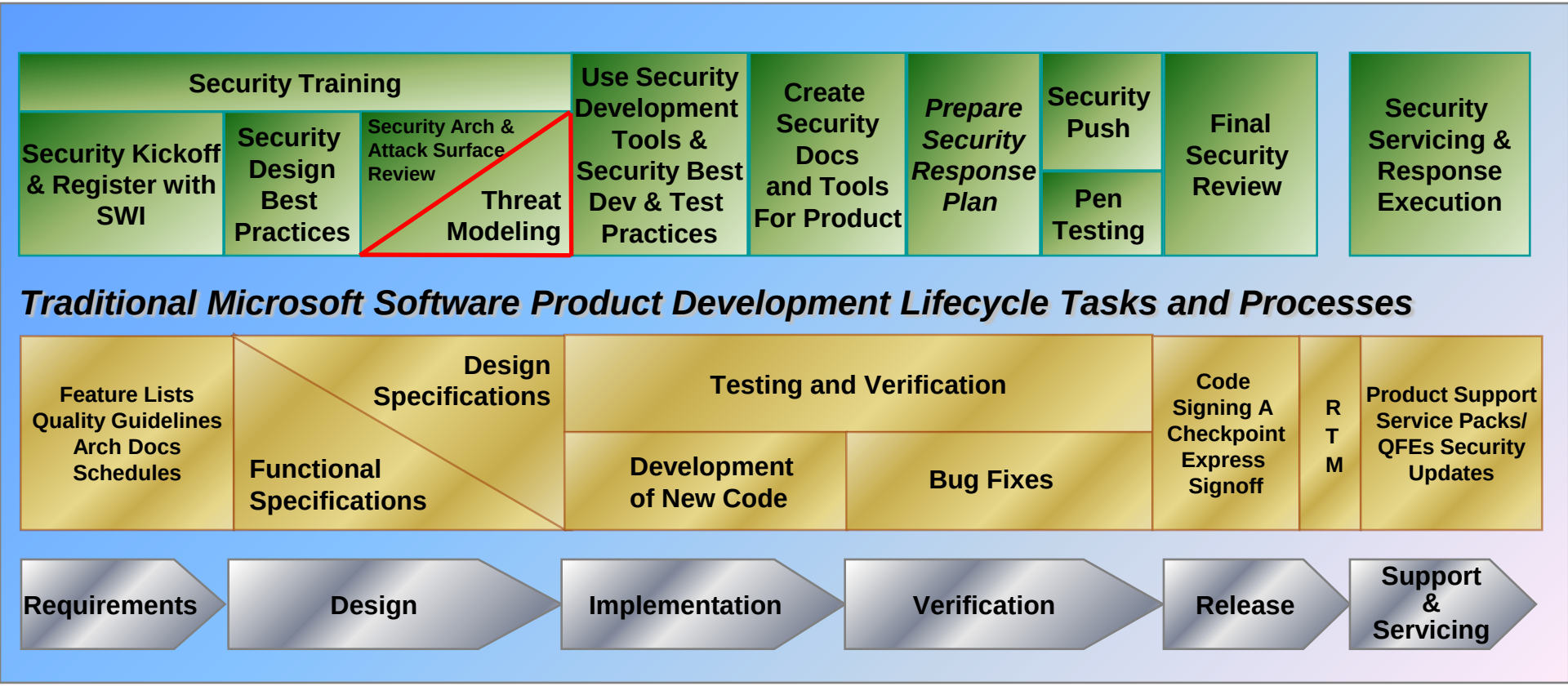
Threat Modeling (Risk Analysis)



Threat Modeling

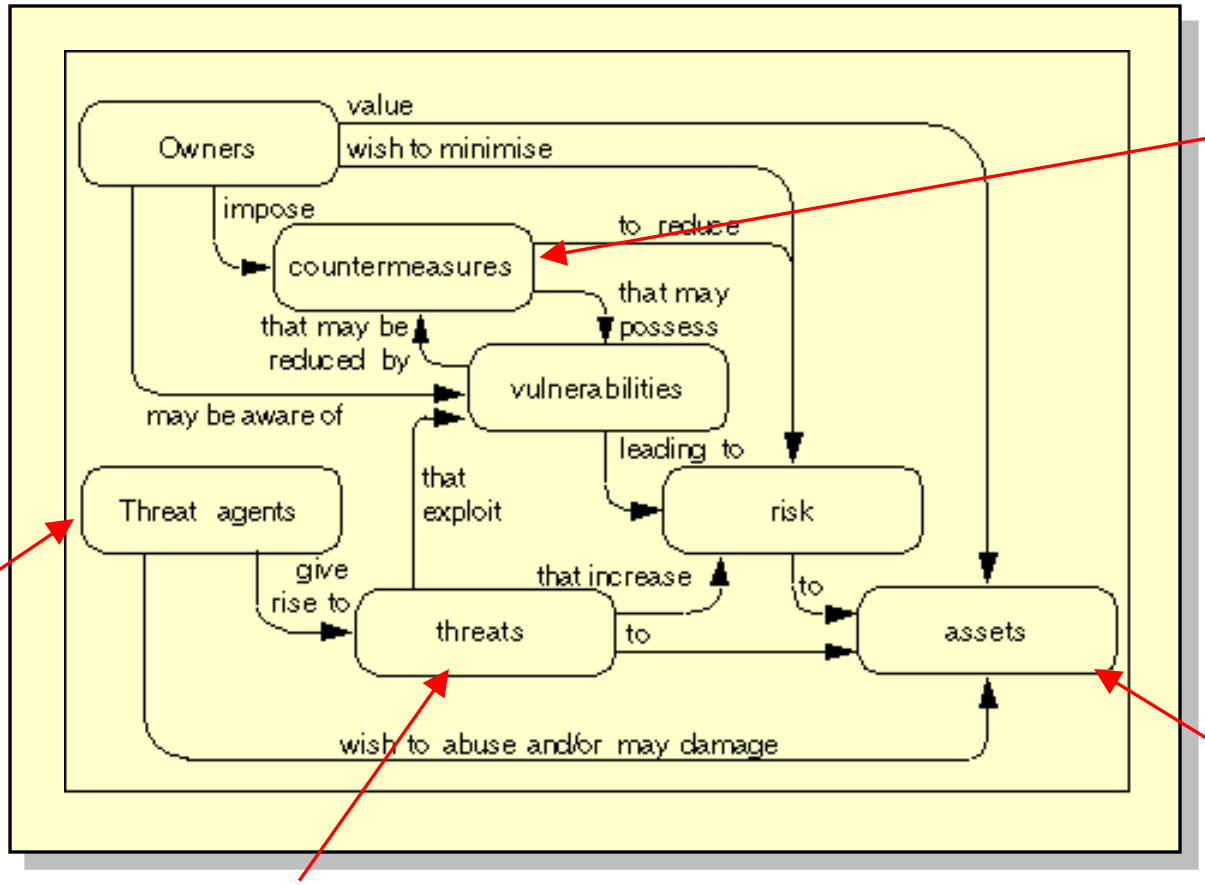
Some Important Definitions

- Threat Agent
  - Someone who could do harm to a system (also adversary)
- Threat
  - An adversary's goal
- Vulnerability
  - A flaw in the system that could help a threat agent realize a threat
- Asset
  - Something of value to valid users and adversaries alike
- Attack
  - When a motivated and sufficiently skilled threat agent takes advantage of a vulnerability

Why do Threat Modelling?

- To identify the threats your component faces and to challenge any assumptions that have been made
- To prioritise other security-related efforts
  - Code reviews
  - Fuzz testing
  - Penetration testing
- To look at the product with a different set of eyes
  - Highly technical and motivated criminal
  - Not your typical happy, paying customer
- To document everything for future generations

Why Threat Modeling?



We must put appropriate defenses in our products

Because attackers

Want to attack

Our customers

Source: Common Criteria for Information Technology Security Evaluation v2.1

Use and Evolve Threat Models in …

- Design
  - Mitigation and security designs
  - Drives attack surface design
- Development
  - Determines the most “insecure” portions of your application
  - General mitigations included in development guidelines
  - Drives security in code reviews and exit criteria
- Testing and Production
  - Drives security testing strategy (threats and mitigations)
  - All threats and mitigations must be tested
  - The job of a good security tester is to find other conditions in the threat tree
  - Attack points derived from threat model

Benefits of Threat Modeling (1)

- Contributes to the risk management process because threats to software and infrastructure are risks to the user and environments deploying the software.
- Uncovers threats to the system before the system is committed to code.
- Revalidates the architecture and design by having the development team go over the design again

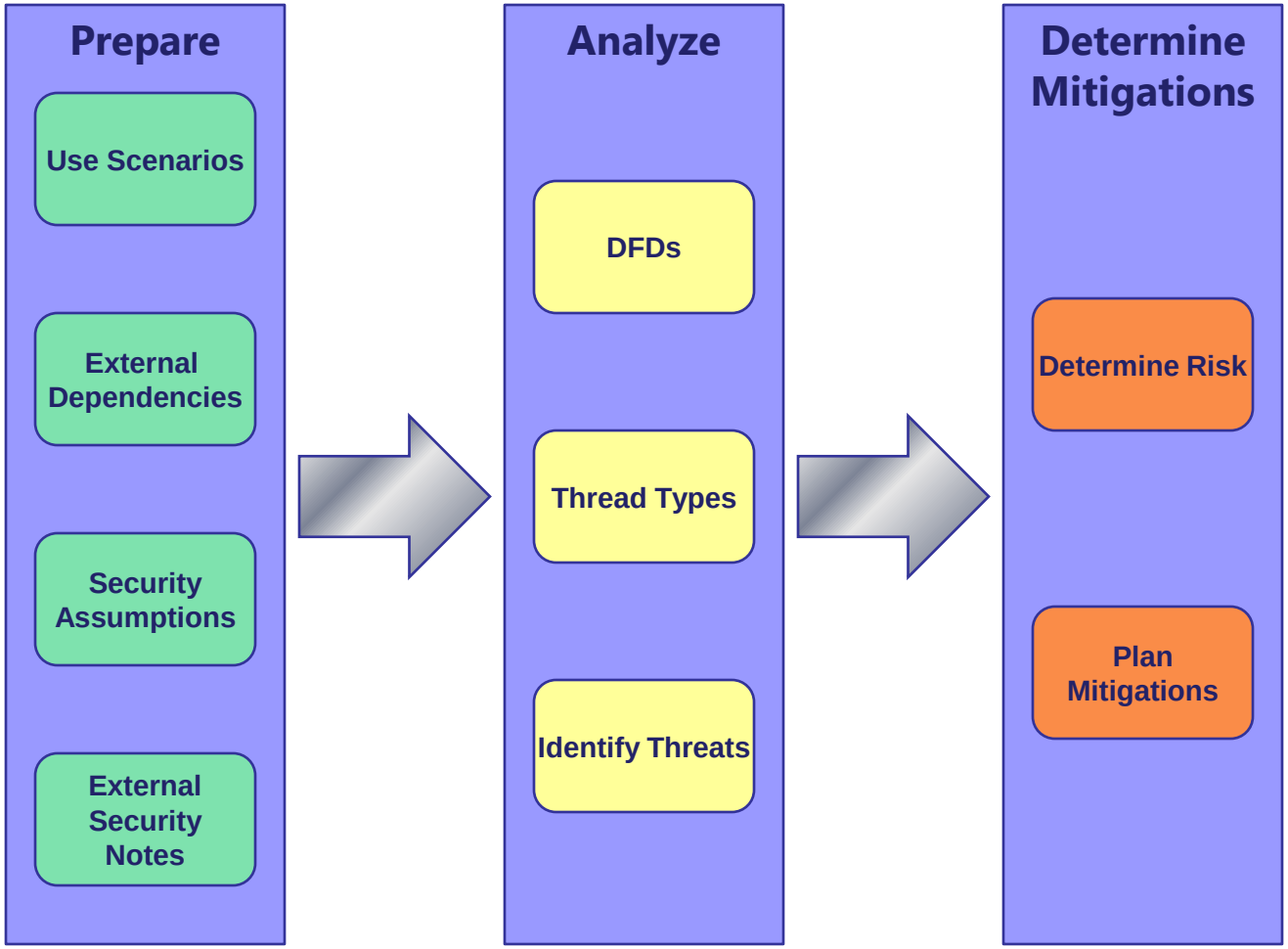
Benefits of Threat Modeling (2)

- Forces development staff to look at the design from a different viewpoint – that of security and privacy.
- Helps clarify the selection of appropriate countermeasures for the application and environment.
- Contributes to the Attack Surface Reduction process for the software.
- Helps guide the code review process.
- Guides the penetration testing process.

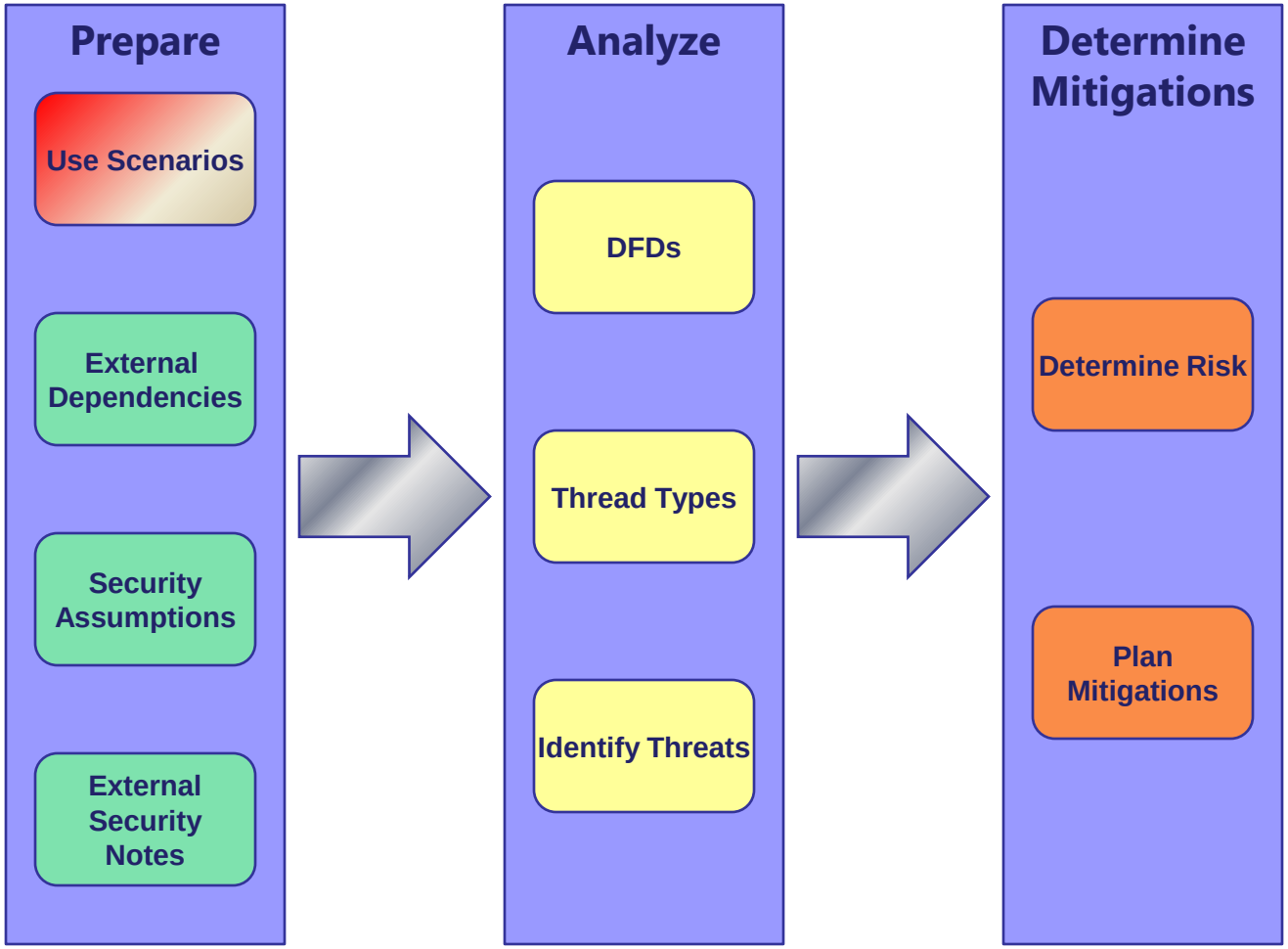
Key Concepts

- A Threat Model describes a system's threat profile.
- A threat is not a vulnerability.
- The point of a threat model is more than just finding vulnerabilities.
- A system is anything that exposes functionality to an end user, and can describe anything from a single feature to a web application and its supporting infrastructure.

