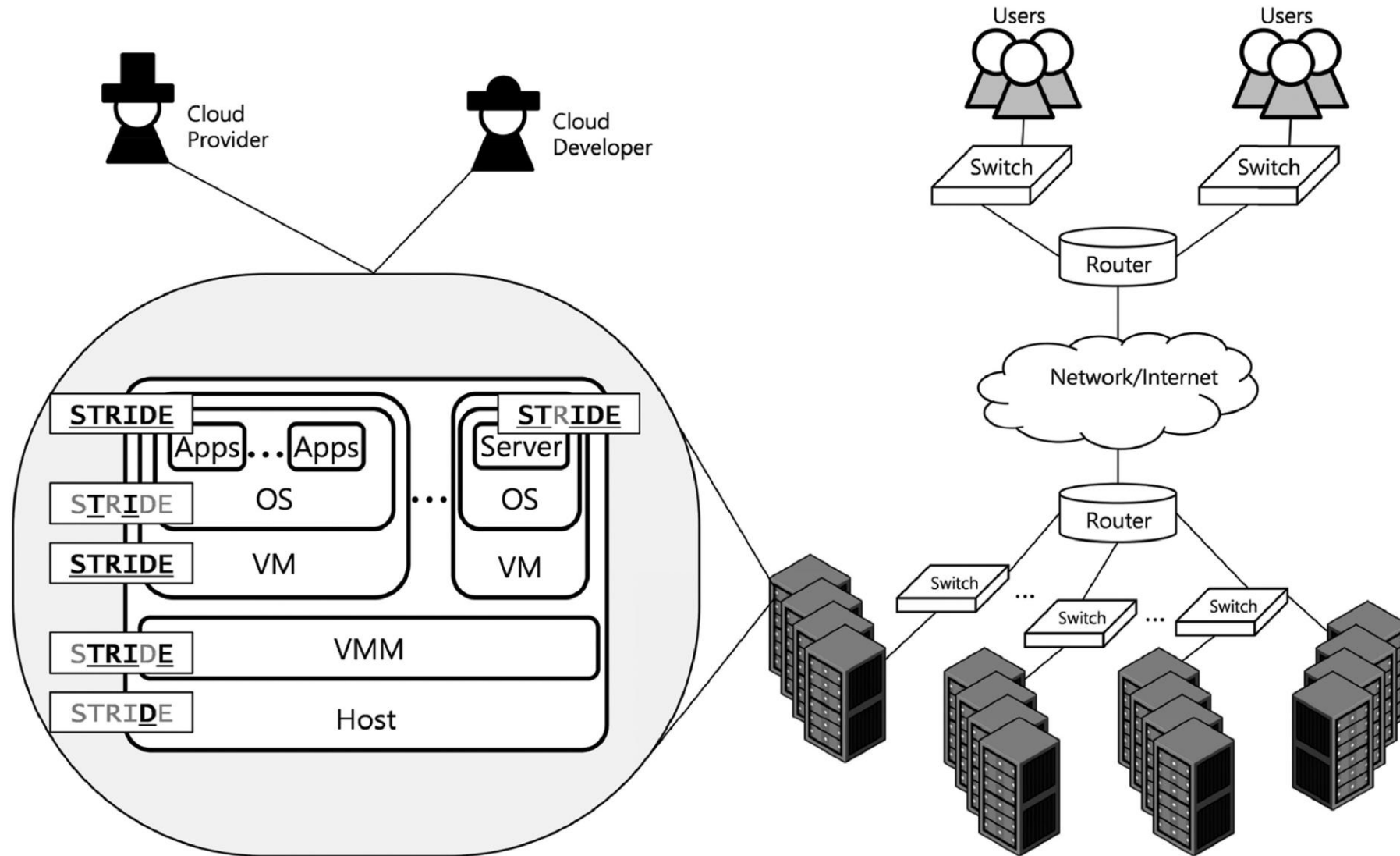
- Cloud computing

- Jin B. Hong, Armstrong Nhlabatsi, **Dong Seong Kim**, Alaa Hussein, Noora Fetais, Khaled M. Khan: Systematic identification of threats in the cloud: A survey. **Comput. Networks** 150: 46-69 (2019)

Mapping the threat, attack and cloud component classifications



Cloud components and STRIDE mapping



STRIDE vs cloud components (partial view).

