Software Security & Privacy

Reference Book:
Software Engineering, Ian Sommerville
10th / Global Edition, Chapter 8
Pearson Publishers

Why is Software Security? HeartBleed Bug

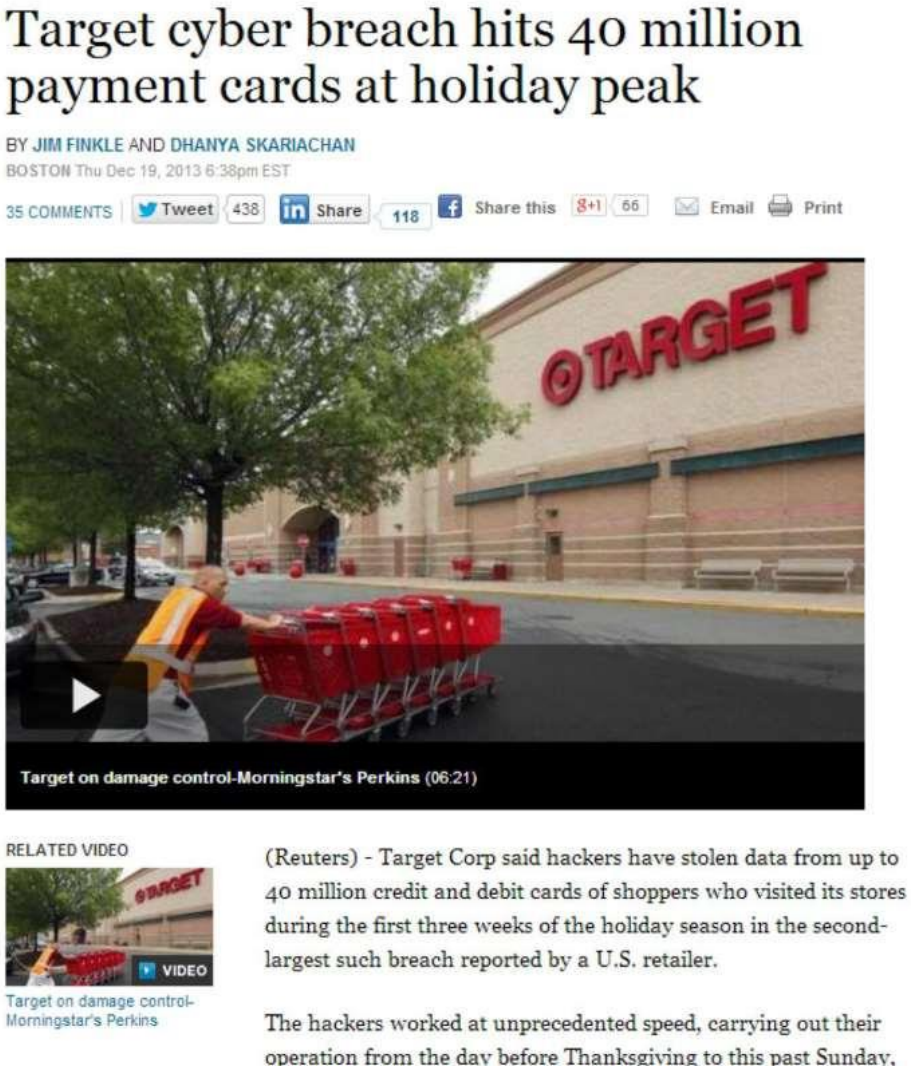
- Buffer overflow bug in OpenSSL cryptographic library
  - ✓ Allows attacker to read server private keys
- 17% of “secure” Internet servers worldwide estimated to be vulnerable (500 Million servers!!)
- Simple patch, but huge cost of patching all these servers

The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet. SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging (IM) and some virtual private networks (VPNs).

The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.

Source: Heartbleed.com

Why is Software Security? Target Credit-Card Attack



Target cyber breach hits 40 million payment cards at holiday peak

BY JIM FINKLE AND DHANYA SKARIACHAN

BOSTON Thu Dec 19, 2013 6:38pm EST

35 COMMENTS | Tweet 438 | Share 118 | Share this | 66 | Email | Print

Target on damage control-Morningstar's Perkins (06:21)

RELATED VIDEO

Target on damage control-Morningstar's Perkins

(Reuters) - Target Corp said hackers have stolen data from up to 40 million credit and debit cards of shoppers who visited its stores during the first three weeks of the holiday season in the second-largest such breach reported by a U.S. retailer.

The hackers worked at unprecedented speed, carrying out their operation from the day before Thanksgiving to this past Sunday,

- 40 Million CC numbers stolen
- 90 lawsuits filed against Target
- Target spent $61 Million responding to breach
- Target profits fell in holiday period by 46%

Why is Software Security? Attack on Bitcoin Exchange

Bitcoin: $64m in cryptocurrency stolen in 'sophisticated' hack, exchange says

Mining marketplace NiceHash suspends operations while it co-operates with authorities over 'professional attack', urging users to change passwords

Software

Software: A program consists of data and instructions that manipulate the data.