Threat Modeling Process



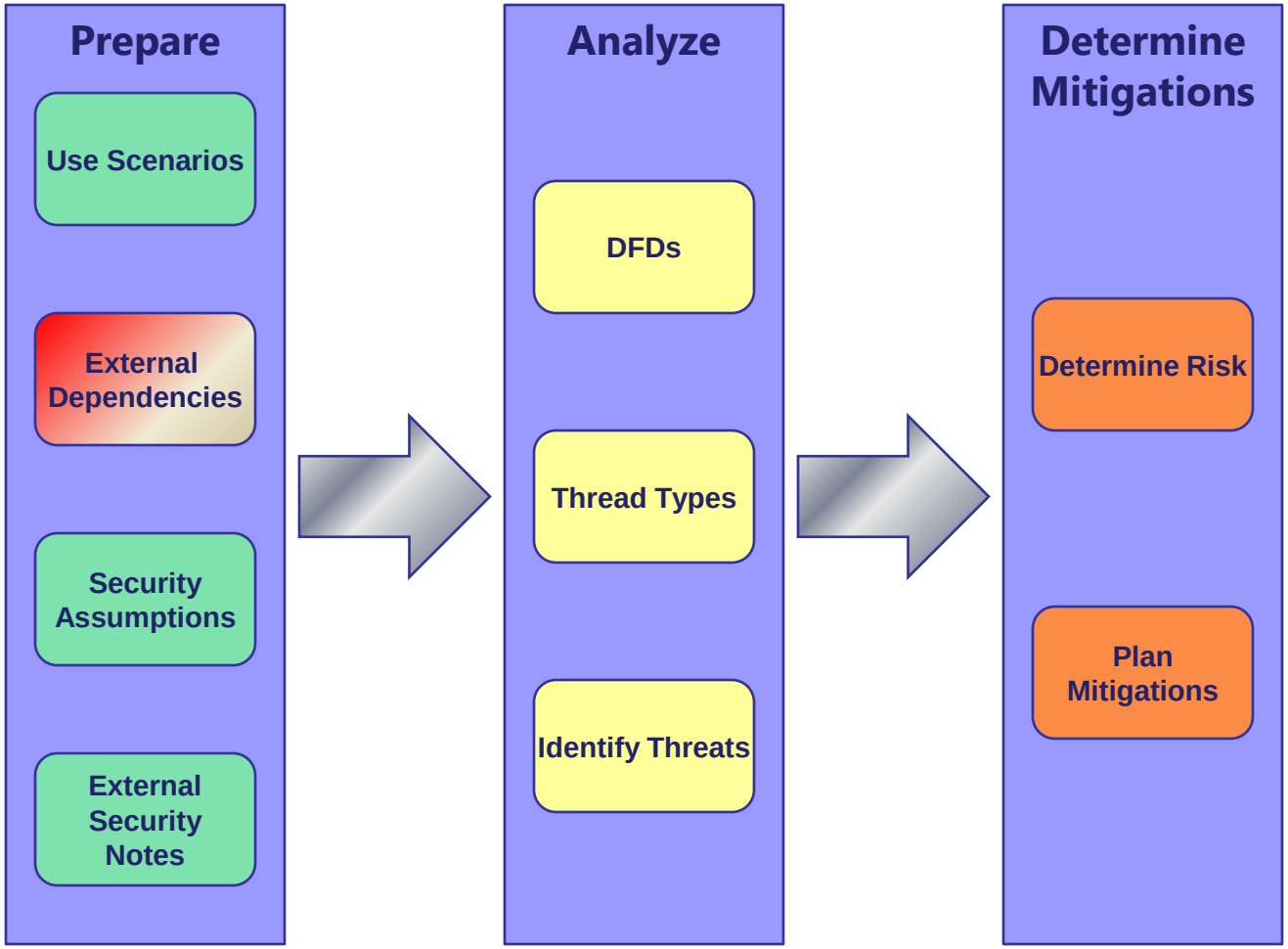
Threat Modeling Process



Define Scenarios

- Identify what the application does
- Define the most common and realistic use scenarios for the application
  - Example from Windows Server 2003 and Internet Explorer
    - "Think about an admin browsing the Internet from a Domain Controller"
- Bounds the scope of what you need to model

Threat Modeling Process



Identify External Dependencies

- External dependencies are requirements levied on systems outside of the system being modeled.
- They are dependencies on a certain behavior or specification compliance in an external system that, if broken, could cause threats in the system being modeled to manifest vulnerabilities.
- Often, these dependencies describe functions such as algorithm consistency across systems. For example, if two systems both normalize a string of text and take action based on the result, it is typically important that the normalized representation is the same across both systems.

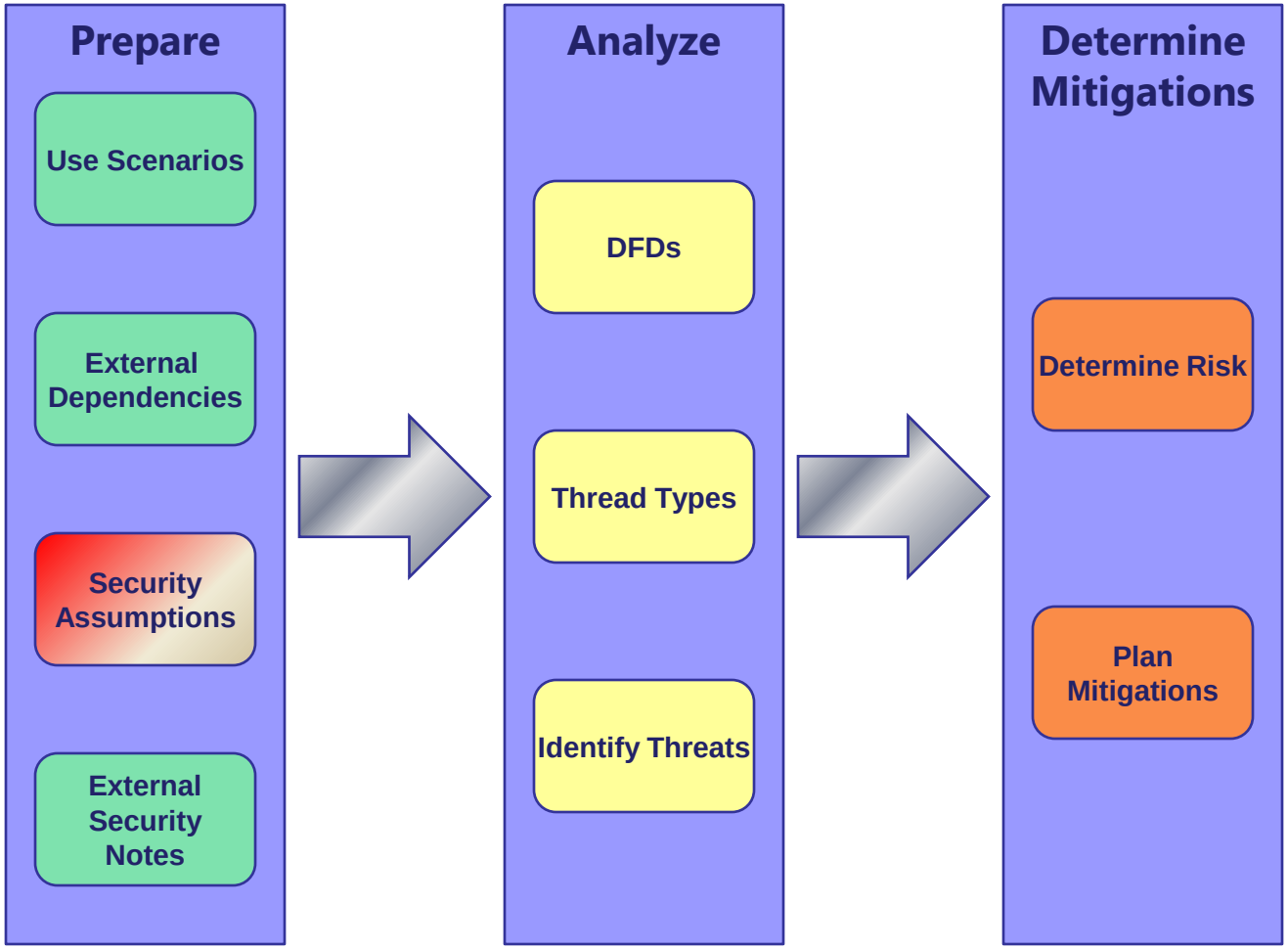
Sample – Pet Shop 4.0

- Example of an e-commerce application (Pet Shop 2006)
- Sample Threat Model from (Howard and Lipner 2006)

External Dependencies

- Client
  - IE 6.0 or later or FireFox 1.5 or later
- Servers
  - Windows Server 2003 SP1
  - IIS 6.0 (Web Servers)
  - ASP.NET 2.0 (Web Servers)
  - SQL Server 2000, SQL Server 2005, or Oracle 10g (DB Servers)
  - Server 2003 Terminal Services (All Servers)
  - MSDTC (all Computers)

Threat Modeling Process



Define Security Assumptions

- It is possible, even beneficial, to start the Threat Modeling process before a system is implemented.
- Security Assumptions are used when some or all of the system is in the design phase, and dictate specifics about how features must be implemented for the system to remain secure.
- Security Assumptions should be validated on completion of the implementation, in addition to revising the Threat Model as a whole to reflect the implementation.

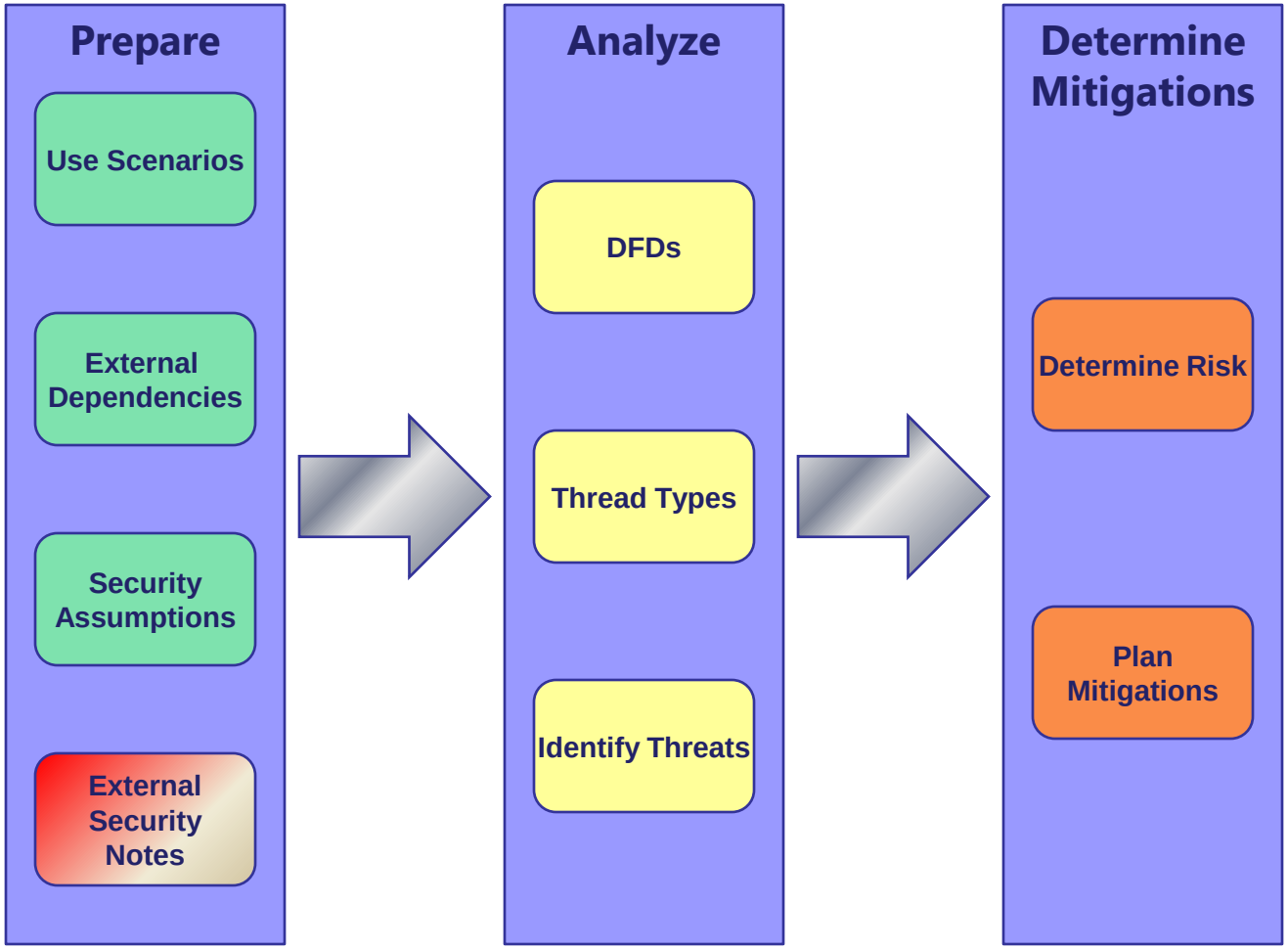
Security Assumptions (1)

- No sensitive data is deliberately persisted on the client, but sensitive data is sent over SSL/TLS connections, and some browsers might locally cache data sent over these connections
- DPAPI is used on the server to protect sensitive connection strings and encryption keys for non-admins
- The database server holds authentication information
- IIS 6.0 and ASP.NET enforce authentication correctly

Security Assumptions (2)

- The server application setup program correctly configures the ACL for the Web.config file
- Only valid admins administer any server by using Terminal Services or phyisically accessing the server when needed
- …