	Cloud components					
STRIDE category	Application	Web server	OS	Virtual machine	VM Monitor	Host
Spoofing	Hidden parameter manipulation, install backdoor [50] , [69] , spoof metadata [32] , phishing [51] , cross-site request forgery [97] , IP spoofing [104] , malicious file [63] , [73] , buffer overflow [17]	Hidden parameter manipulation, install backdoor [50] , [69] , cross-site request forgery [97] , SOAP message variable manipulation [39] , [72] , IP spoofing [104] , malicious file [63] , [73] , buffer overflow [17]	Alter hidden parameter, install backdoor [50] , [69] , manipulate SOAP message variable [39] , [72] , IP spoofing [104]	ARP spoofing [116] , DNS spoofing [119] , IP spoofing [104] , malicious VM [105]	Malicious VM [105]	ARP spoofing [116] , DNS spoofing [119] , IP spoofing [104]
Tampering	Manipulate RIA components [86] , direct object reference manipulation, XSS [16] , [121] , LDAP [68] , [96] , SQL, Javascript [78] , malware [55] injections, OS commanding [30] , malicious file [63] , [73] , buffer overflow [17] , API vulnerability exploitation [38]	Manipulate RIA components [86] , manipulate SOAP message variable [72] , SQL [39] , LDAP [96] , Javascript [78] , malware [55] injections, OS commanding [30] , direct object reference manipulation [121] , malicious file [73] , buffer overflow [17]	SQL injection [39] , SOAP message variable manipulation [72] , OS commanding [30]	Attack shared memory [125] , modify VM configuration [18] , [23]	Manipulate shared folder [36] , malicious code execution, VM escape [104]	

A threat modeling process

- Step 1: Scope your work
- Step 2: Determine Threats
- Step 3: Determine Countermeasures and Mitigation
- Step 4: Assess your work

A threat modeling process:

Step 1 - Scope your work

- is concerned with gaining an understanding of what you're working on
- can involve
 - Drawing diagrams, often data flow diagrams (DFD).
 - Identifying entry points to see where a potential attacker could interact with the application.
 - Trying to identifying "assets"
 - Identifying trust levels that represent the access rights that the application will grant to external entities.
 - Reading a user story or creating one.

A threat modeling process:

Step 2 - Determine Threats

- Critical to the identification of threats is using a threat categorization methodology.
- STRIDE is frequently used in threat modeling, and cyber kill chains including MITRE ATT&CK are frequently used for operational threat modeling.

A threat modeling process:

Step 3 - Determine Countermeasures and Mitigation

- A vulnerability may be mitigated with the implementation of a countermeasure.
- Such countermeasures can be identified using threat-countermeasure mapping lists.
- Prioritization of countermeasures is a complex and contentious topic. Many approaches exist, and organizations need to select ones that will work for them.
- The risk mitigation strategy
 - Accept: decide that the business impact is acceptable, and document who has chosen to accept the risk
 - Eliminate: remove components that make the vulnerability possible
 - Mitigate: add checks or controls that reduce the risk impact, or the chances of its occurrence
 - Transfer: Transfer risk to an insurer or customer.

A threat modeling process:

Step 4 - Assess your work

- First, determine if you've done the work.
- Second, showing one or more diagram(s), a threats list and a control list.

Threat Model Information (Sample)

- Application Version: 1.0
 - 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.
 - Document Owner: David Lowry
 - Participants: David Rook
 - Reviewer: Eoin Keary

Threat Model Information (Sample) (cont.)

■ External Dependencies (Sample)

- 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.
- The first area to consider when investigating external dependencies is the production environment and requirements.
- It is useful to understand how the application is or is not intended to be run.
 - ✓ For example, if the application is expected to be run on a server that has been hardened to the organization's hardening standard and it is expected to sit behind a firewall, then this information should be documented in the external dependencies section.
- External dependencies should be documented as follows:
 1. ID: A unique ID assigned to the external dependency.
 2. Description: A textual description of the external dependency.

Threat Model Information (Sample) (cont.)

■ External Dependencies (Sample)

ID	Description
1	The college library website will run on a Linux server running Apache. This server will be hardened per the college's server hardening standard. This includes the installation of the latest operating system and application security patches.
2	The database server will be MySQL and it will run on a Linux server. This server will be hardened per the college's server hardening standard. This will include the installation of the latest operating system and application security patches.
3	The connection between the web server and the database server will be over a private network.
4	The web server is behind a firewall and the only communication available is TLS.

Threat Model Information (Sample) (cont.)