Threat Modeling Process

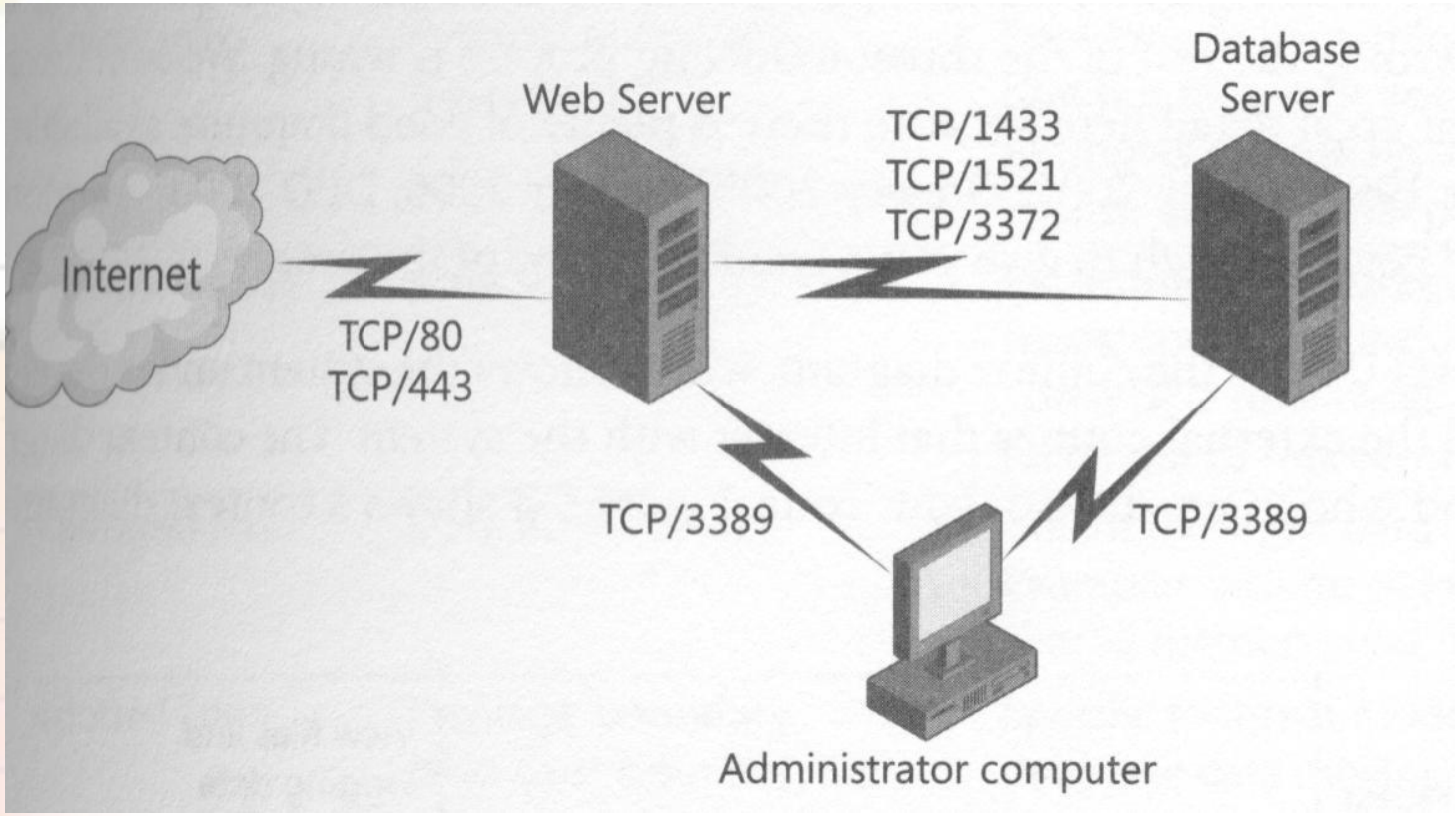


External Security Notes

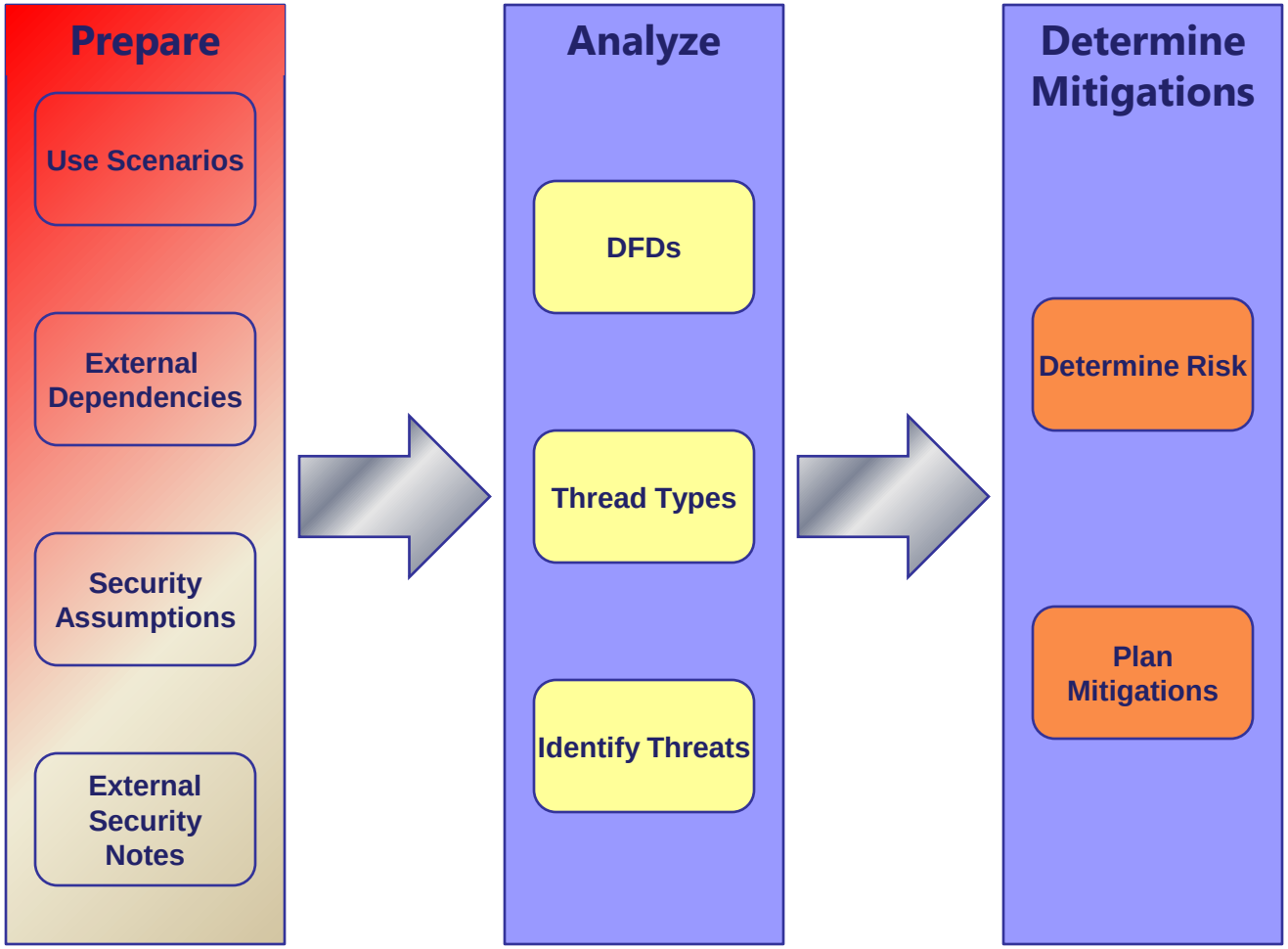
- External Security Notes are the counterpart to external dependencies.
- They provide security-relevant information to users that interface with the system being modeled.
- This information can be in the form of warnings against potential misuse that, while not constituting a vulnerability in the system being modeled, may surface a vulnerability in another system if it is not used correctly.
- Or, the information can be in the form of guarantees that the system makes for users.
- As an example, it may contain the specification for how filenames are normalized internal to the system.

External Security Information

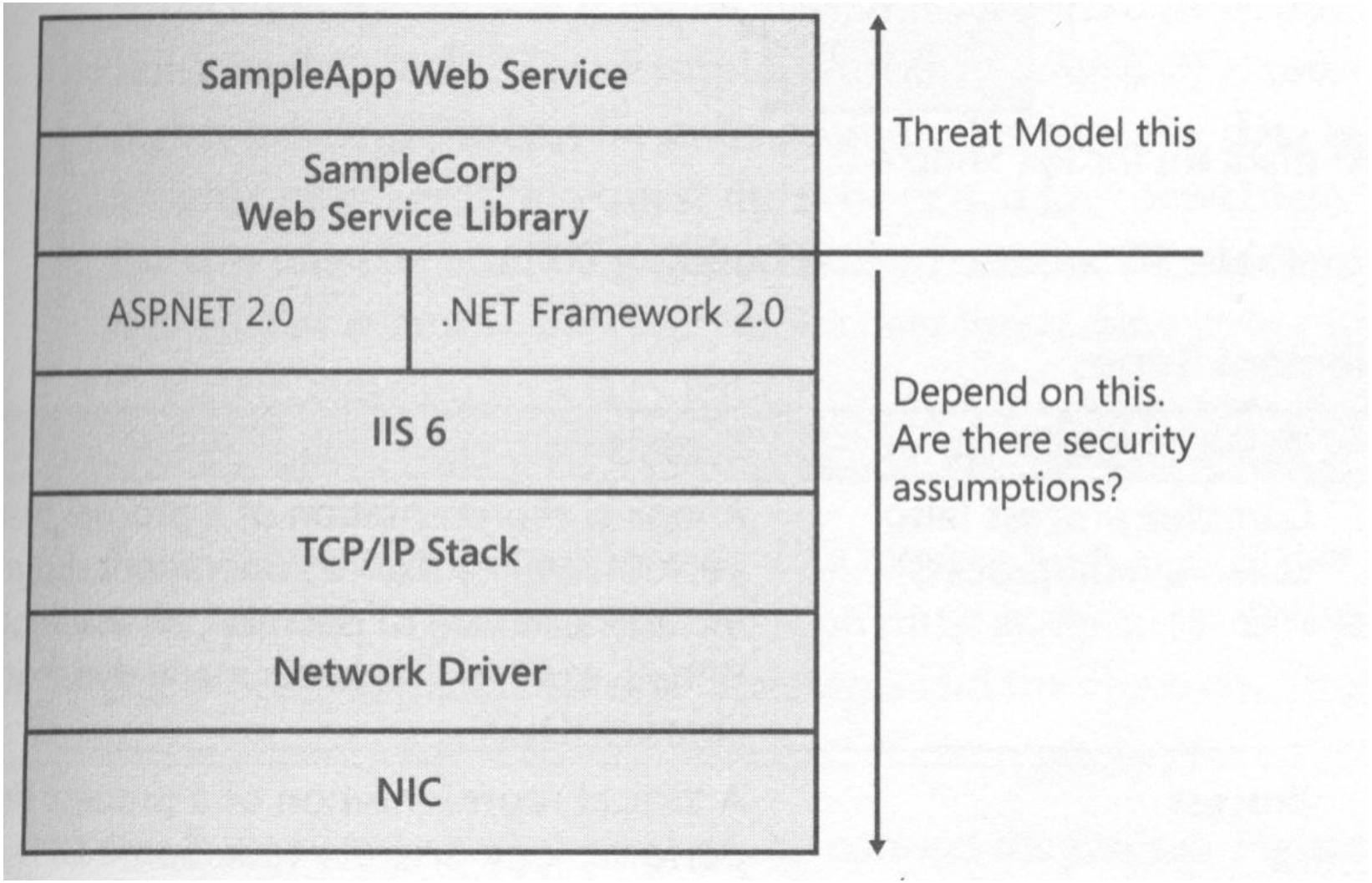
- Admins can change any setting in the system, including the Web service



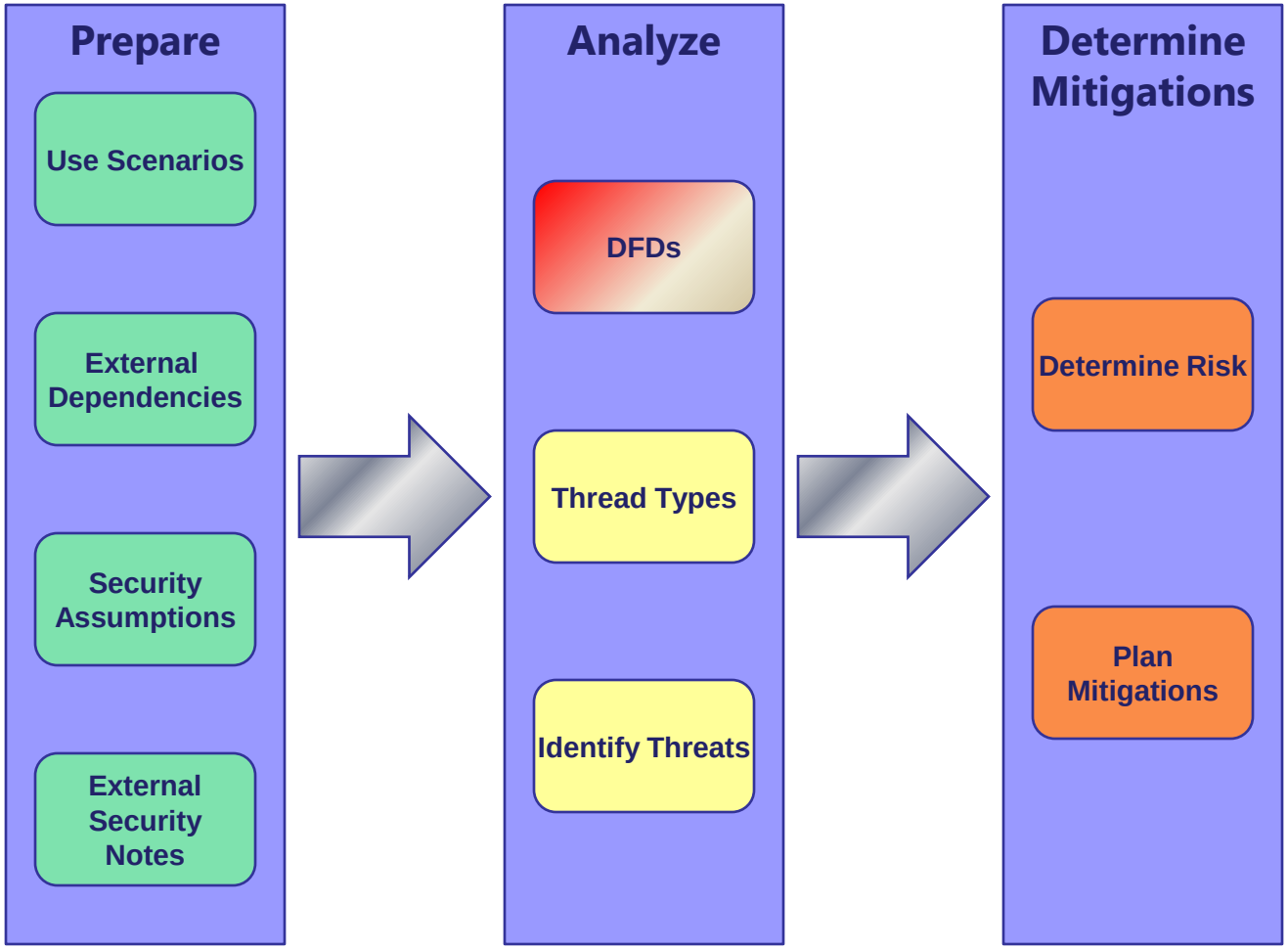
Threat Modeling Process



What Is Modeled and What Do You Depend On?



Threat Modeling Process



Model the System

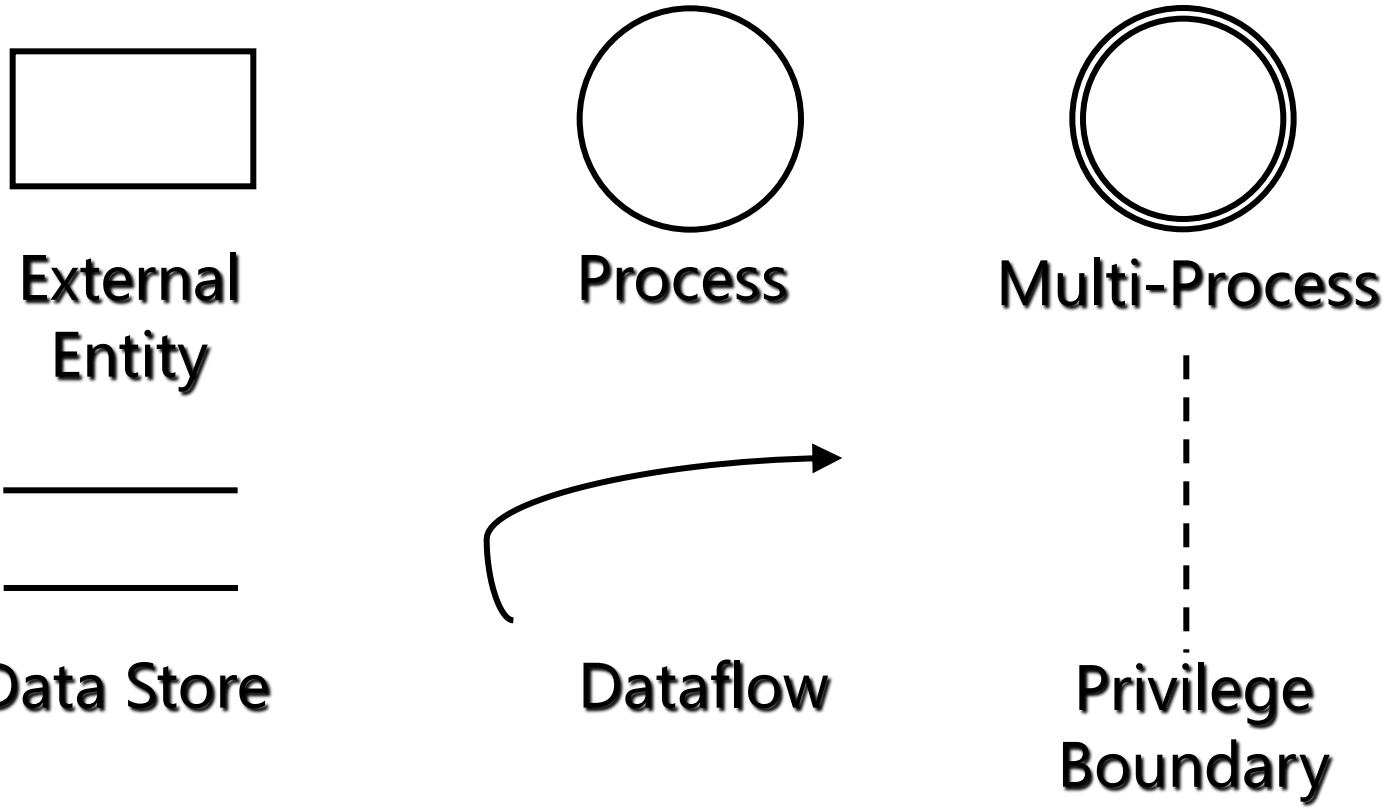
- Modeling the system is critical to determining threats.
- It helps the threat modeling team understand the adversary's view of the system.
- It helps the team understand the internal workings of the system, allowing them to identify design- and implementation-specific threats.

Data Flow Diagrams

- A DFD is a graphical representation of how data enters, leaves, and traverses your component
  - It is not a Class Diagram or Flow Chart!
  - Shows all data sources and destinations
  - Shows all relevant processes that data goes through
- Good DFDs are critical to the process
  - This point can't be emphasised enough!
  - Building DFDs == understanding the system
  - Analysing DFDs == understanding the threats

Create the DFD's

- Most "whiteboard architectures" are DFD-like



Privilege Boundaries

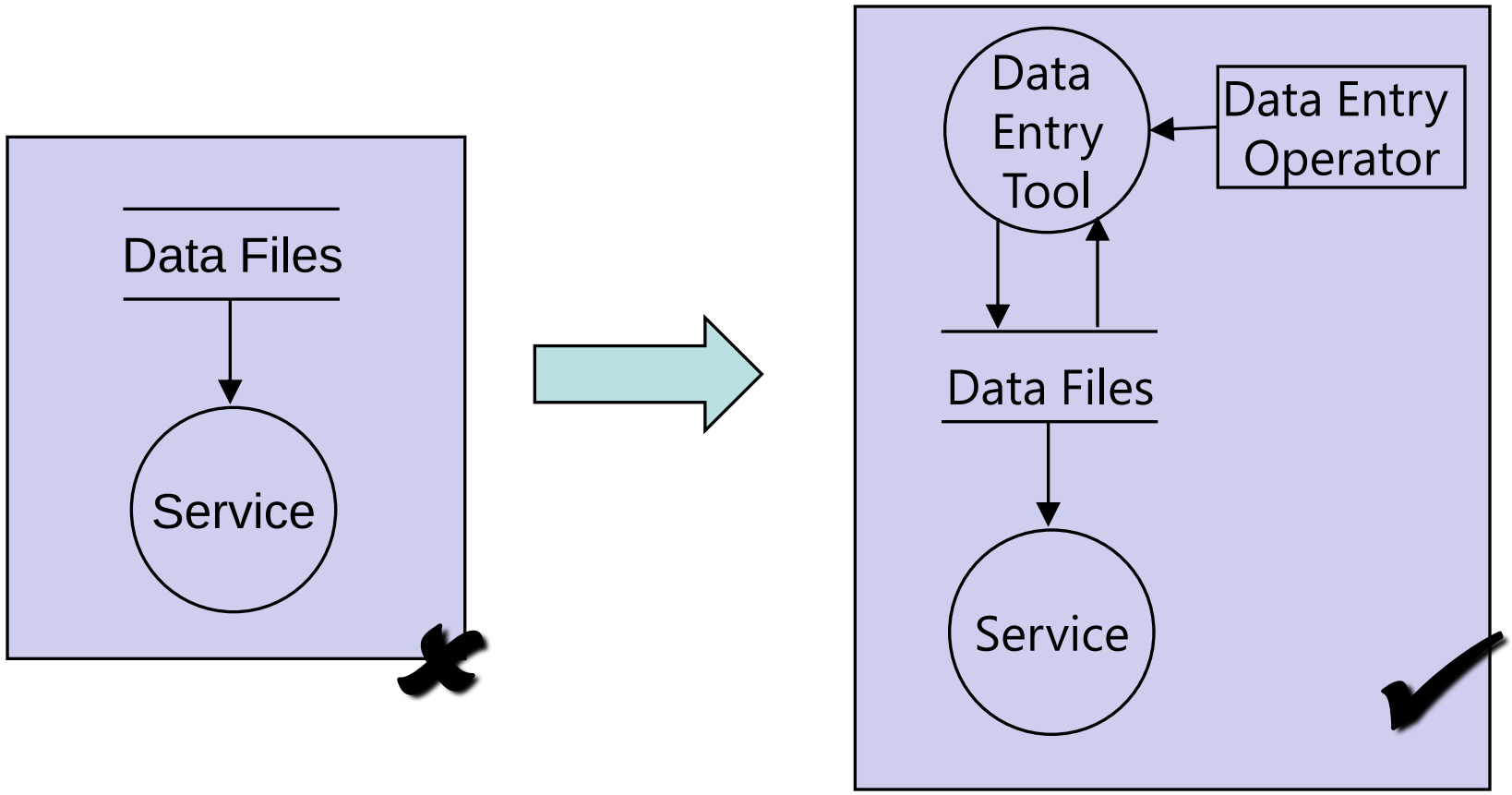
- Specific DFD addition to TMs
- Boundary between DFD elements with different privilege levels
  - Machine boundary (data from the other machine could be anonymous)
  - Process boundary
    (e.g.; User process ←→ SYSTEM process)
  - Kernel ←→ User mode

Types of DFDs

- Context Diagram
  - Very high-level; entire component / product / system
- Level 0 Diagram
  - High level; single feature / scenario
- Level 1 Diagram
  - Low level; detailed sub-components of features
- Level *n* Diagram
  - Even more detailed; unlikely to go beyond Level 2

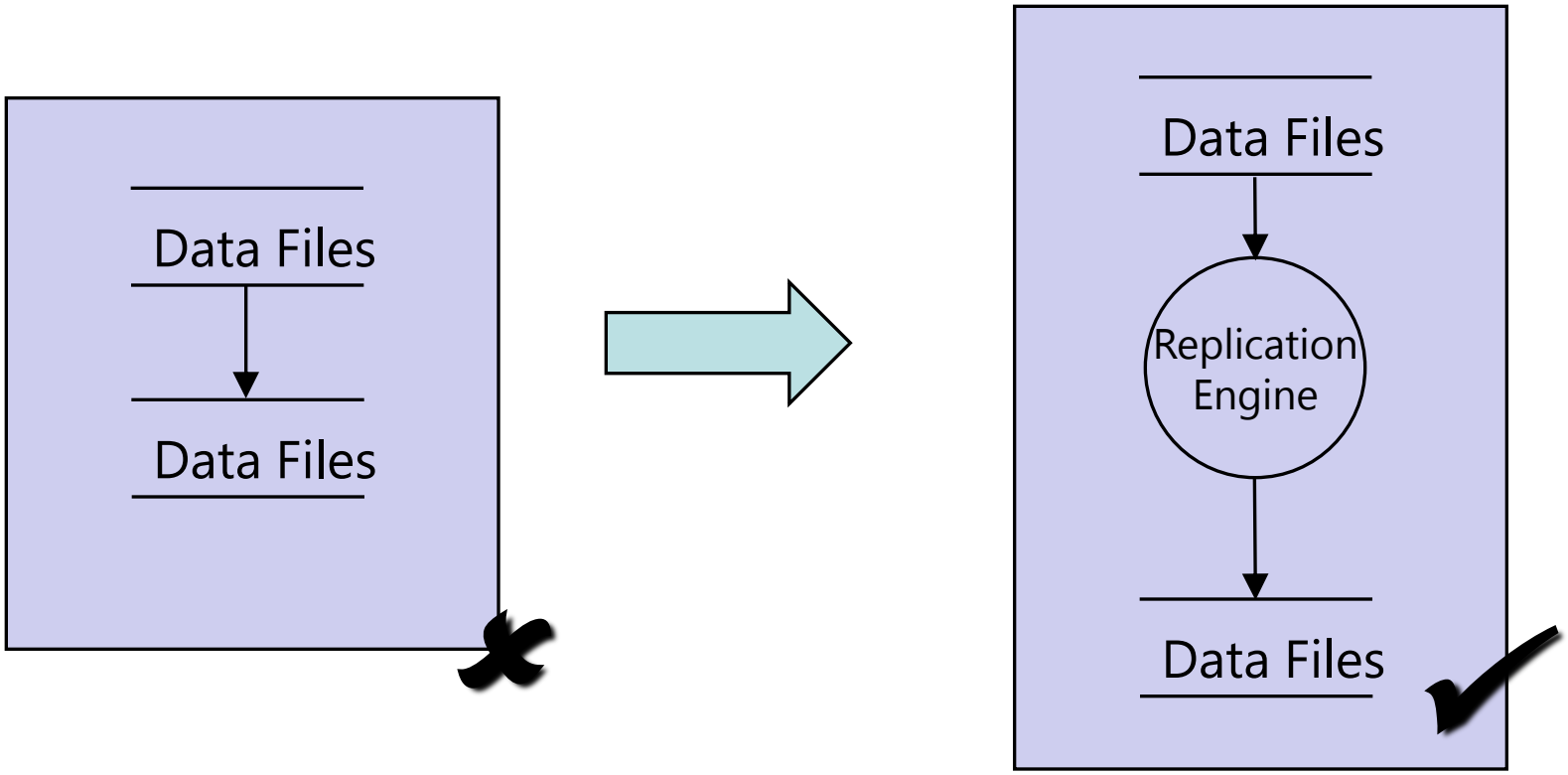
Common DFD "bugs"

(1) How does the data get into the data store?



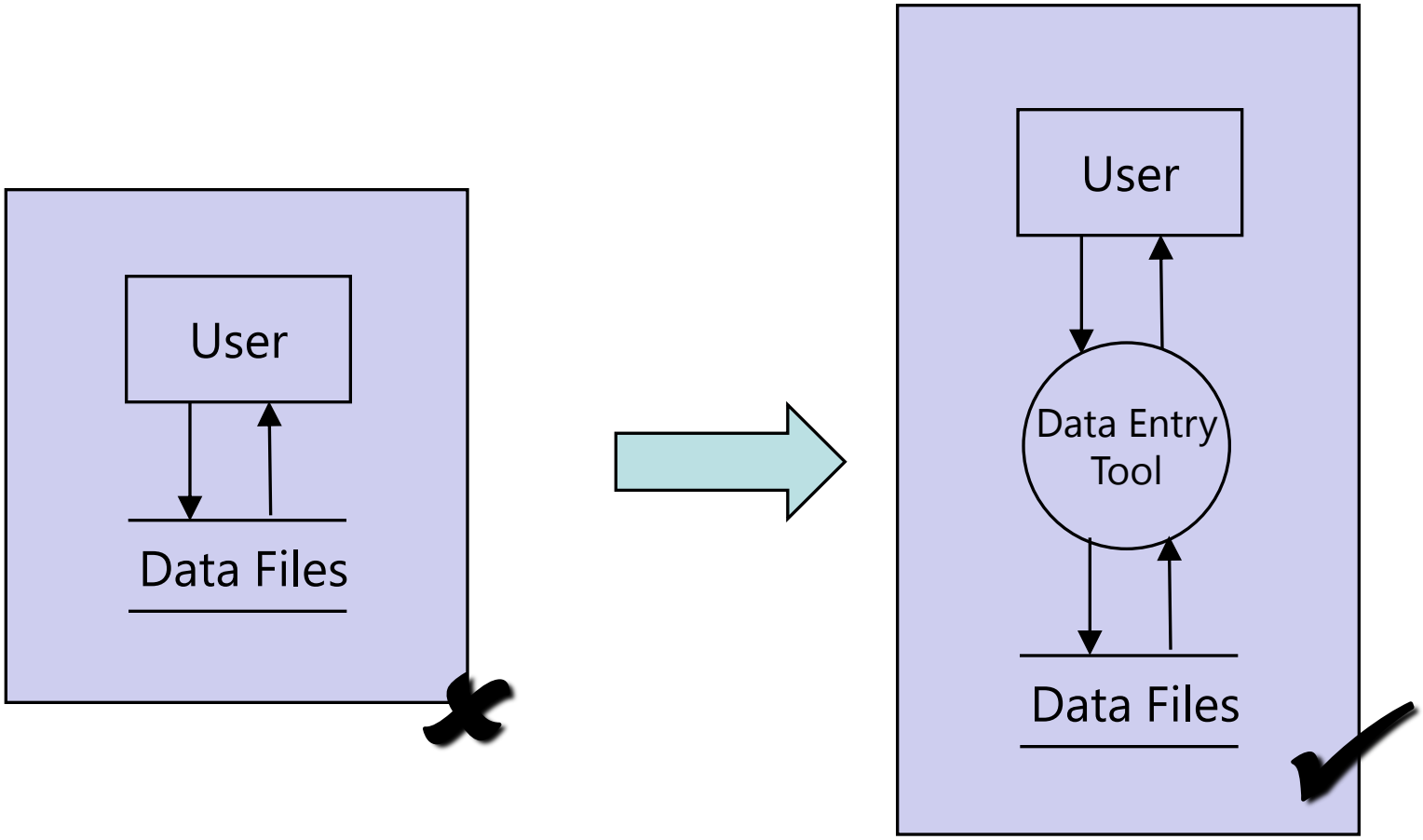
Common DFD “bugs”

(2) How does data move from one data store to another?

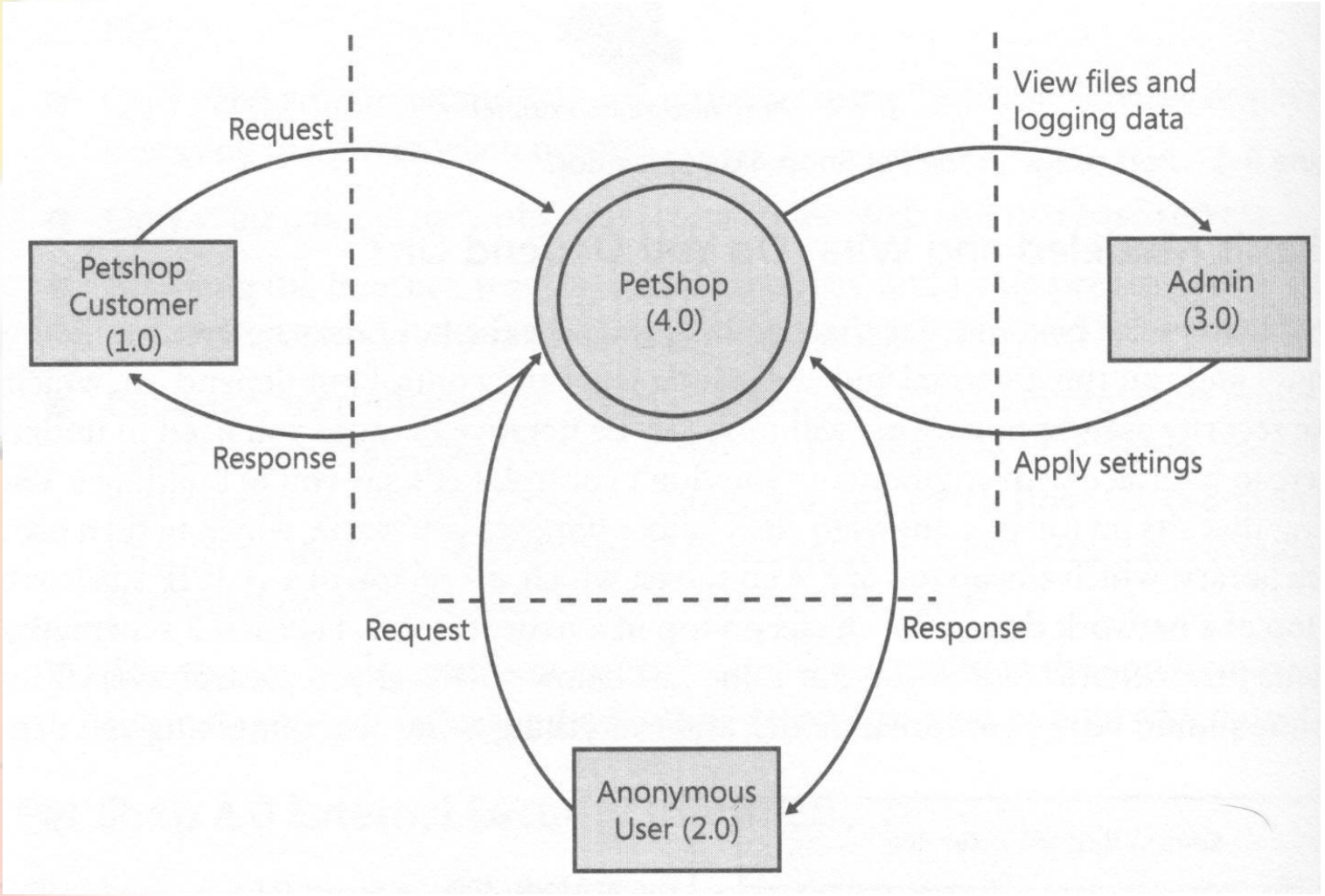


Common DFD “bugs”

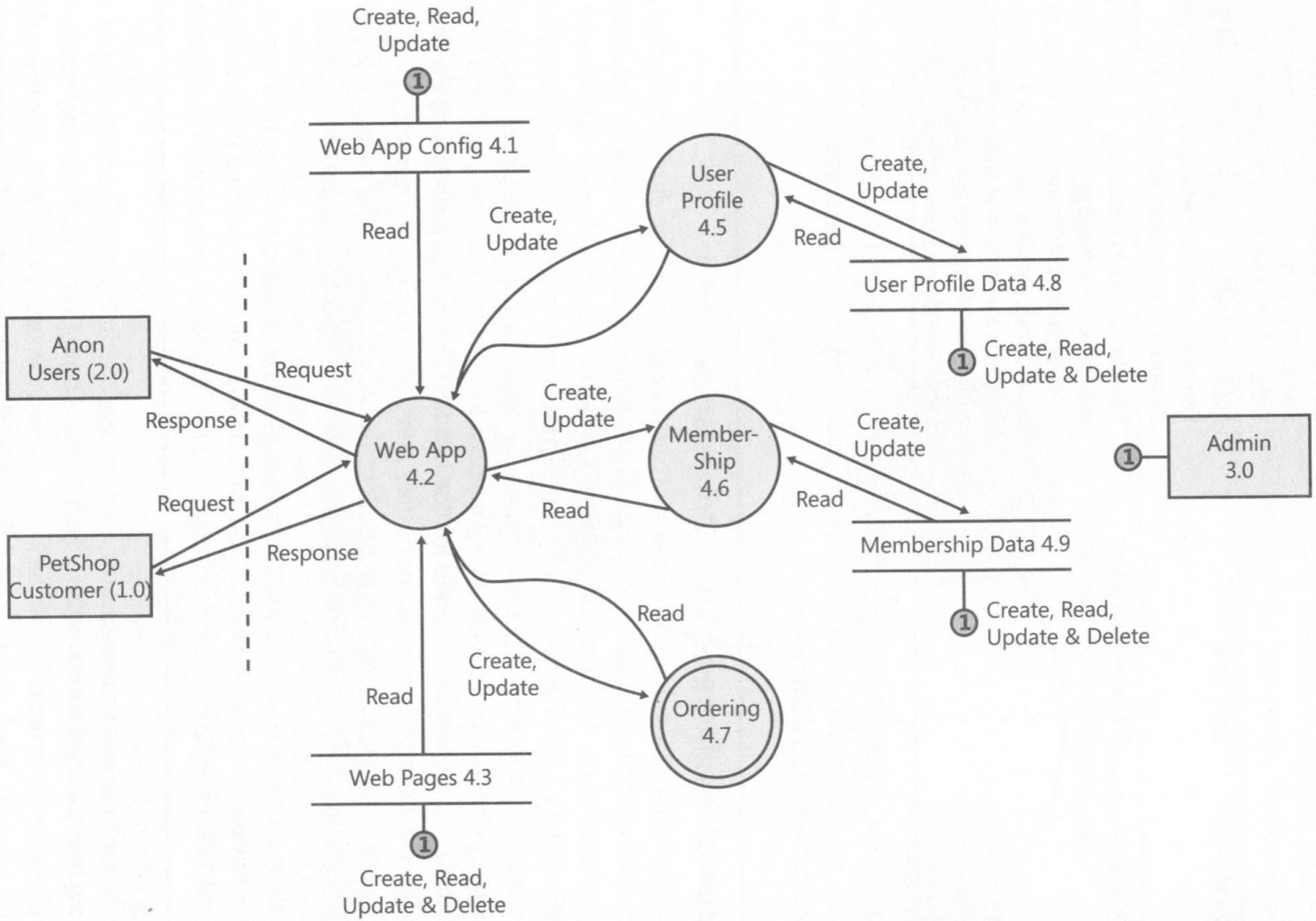
(3) How does data move from a user to a data store?