- Entry and exit points define a trust boundary
- 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.
 - in an application can be layered. For example, each web page in a web application may contain multiple entry points.
 - show where data enters the system (i.e. input fields, methods) and exit points are where it leaves the system (i.e. dynamic output, methods), respectively.
 - should be documented as follows:
 1. ID: A unique ID assigned to the entry point. This will be used to cross-reference the entry point with any threats or vulnerabilities that are identified. In the case of layered entry points, a major.minor notation should be used.
 2. Name: A descriptive name identifying the entry point and its purpose.
 3. Description: A textual description detailing the interaction or processing that occurs at the entry point.
 4. Trust Levels: The level of access required at the entry point. These will be cross-referenced with the trust levels defined later in the document.

Threat Model Information (Sample) (cont.)

- Entry and exit points define a trust boundary
- Entry Points (examples)

ID	Name	Description	Trust Levels
1	HTTPS Port	The college library website will be only be accessible via TLS. All pages within the college library website are layered on this entry point.	(1) Anonymous Web User (2) User with Valid Login Credentials (3) User with Invalid Login Credentials (4) Librarian
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
1.2	Login Page	Students, faculty members and librarians must log in to the college library website before they can carry out any of the use cases.	(1) Anonymous Web User (2) User with Login Credentials (3) User with Invalid Login Credentials (4) Librarian
1.2.1	Login Function	The login function accepts user supplied credentials and compares them with those in the database.	(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

Threat Model Information (Sample) (cont.)

■ Exit Points

- might prove useful when attacking the client: for example, cross-site-scripting vulnerabilities and information disclosure vulnerabilities both require an exit point for the attack to complete.
- In the case of exit points from components handling confidential data (e.g. data access components), exit points lacking security controls to protect confidentiality and integrity can lead to disclosure of such confidential information to an unauthorized user.
- In many cases threats enabled by exit points are related to the threats of the corresponding entry point.
 - ✓ In the login example, error messages returned to the user via the exit point (the log in page) might allow for entry point attacks, such as account harvesting (e.g. username not found), or SQL injection (e.g. SQL exception errors).

Threat Model Information (Sample) (cont.)

■ Assets

- Many have something that one or more attackers are interested in; these items or areas of interest are often labelled “assets.”
- can be both physical assets and abstract assets.
 - ✓ For example, an asset of an application might be a list of clients and their personal information; this is a physical asset.
 - ✓ An abstract asset might be the reputation of an organization.
- are documented in this sample threat model as follows:
 1. ID: A unique ID is assigned to identify each asset. This will be used to cross-reference the asset with any threats or vulnerabilities that are identified.
 2. Name: A descriptive name that clearly identifies the asset.
 3. Description: A textual description of what the asset is and why it needs to be protected.
 4. Trust Levels: The level of access required to access the entry point is documented here. These will be cross-referenced with the trust levels defined in the next step.

Threat Model Information (Sample) (cont.)

■ Assets (example)

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
1.2	Librarian Login Details	The login credentials that a Librarian will use to log into the College Library website.	(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

Threat Model Information (Sample) (cont.)

■ Assets (example)

ID	Name	Description	Trust Levels
2	System	Assets relating to the underlying system.	
2.1	Availability of College Library Website	The College Library website should be available 24 hours a day and can be accessed by all students, college faculty members, and librarians.	(5) Database Server Administrator (6) Website Administrator
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
2.3	Ability to Execute SQL as a Database Read User	This is the ability to execute SQL select queries on the database, and thus retrieve any information stored within the College Library database.	(5) Database Server Administrator (8) Database Read User (9) Database Read/Write User
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

Threat Model Information (Sample) (cont.)

■ Assets (example)

ID	Name	Description	Trust Levels
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
3.2	Access to the Database Server	Access to the database server allows you to administer the database, giving you full access to the database users and all data contained within the database.	(5) Database Server Administrator
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
3.4	Access to Audit Data	The audit data shows all audit-able events that occurred within the College Library application by students, staff, and librarians.	(6) Website Administrator

Threat Model Information (Sample) (cont.)

■ Trust Levels

- represent the access rights that the application will grant to external entities.
- are cross-referenced with the entry points and assets.
- allow one to define the access rights or privileges required at each entry point, and those required to interact with each asset.
- are documented in the threat model as follows:
 1. ID: A unique number is assigned to each trust level. This is used to cross-reference the trust level with the entry points and assets.
 2. Name: A descriptive name that allows you to identify the external entities that have been granted this trust level.
 3. Description: A textual description of the trust level detailing the external entity who has been granted the trust level.

Threat Model Information (Sample) (cont.)







■ 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.