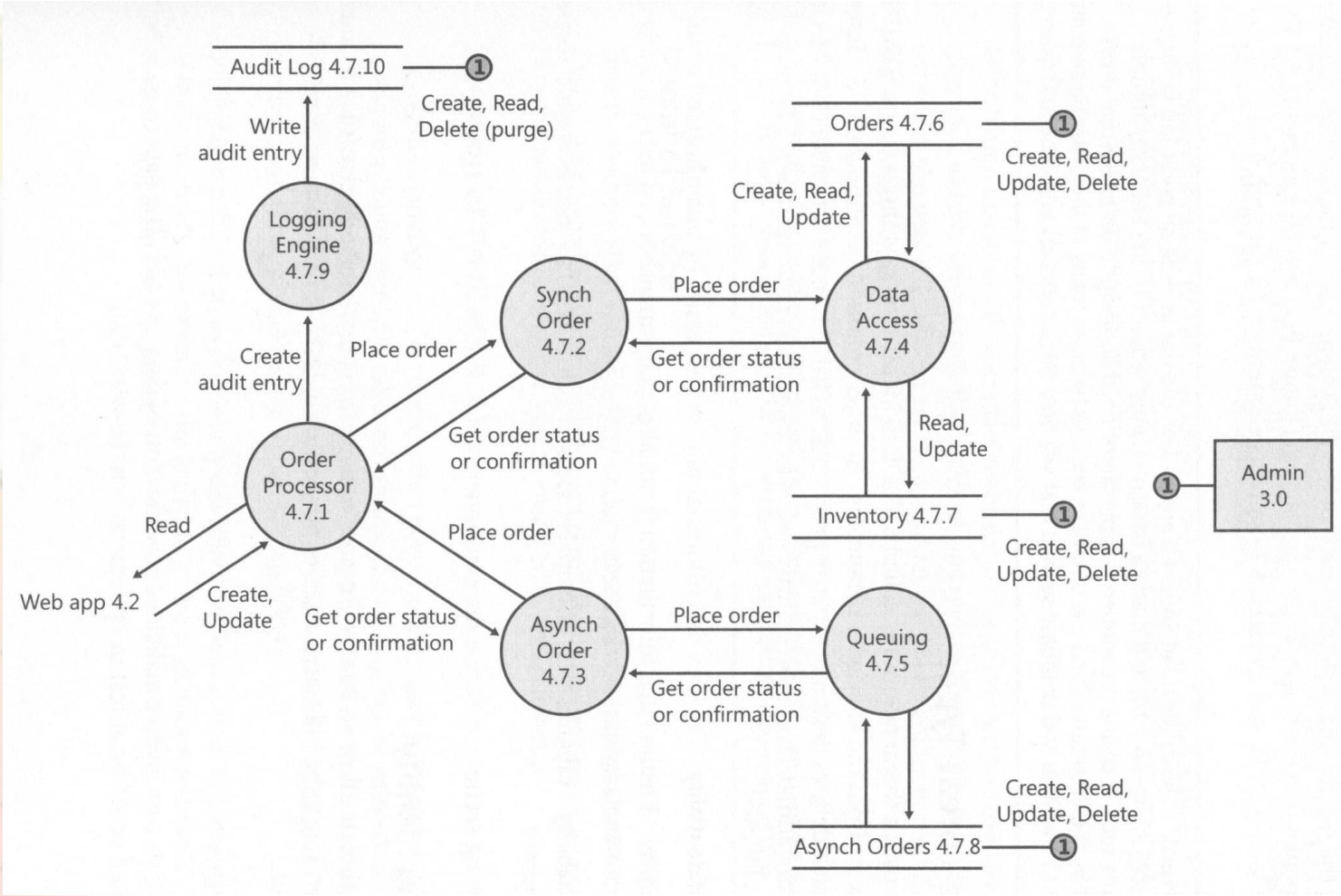
Context Diagram



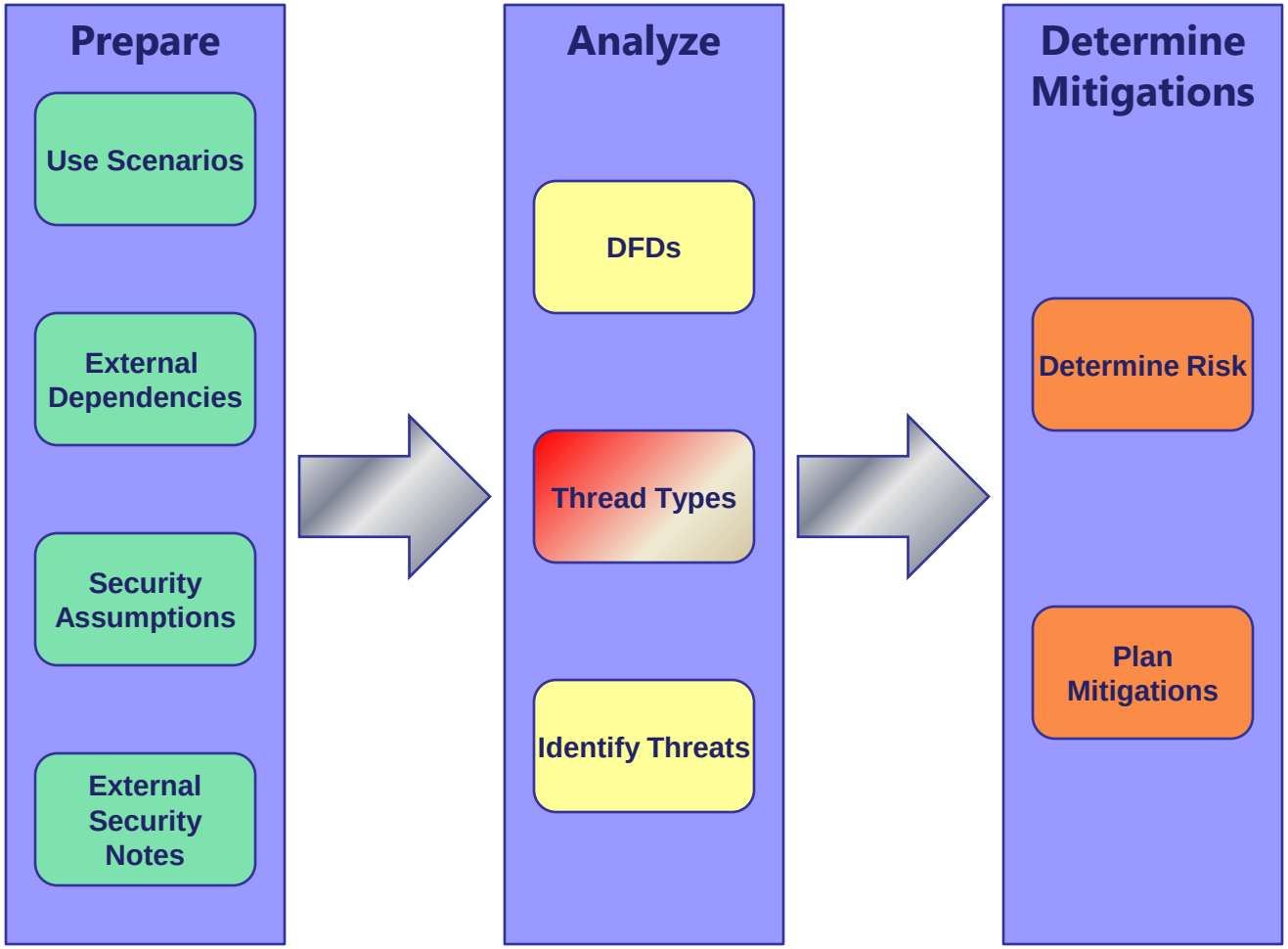
Level-0 DFD


Create, Read, Update
1
Web App Config 4.1
Read
Create, Update
User Profile 4.5
Create, Update
Read
User Profile Data 4.8
1
Create, Read, Update & Delete
Anon Users (2.0)
Request
Response
Web App 4.2
Create, Update
Member-Ship 4.6
Create, Update
Read
Membership Data 4.9
1
Create, Read, Update & Delete
1
Admin 3.0
Request
Response
PetShop Customer (1.0)
Read
Create, Update
Ordering 4.7
Read
Web Pages 4.3
1
Create, Read, Update & Delete

Level-1 DFD


Audit Log 4.7.10
Create, Read, Delete (purge)
Write audit entry
Logging Engine 4.7.9
Create audit entry
Order Processor 4.7.1
Read
Web app 4.2
Create, Update
Place order
Synch Order 4.7.2
Get order status or confirmation
Place order
Get order status or confirmation
Asynch Order 4.7.3
Get order status or confirmation
Place order
Orders 4.7.6
Create, Read, Update, Delete
Create, Read, Update
Data Access 4.7.4
Place order
Get order status or confirmation
Read, Update
Inventory 4.7.7
Create, Read, Update, Delete
Admin 3.0
Queuing 4.7.5
Place order
Get order status or confirmation
Create, Read, Update, Delete
Asynch Orders 4.7.8

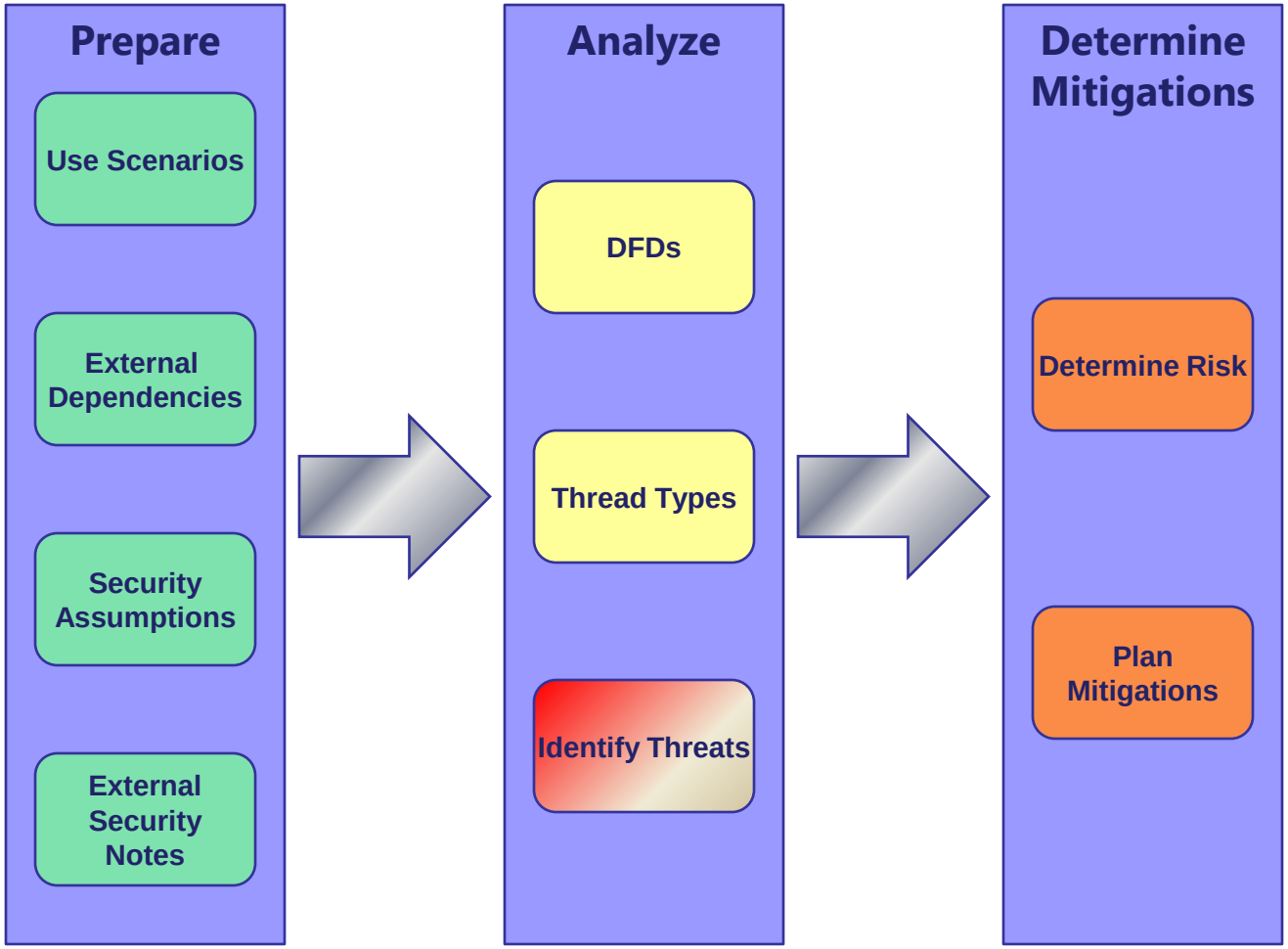
Threat Modeling Process



Categorize Threats

Types of threats	Examples
Spoofing	• Forging e-mail messages • Replaying authentication packets
Tampering	• Altering data during transmission • Changing data in files
Repudiation	• Deleting a critical file and deny it • Purchasing a product and deny it
Information disclosure	• Exposing information in error messages • Exposing code on Web sites
Denial of service	• Flooding a network with SYN packets • Flooding a network with forged ICMP packets
Elevation of privilege	• Exploiting buffer overruns to gain system privileges • Obtaining administrator privileges illegitimately

Threat Modeling Process



Determine Threats

- Enumerating threats creates a threat profile for a system, describing all of the potential attacks that should be mitigated against.
- Threats with valid attack paths are vulnerabilities.
- The security of a system can be expressed in terms of threats with appropriate mitigation vs. total threats, taking into account the severity of the threats with insufficient mitigation (vulnerabilities).

DFD Elements (1)

DFD Element Type	DFD Item Numbers
External Entities	Pet Shop customer (1.0)
	Anonymous user (2.0)
	Administrator (3.0)
Processes	Web application (4.2)
	User profile (4.5)
	Membership (4.6)
	Order processor (4.7.1)
	Synchronous order processor (4.7.2)
	Asynchronous order processor (4.7.3)
	Data access component (4.7.4)
	Queuing component (4.7.5)
	Auditing engine (4.7.9)
Data Stores	Web application configuration data (4.1)
	Web pages (4.3)
	User profile data (4.8)
	Membership data (4.9)
	Orders data (4.7.6)
	Inventory data (4.7.7)
	Asynch orders data (4.7.8)
	Audit-log data (4.7.10)

DFD Elements (2)

DFD Element Type	DFD Item Numbers
Data Flows (partial list for brevity)	Anonymous user request (2.0→4.2)
	Anonymous user response (4.2→2.0)
	Pet Shop customer request (1.0→4.2)
	Pet Shop customer response (4.2→1.0)
	Web application reading configuration data (4.1→4.2)
	Web pages read by Web application (4.3→4.2)
	Admin creating or updating Web application configuration data (3.0→4.1)
	Admin reading Web application configuration data (4.1→3.0)
	Admin creating, updating, or deleting Web pages (3.0→4.3)
	Admin reading Web pages (4.3→3.0)
	Web application creating or updating an order (4.2→4.7.1)
	Web application reading an order (4.7.1→4.2)

DFD Elements Reduction

- First consider bidirectional flows
  - Collapse 2.0->4.2 & 4.2->2.0
    - Data flows use same technology (HTTP over TCP)
    - Share the same process and external entity
    - Data content in either directions public & anonymous

DFD Elements Reduced

DFD Element Type	DFD Item Number
External Entities	Pet Shop customer (1.0)
	Anonymous user (2.0)
	Administrator (3.0)
Processes	Web application (4.2)
	User profile (4.5)
	Membership (4.6)
	Order processor (4.7.1)
	Sync/Async order processors (4.7.2 and 4.7.3)
	Data-access or queuing components (4.7.4 and 4.7.5)
	Auditing engine (4.7.9)
Data Stores	Web application configuration data (4.1)
	Web pages (4.3)
	Use profile data (4.8)
	Membership data (4.9)
	Order and async orders data (4.7.6 and 4.7.8)
	Inventory data (4.7.7)
	Audit-log data (4.7.10)
Data Flows (partial list)	Web application reading configuration data (4.1→4.2)
	Web pages read by Web application (4.3→4.2)
	Anonymous user request/response (2.0→4.2→2.0)
	Pet Shop customer request/response (1.0→4.2→1.0)
	Admin reading, creating, updating Web application configuration data (3.0→4.1→3.0)
	Admin reading, creating, updating, deleting Web pages (3.0→4.3→3.0)
	Web application reading, creating, updating an order (4.2→4.7.1→4.2)

Mapping STRIDE to DFD Elements

DFD Element Type	S	T	R	I	D	E
External Entity	X		X			
Data Flow		X		X	X	
Data Store		X	†	X	X	
Process	X	X	X	X	X	X

Types of threats	Examples
Spoofing	• Forging e-mail messages • Replaying authentication packets
Tampering	• Altering data during transmission • Changing data in files
Repudiation	• Deleting a critical file and deny it • Purchasing a product and deny it
Information disclosure	• Exposing information in error messages • Exposing code on Web sites
Denial of service	• Flooding a network with SYN packets • Flooding a network with forged ICMP Packets
Elevation of privilege	• Exploiting buffer overruns to gain system privileges • Obtaining administrator privileges illegitimately

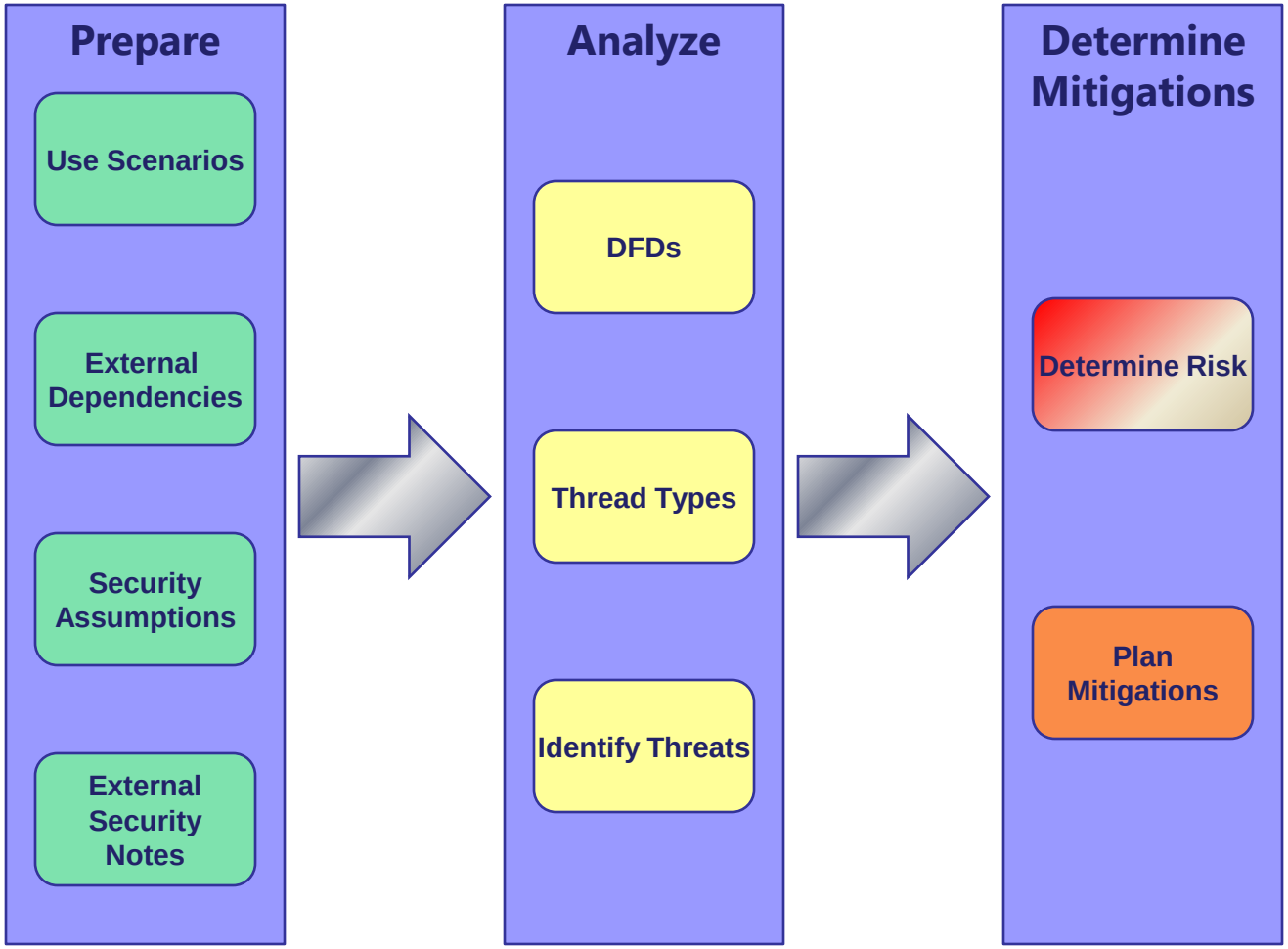
Threats for DVD Elements

DFD Element Type	Threat Types (STRIDE)	DFD Item Numbers
External entities	SR	(1.0), (2.0), (3.0)
Processes	STRIDE	(4.2), (4.5), (4.6), (4.7.1), (4.7.2 and 4.7.3), (4.7.4 and 4.7.5), (4.7.9)
Data stores	T(R)ID	(4.1), (4.3), (4.8), (4.9), (4.7.6 and 4.7.8), (4.7.7), (4.7.10 repudiation)
Data flows (partial list for brevity)	TID	(4.1→4.2), (4.3→4.2), (2.0→4.2→2.0), (1.0→4.2→1.0), (3.0→4.1→3.0), (3.0→4.3→3.0), (4.2→4.7.1→4.2)

Threats to the System

Threat Type (STRIDE)	DFD Item Numbers
Spoofing	External entities: (1.0), (2.0), (3.0) Processes: (4.2), (4.5), (4.6), (4.7.1), (4.7.2 and 4.7.3), (4.7.4 and 4.7.5), (4.7.9)
Tampering	Processes: (4.2), (4.5), (4.6), (4.7.1), (4.7.2 and 4.7.3), (4.7.4 and 4.7.5), (4.7.9) Data stores: (4.1), (4.3), (4.8), (4.9), (4.7.6 and 4.7.8), (4.7.7), (4.7.10) Data flows: (4.1→4.2), (4.3→4.2), (2.0→4.2→2.0), (1.0→4.2→1.0), (3.0→4.1→3.0), (3.0→4.3→3.0), (4.2→4.7.1→4.2)
Repudiation	External entities: (1.0), (2.0), (3.0) Data flow: (4.7.10)
Information disclosure	Processes: (4.2), (4.5), (4.6), (4.7.1), (4.7.2 and 4.7.3), (4.7.4 and 4.7.5), (4.7.9) Data stores: (4.1), (4.3), (4.8), (4.9), (4.7.6 and 4.7.8), (4.7.7), (4.7.10) Data flows: (4.1→4.2), (4.3→4.2), (2.0→4.2→2.0), (1.0→4.2→1.0), (3.0→4.1→3.0), (3.0→4.3→3.0), (4.2→4.7.1→4.2)

Threat Modeling Process



Prioritize Threats based on Risk

- Asset:
  - Internal mailbox of your Managing Director
- Risk Impact Estimate (examples!)
  - Risk of loss: Medium impact
  - Risk of access by staff: High impact
  - Risk of access by press: Catastrophic impact
  - Risk of access by a competitor: High impact
  - Risk of temporary no access by MD: Low impact
  - Risk of change of content: Medium impact

„Risk Level“ – Threat Characteristics

- Server application vs. client application
- Local vs. Remote accessibility
- Accessibility to anonymous versus authenticated users
- Accessibility to authenticated users vs. Administrators
- On by default vs. Off by default
- The degree of user interaction required
- In case of an information disclosure; PII or sensitive PII
- In case of a DOS attack; will service continue when attack stops?

„Risk Level“

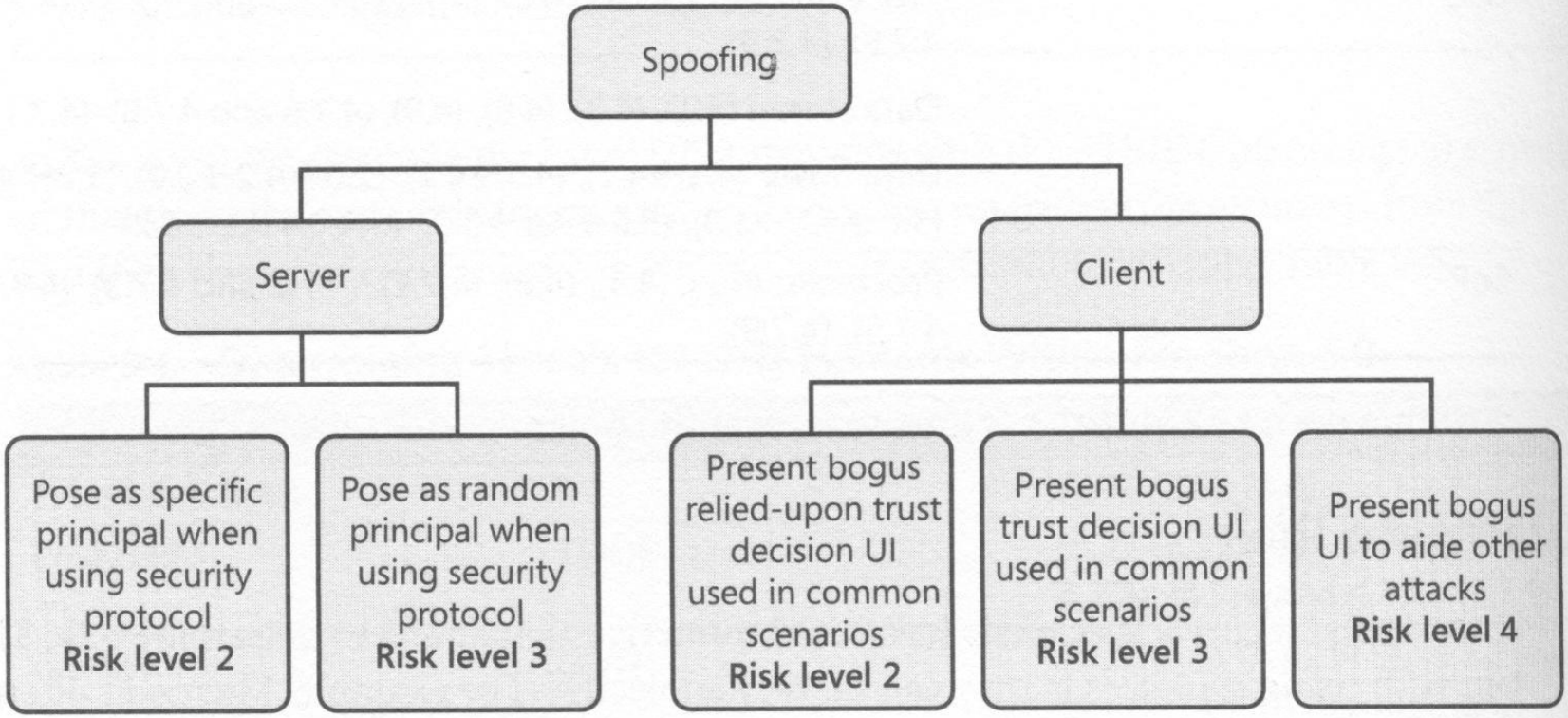


Figure 9-6 Spoofing threats risk ranking.

„Risk Level“

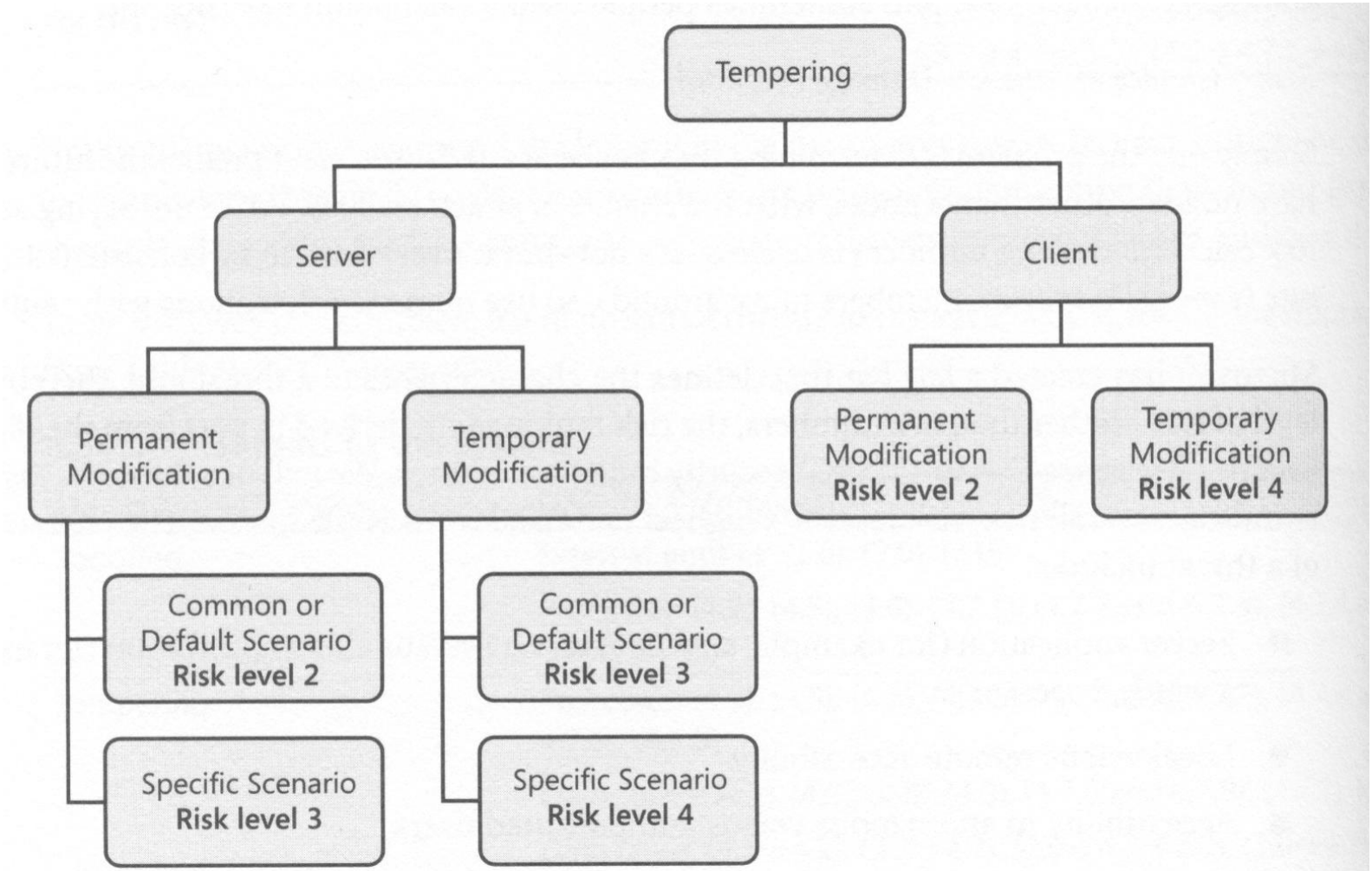


Figure 9-7 Tampering threats risk ranking.

„Risk Level“

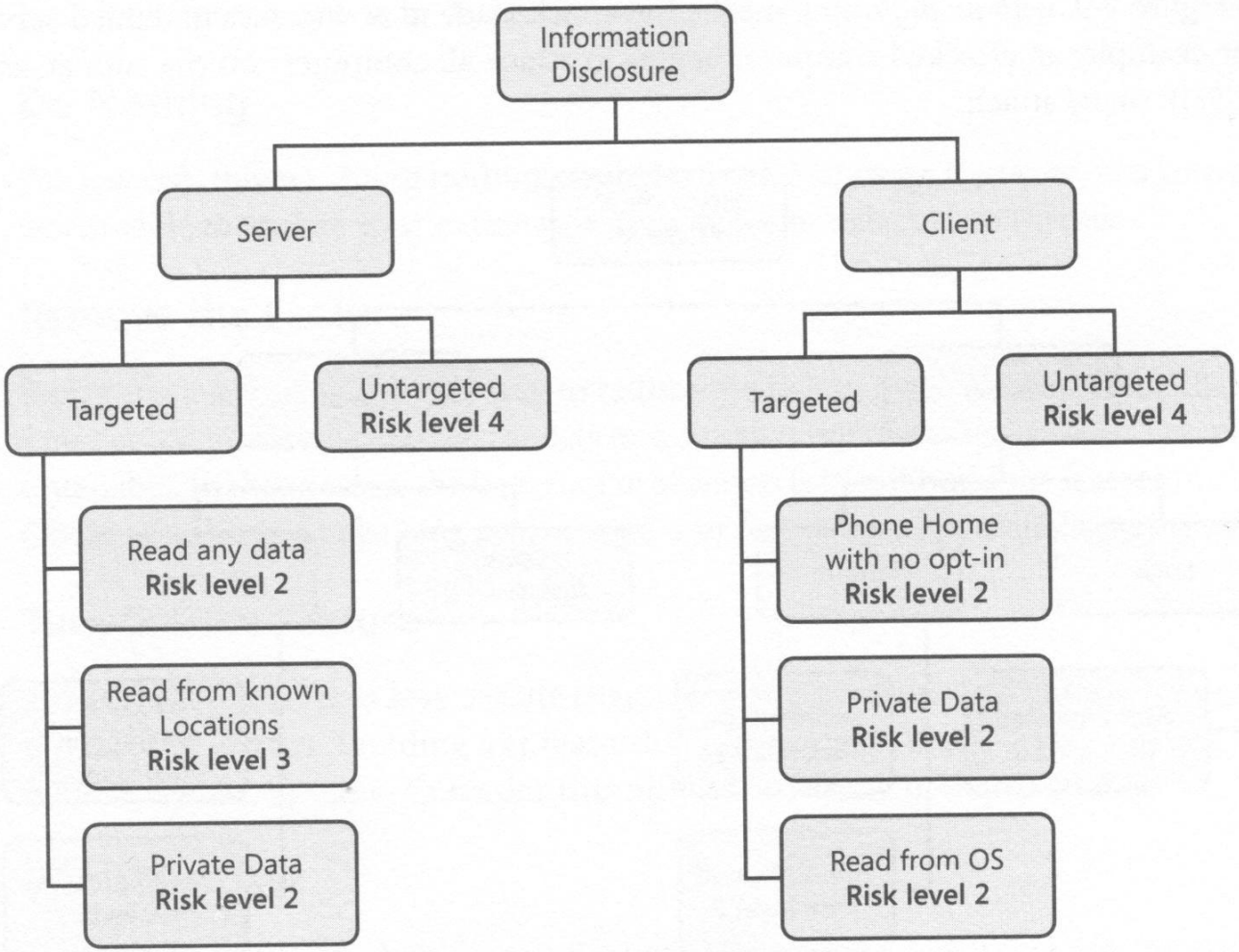


Figure 9-8 Information disclosure threats risk ranking.