Threat Model Information (Sample) (cont.)

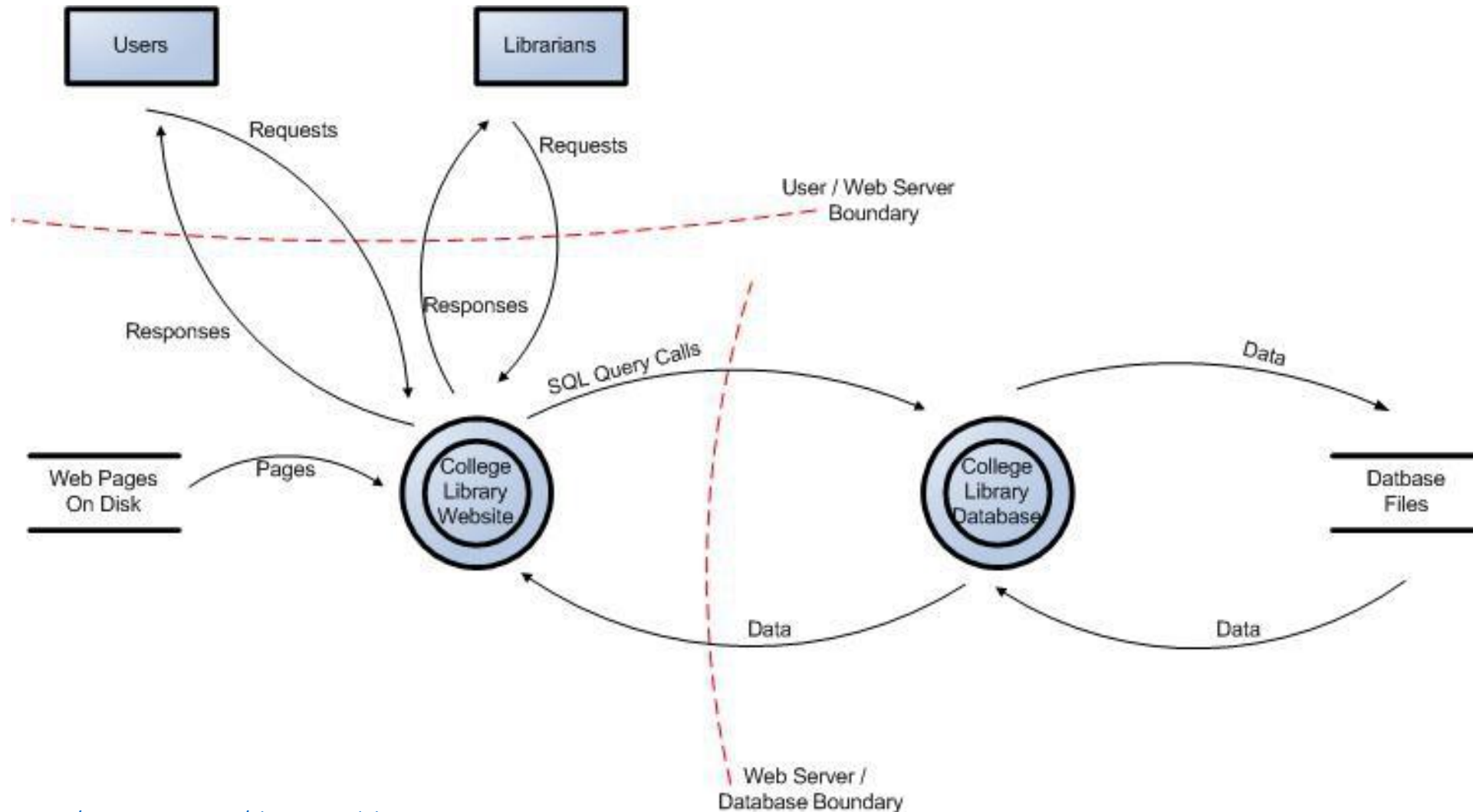
■ Data Flow Diagrams (DFD)

- There are a number of symbols that are used in DFDs for threat modeling.

Symbol	Name	Description
	External Entity	The external entity shape is used to represent any entity outside the application that interacts with the application via an entry point.
	Process	The process shape represents a task that handles data within the application. The task may process the data or perform an action based on the data.
	Multiple Process	The multiple process shape is used to present a collection of subprocesses. The multiple process can be broken down into its subprocesses in another DFD.
	Data Store	The data store shape is used to represent locations where data is stored. Data stores do not modify the data, they only store data.
	Data Flow	The data flow shape represents data movement within the application. The direction of the data movement is represented by the arrow.
	Privilege Boundary	The privilege boundary (or trust boundary) shape is used to represent the change of trust levels as the data flows through the application. Boundaries show any location where the level of trust changes.