„Risk Level“

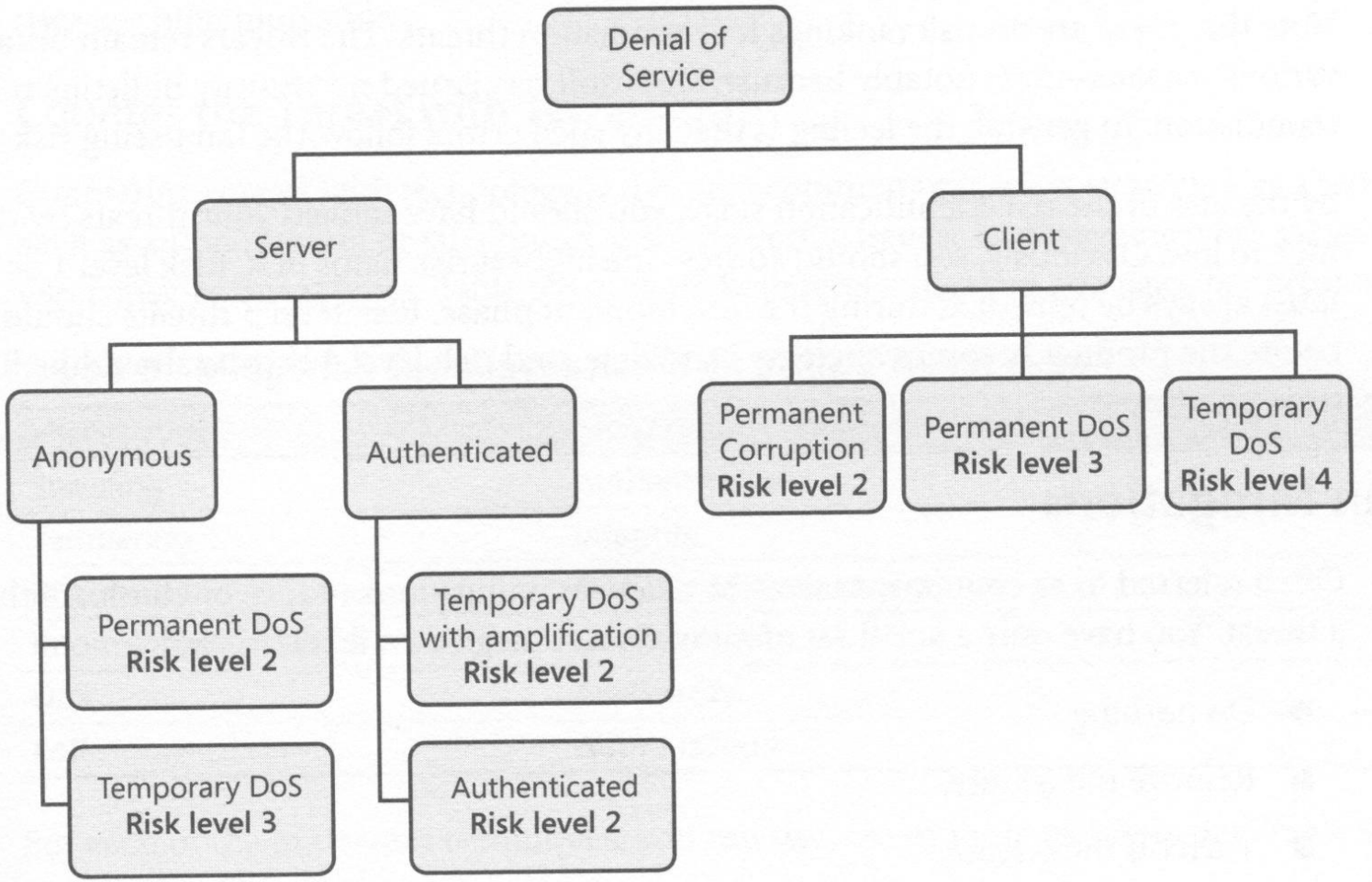


Figure 9-9 DoS threats risk ranking.

„Risk Level“

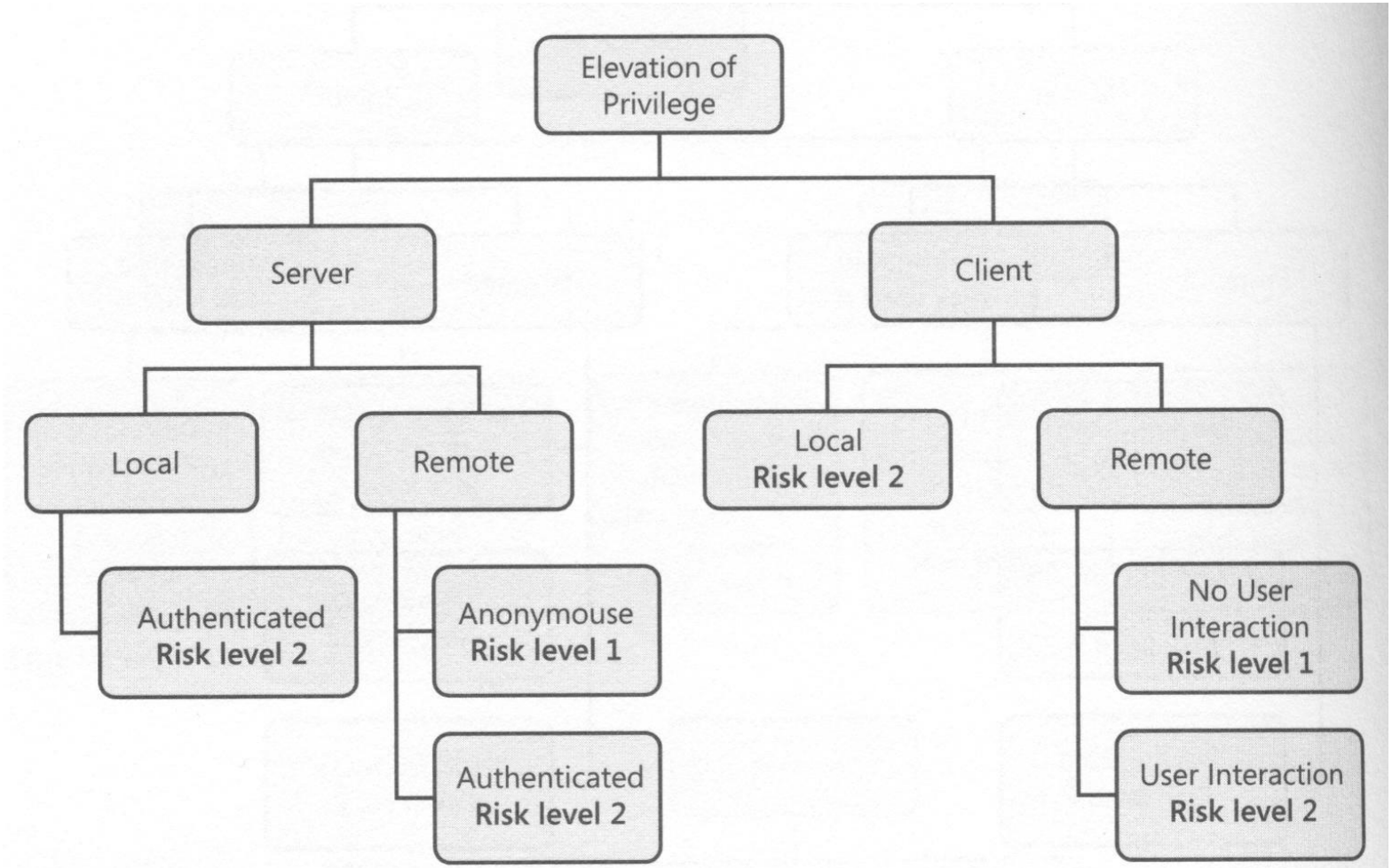
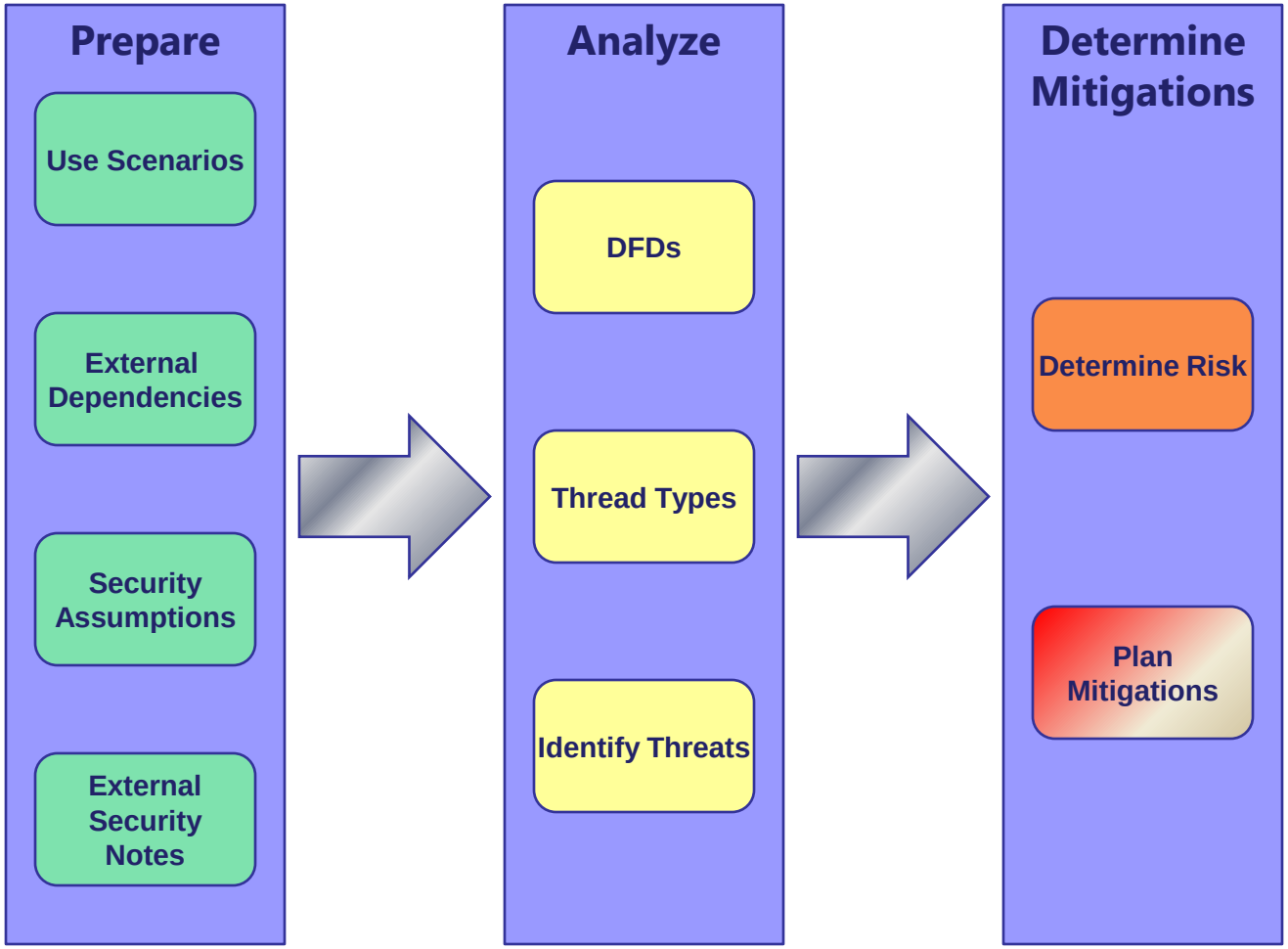


Figure 9-10 EoP threats risk ranking.

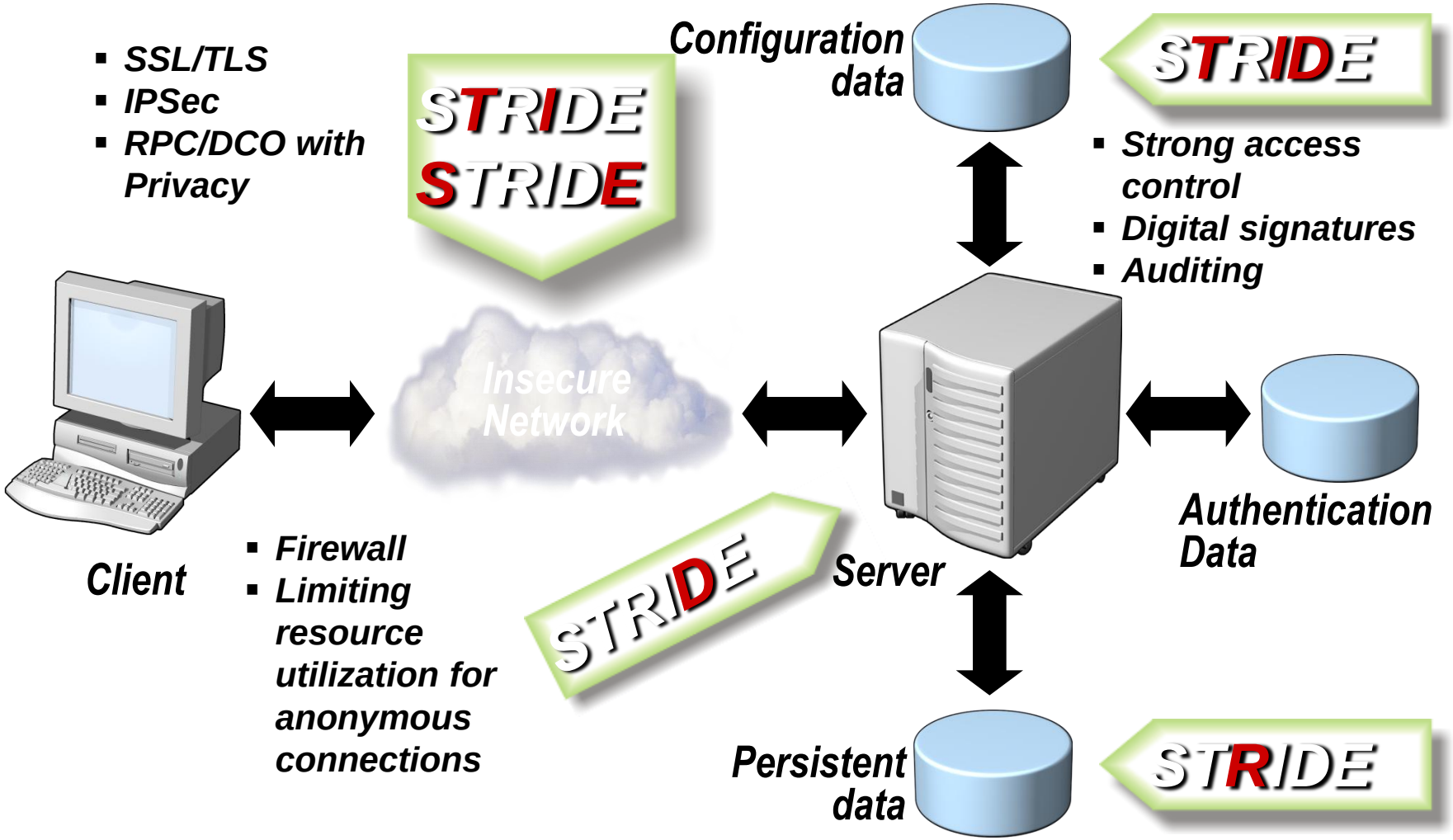
Threat Modeling Process



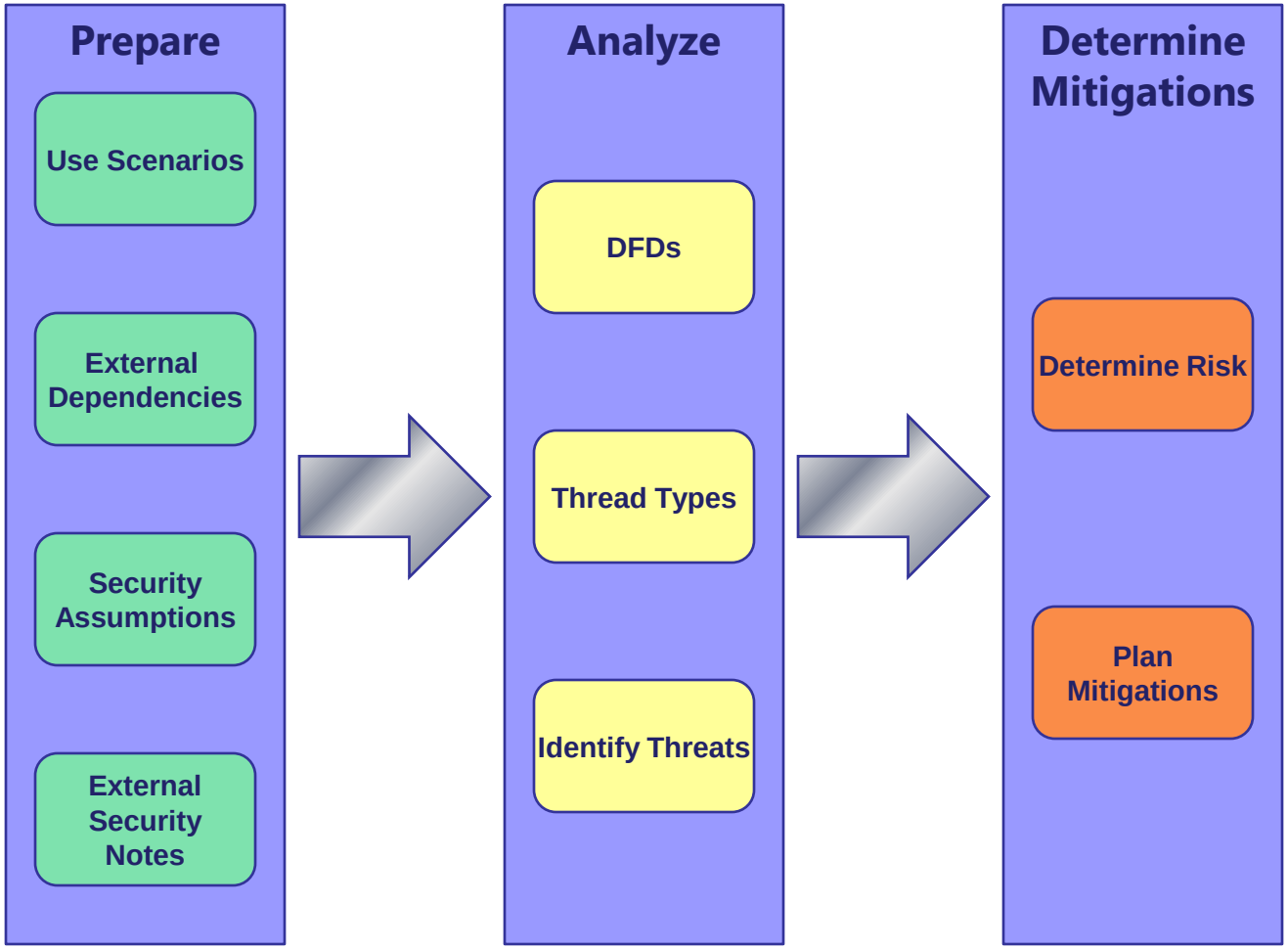
Plan Mitigation

- Do nothing
- Remove the Feature
- Turn off the Feature
- Warn the User
- Counter the threat with Technology