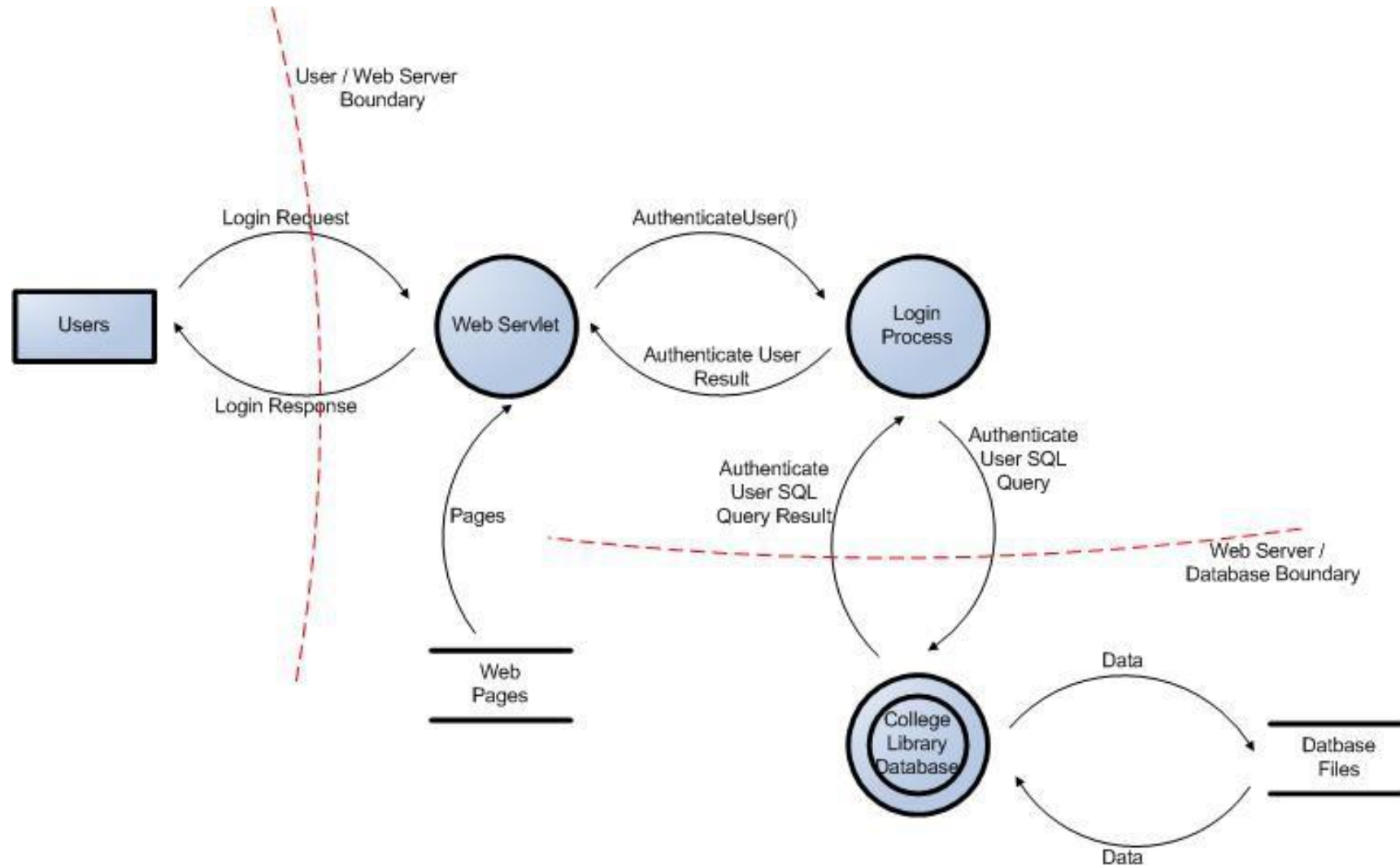
DFD example

- *Figure 1: Data Flow Diagram for the College Library Website.*



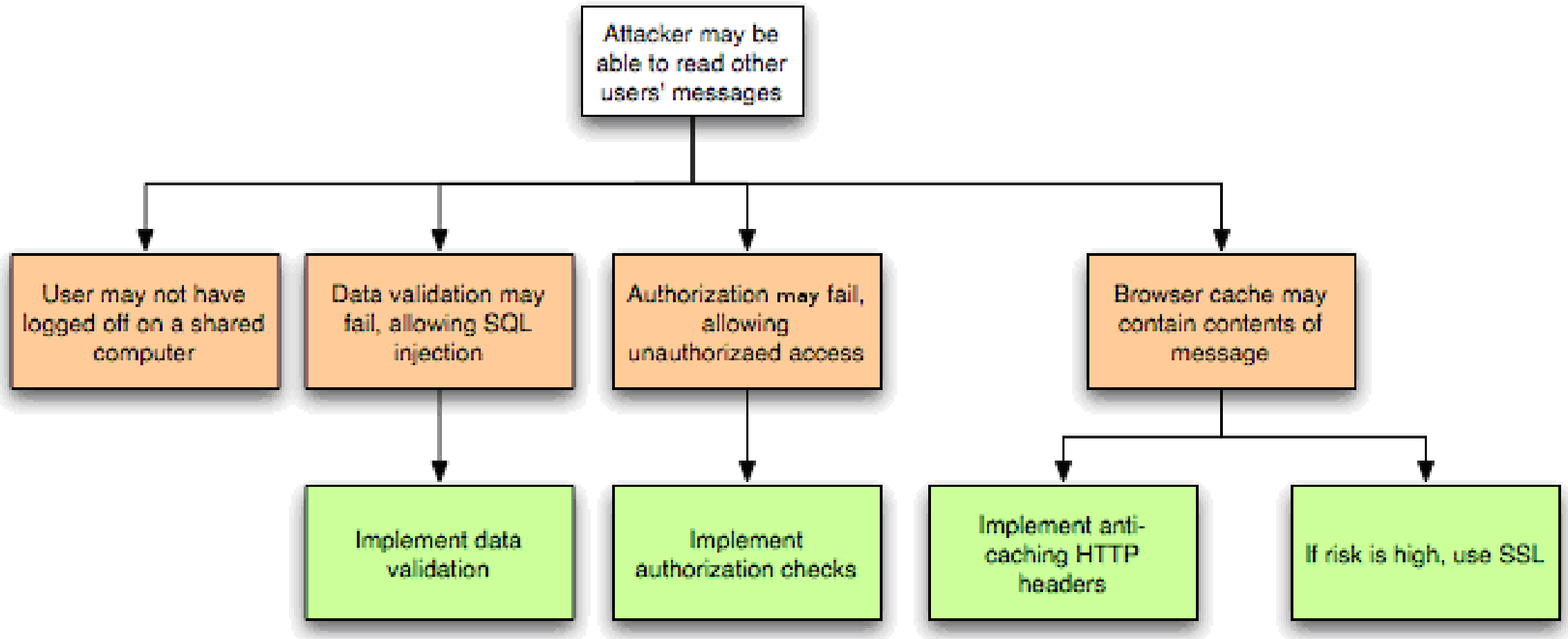
DFD example

- Figure 2: User Login Data Flow Diagram for the College Library Website.