Select Mitigation Technique



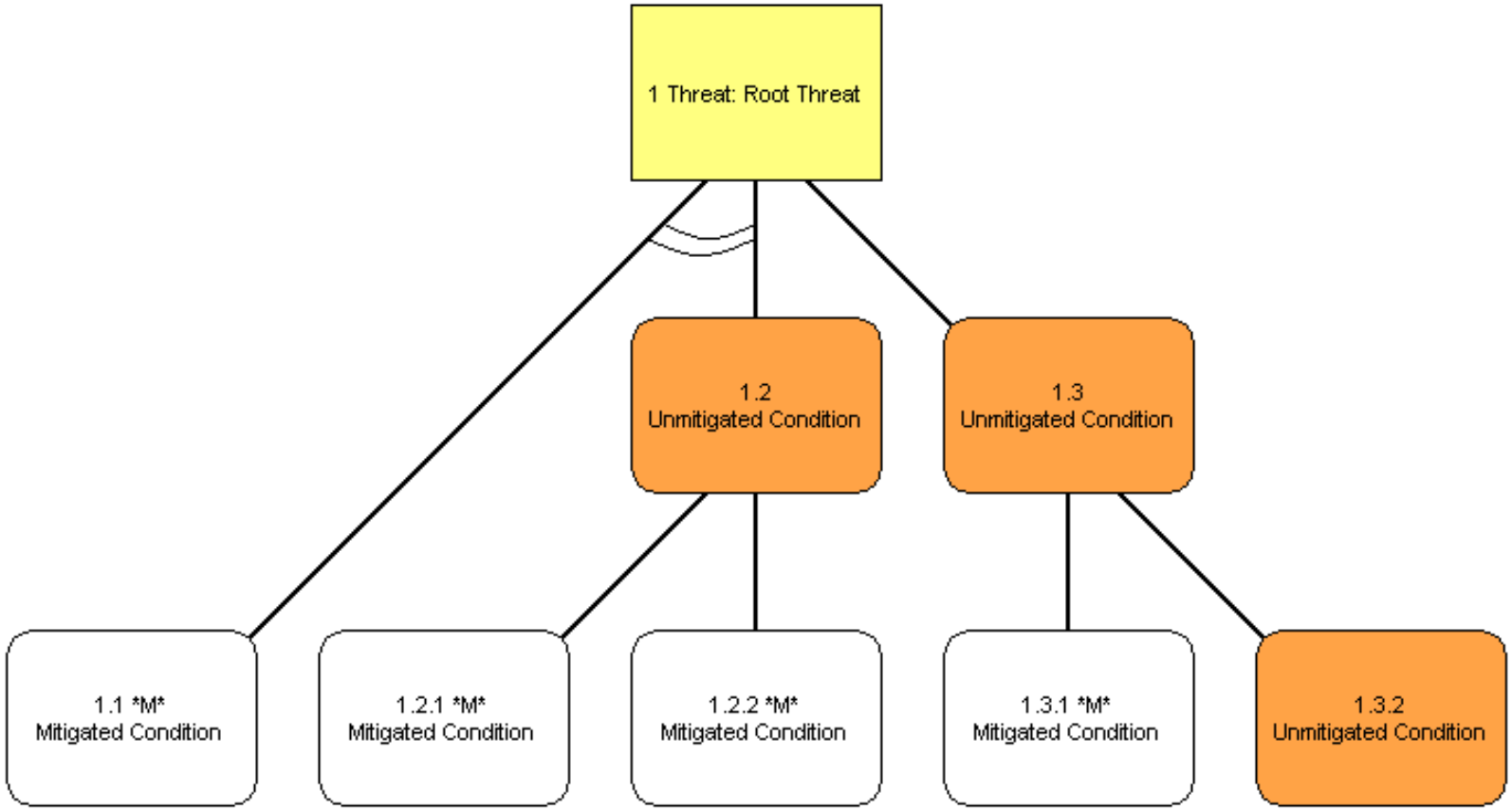
Threat Modeling Process



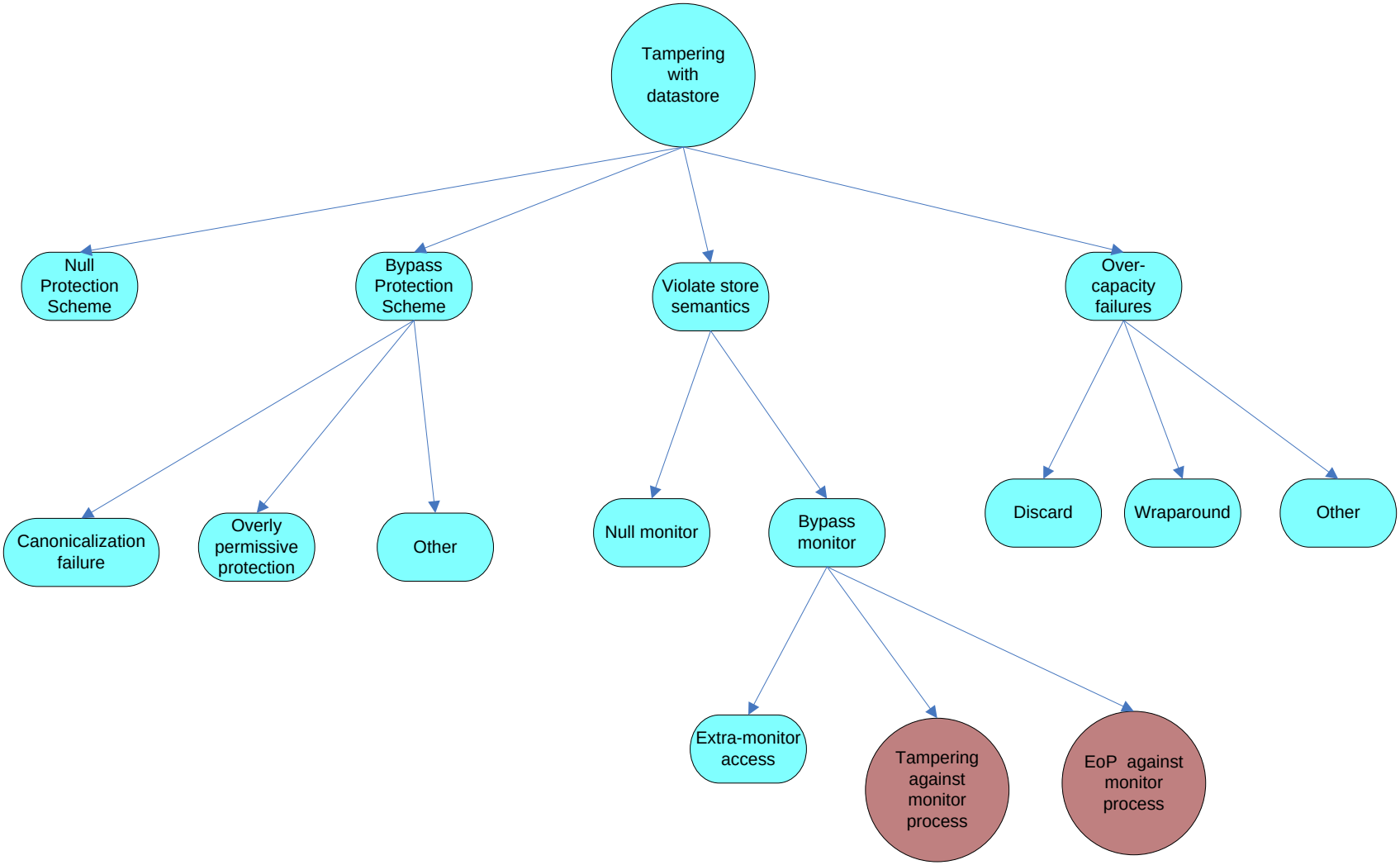
Threat Trees

- Threat Trees (also called Attack Trees: Bruce Schneier, Dr. Dobb's Journal December 1999, "Modeling Security Threats") are used in Threat Modeling to analyze how a threat might be accomplished. It is a hierarchical representation of conditions, with the root node being the threat. An *attack path* is a route from a leaf condition to the root threat, inclusive of any *and* condition. Threat Trees are used to determine valid attack paths for a threat. That is, any attack path that does not have a mitigating node is classified as a vulnerability.
- In its most basic form, a Threat Tree consists of a single Threat, and multiple Mitigated Conditions and Unmitigated Conditions.

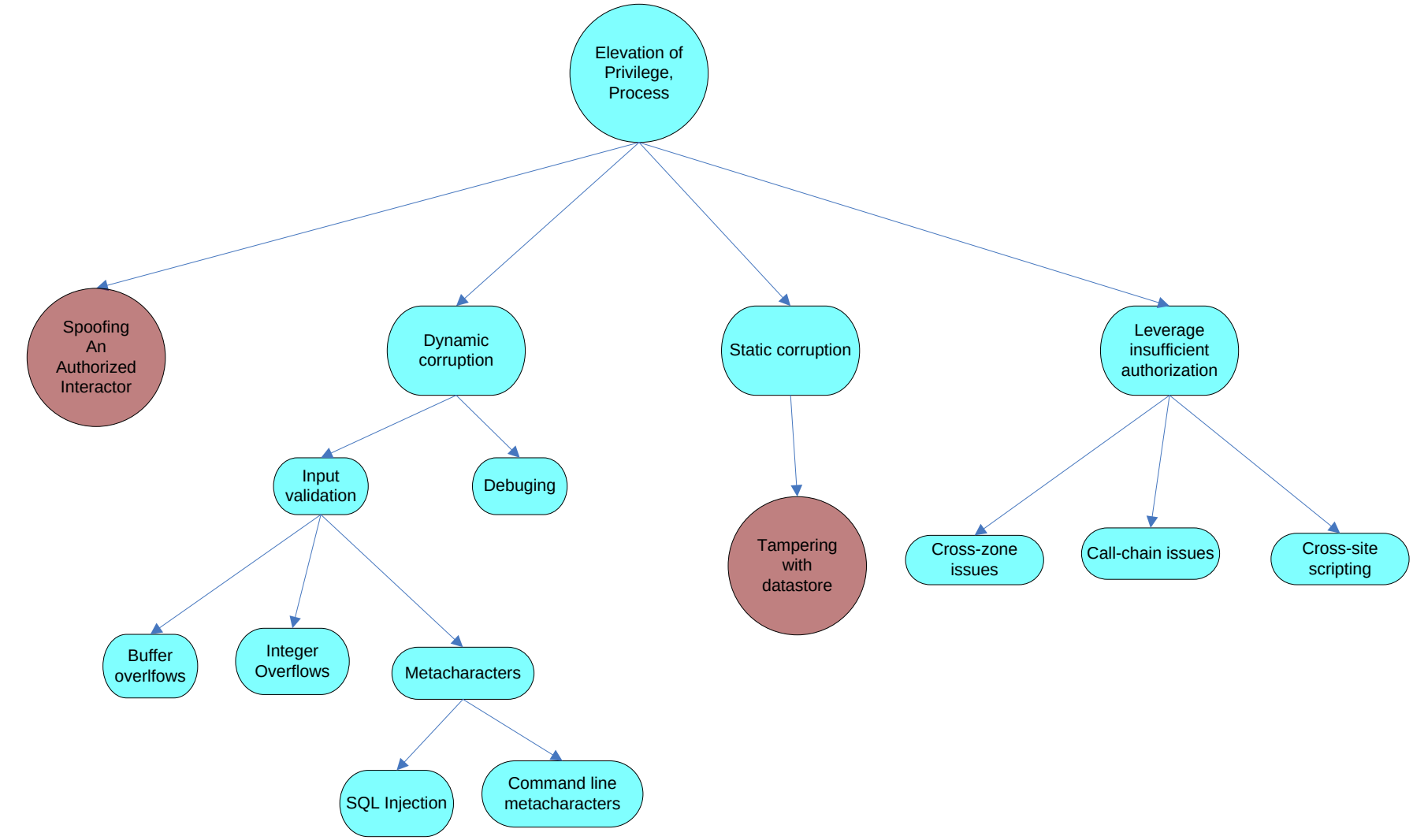
Threat Trees



Threat Tree – Tampering with Data store



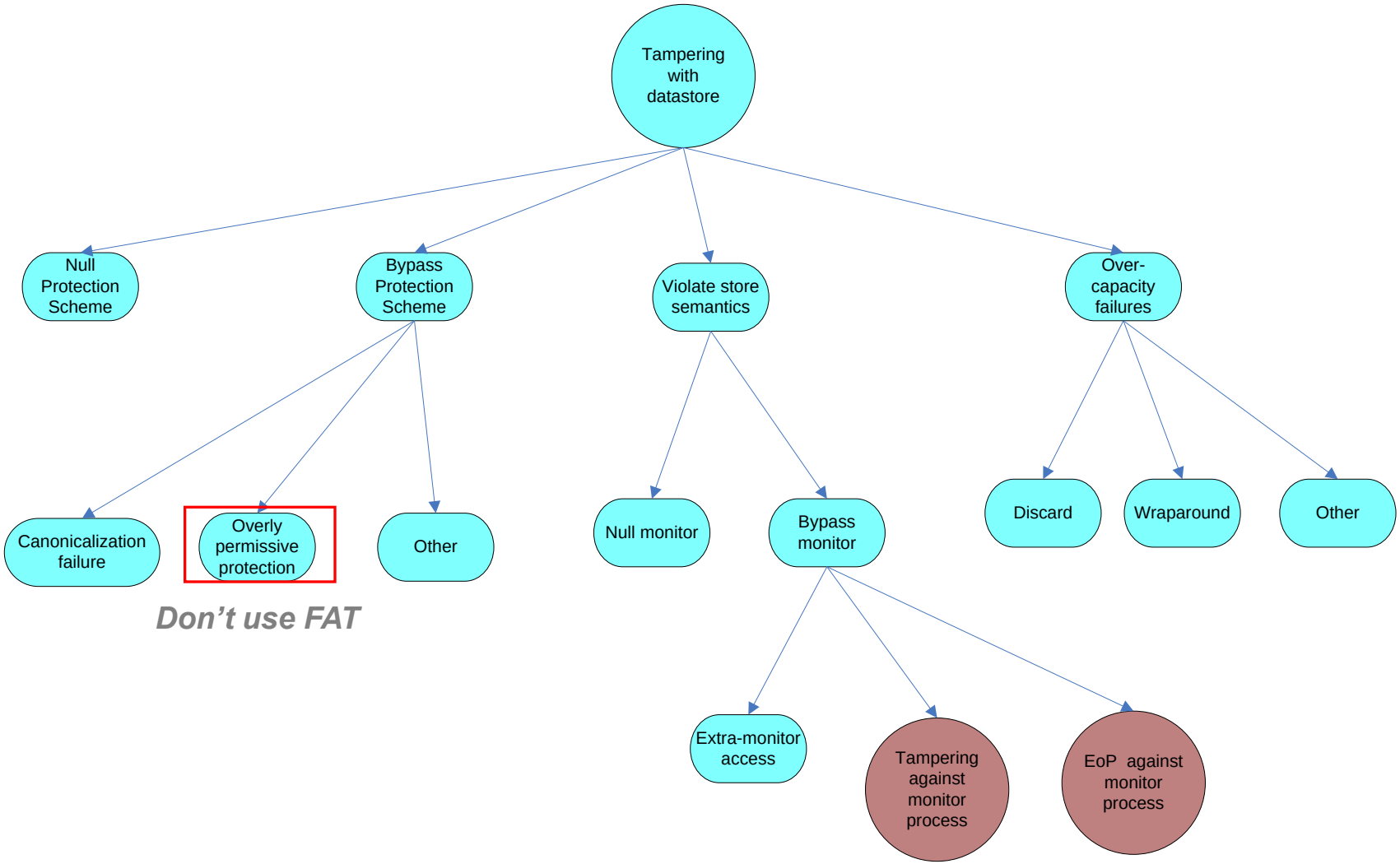
Threat Tree Elevation of Privilege



“Best Practice” Mitigations

- Mitigate condition as high in the tree as possible
- Mitigate across one side of an AND-clause
- Application defenses are better than infrastructure defenses

“Best Practice” Mitigations



Don’t use FAT

Readings

(Torr 2005) Torr, Peter. “Guerilla Threat Modelling (or ‘Threat Modeling’ if you’re American),” http://blogs.msdn.com/ptorr/archive/2005/02/22/GuerillaThreatModelling.aspx. February 2005.

(Howard and Lipner 2006) Howard, Michael, and Steve Lipner. *The Security Development LIFECYCLE*. Redmond, WA: Microsoft Press, 2006.

(Pet Shop 2006) Leake, Gregory, Microsoft Corporation. “Microsoft .NET Pet Shop 4: Migrating an ASP.NET 1.1 Application to 2.0,” http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/bdasamppet4.asp. MSDN, February 2006.