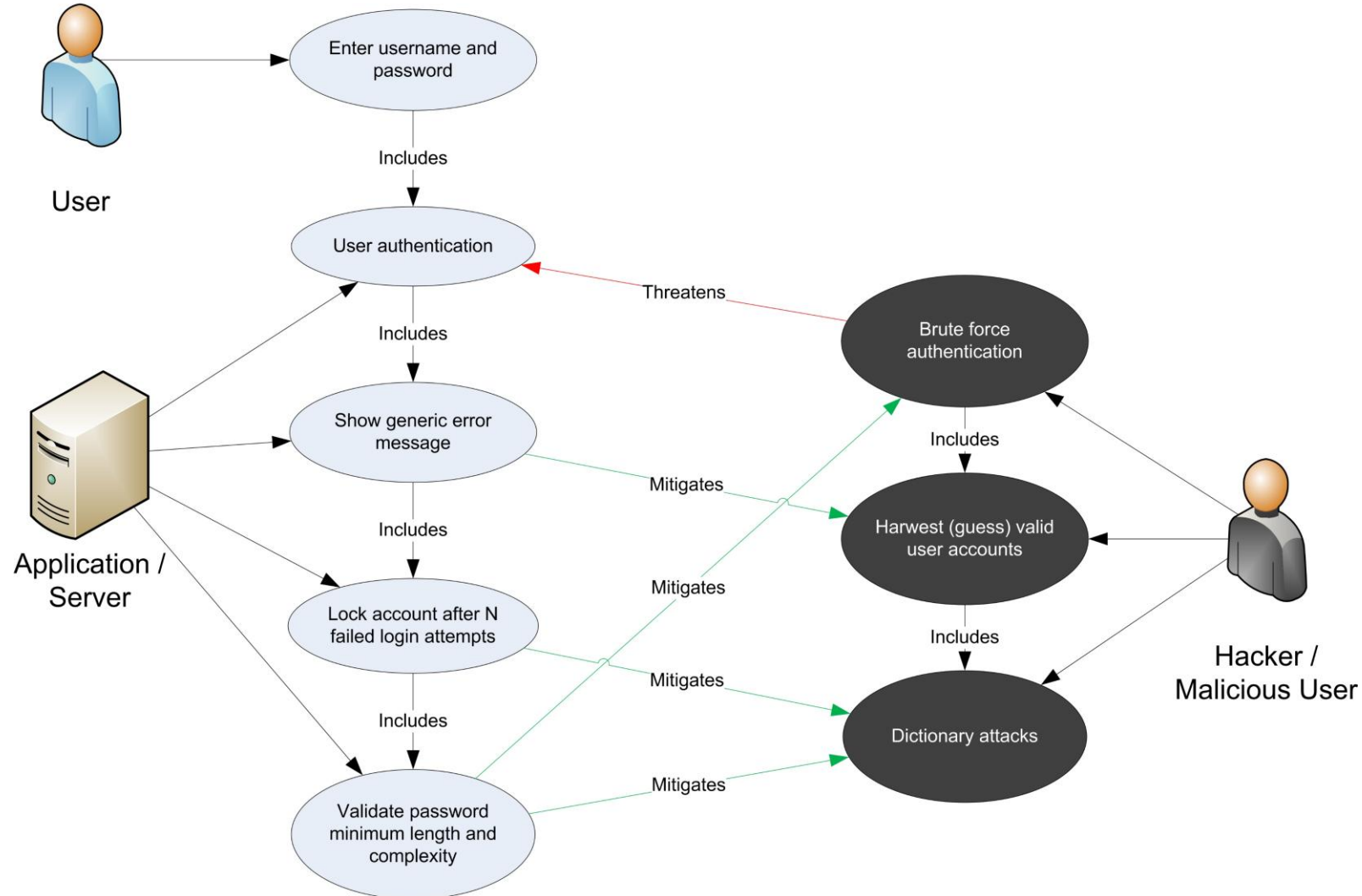
Threat Analysis

- Figure 3: Threat Tree Diagram.



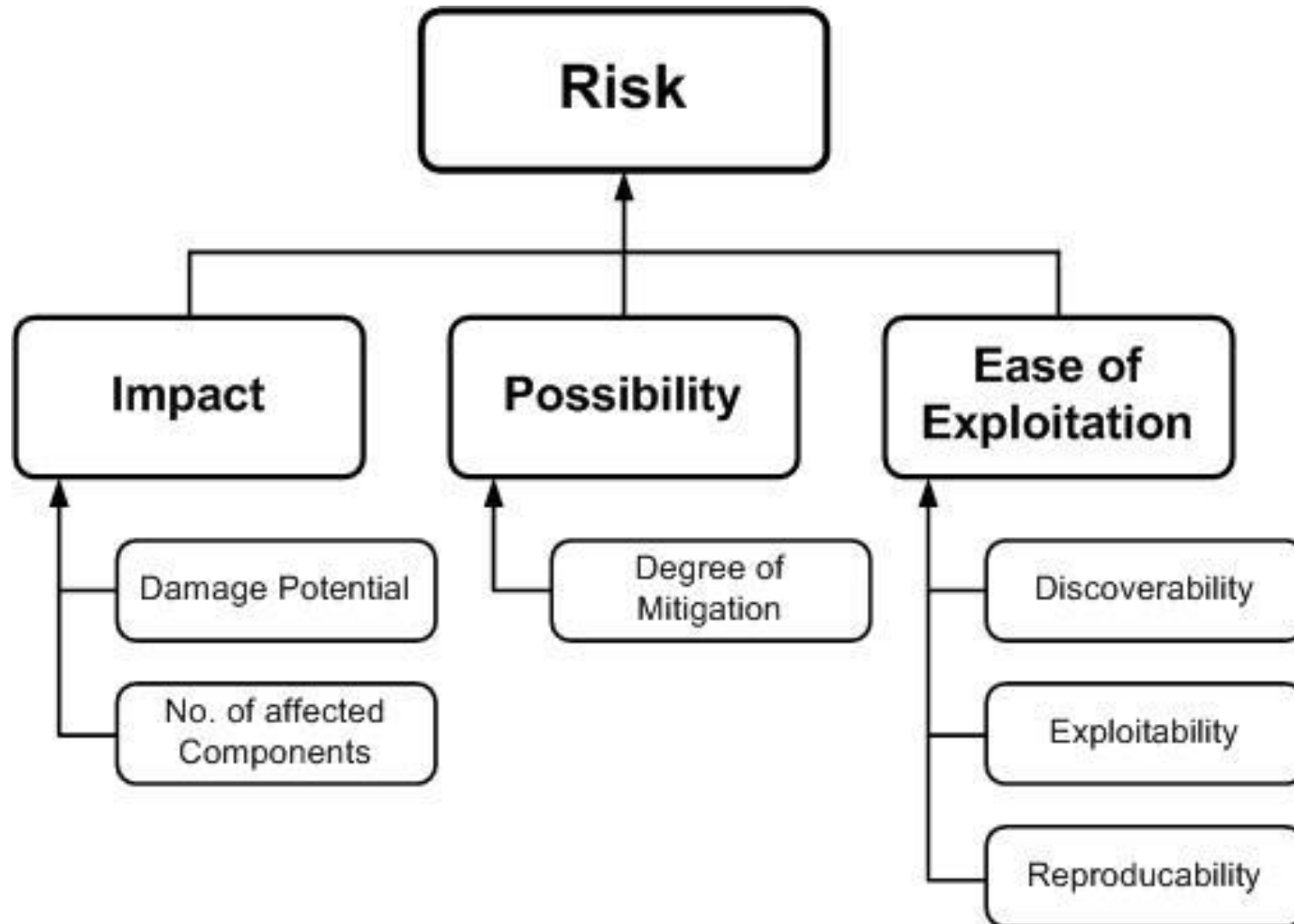
Threat Analysis

■ Figure 4: Use and Misuse Cases



Ranking of Threats

- Figure 5: Ranking Risk Factors.



Qualitative Risk Model

- determine **ease of exploitation**:
 1. Can an attacker exploit this remotely?
 2. Does the attacker need to be authenticated?
 3. Can the exploit be automated?
- determine the **damage potential** are:
 1. Can an attacker completely take over and manipulate the system?
 2. Can an attacker gain administration access to the system?
 3. Can an attacker crash the system?
 4. Can the attacker obtain access to sensitive information such as secrets or Personally identifiable information (PII)?
- determine the number of components that are affected by a threat:
 1. How many connected data sources and systems can be impacted?
 2. How many layers into infrastructure components can the threat agent traverse?
- These examples help in the calculation of the overall risk values by assigning qualitative values such as High, Medium and Low to the likelihood and impact factors

Determine Countermeasures and Mitigation

■ STRIDE Threat & Mitigation Techniques

Threat Type	Mitigation Techniques
Spoofing Identity	<ol style="list-style-type: none">1. Appropriate authentication2. Protect secret data3. Don't store secrets
Tampering with data	<ol style="list-style-type: none">1. Appropriate authorization2. Hashes3. MACs4. Digital signatures5. Tamper resistant protocols
Repudiation	<ol style="list-style-type: none">1. Digital signatures2. Timestamps3. Audit trails

Determine Countermeasures and Mitigation

■ STRIDE Threat & Mitigation Techniques (cont.)

Threat Type	Mitigation Techniques
Information Disclosure	<ol style="list-style-type: none">1. Authorization2. Privacy-enhanced protocols3. Encryption4. Protect secrets5. Don't store secrets
Denial of Service	<ol style="list-style-type: none">1. Appropriate authentication2. Appropriate authorization3. Filtering4. Throttling (e.g. reduce clock speed)5. Quality of service
Elevation of privilege	<ol style="list-style-type: none">1. Run with least privilege

References

- Stallings and Brown, Computer Security: Principles and Practice, 4th eds Chap 6, 7, 10, 11
- W. Du, Computer & Internet Security: A Hands-on Approach, 2nd eds.
 - Chaps 4, 5; Chap 10, 11, 12, 15, 16
- R. Anderson, Security Engineering, 2nd eds., Chap 21: Network Attack and Defense, <https://www.cl.cam.ac.uk/~rja14/Papers/SEv2-c21.pdf>