Examples of Software:

- Operating systems
- Stand alone applications
- Web applications (Server Side: Gmail, Amazon, online Banking
  Client Side: Internet Explorer, Chrome, Firefox)
- Mobile applications
- Embedded Software (Built into special purpose hardware platform)
  - ✓ Smart / Chip cards (Bank debit / credit cards), Cars, TVs, PlayStation, Cameras
- Software components
  - ✓ Libraries, e.g., STL, OpenSSL
  - ✓ Frameworks, e.g., Apache Thrift, Tensorflow

Software Security

Software Security is about the **secure design** and **implementation** of software.

Focus of the Study:

- the code (secure implementation)
- the design (threat modelling)
- the workflow (secure software development life cycle)

Software Security: Gap

- **Developers are concerned with correctness**
  - ✓ Software operates as intended, achieving desired behavior
- **Security is concerned with preventing undesired behavior**
  - ✓ Considers an enemy / opponent / hacker / adversary who is maliciously trying to circumvent any protective measures you put in place

Undesired Behavior:

- Stealing Information
  - ✓ Corporate Secrets (product plan, source code, IP….)
  - ✓ Personal Information (health record, credit card number, address….)
- Modifying Information or Functionality
  - ✓ Destroying records (accounts, logs, files….)
  - ✓ Installing unwanted software (spyware, botnet client, ransomware….)
- Denying Access
  - ✓ Unable to access a website, database, cloud drive…

Software Security: CIA Triad

Three Dimensions of Security:

- **Confidentiality:** concealment of information or resources.
  - Prevent the disclosure of sensitive information from unauthorized people, resources, and processes.
  - Access control mechanisms and Resource hiding supports confidentially.
- **Availability:** the ability to use the information or resources desired.
  - The assurance that systems and data are accessible by authorized users when needed.
  - Availability is an important aspect of reliability.
  - Attempts to block availability, called DOS attack is the most difficult to detect, if unusual access patterns are attributable to deliberate manipulation of resources of environment.

Hacking is a driving force of security.

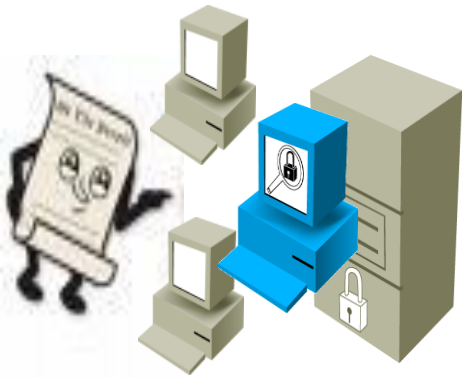
Software Security: CIA Triad

- **Integrity:** Trustworthiness of data and resources.
  - The protection of system information or processes from intentional or accidental modification.
  - Integrity includes data integrity (the contents of information) and origin integrity (the source of data often called authentication). The source of information may bear on its accuracy and credibility.
  - Integrity mechanisms fall into two classes: Prevention Mechanism and detection mechanism (report that the data integrity is no longer trustworthy).
    - Prevention blocks any unauthorized attempts to change the data (lack of authentication) or any attempts to change the data in unauthorized way (lack of authorization)

- **Hackers**
  - Negative – **Black** Hat Hacker
  - Positive – **White** Hat Hacker
    - **Gray** Hat Hacker

IT Security Policy

- A document that states how an organization plans to protect its tangible and intangible information assets
  - Management instructions indicating a course of action, a guiding principle, or appropriate procedure
  - High-level statements that provide guidance to workers who must make present and future decisions
  - Generalized requirements that must be written down and communicated to others



- Software security policy outlines:
  - Rule of software access: Establishes a hierarchy of access permissions.
  - How policies are enforced
  - Describes the basic architecture of the organization's software security environment

Documents Supporting Security Policies

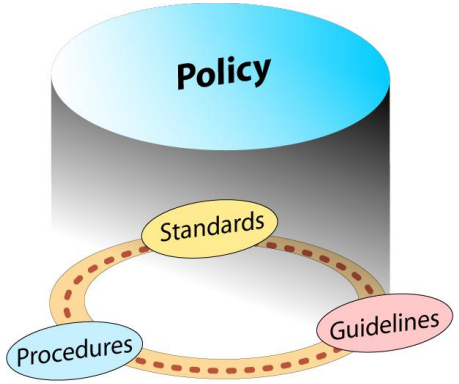
- Standards – dictate specific minimum requirements in our policies
- Guidelines – suggest the best way to accomplish certain tasks
- Procedures – provide a method by which a policy is accomplished (the instructions)

Subsection	6.1 PERSONNEL SECURITY Change Control #: 1.0
Policy	6.1.3 Confidentiality Agreements Approved by: SMH
Objectives	Confidentiality of organizational data is a key tenet of our information security program. In support of this goal, ABC Co will require signed confidentiality agreements of all authorized users of information systems. This agreement shall conform to all federal, state, regulatory, and union requirements.
Purpose	The purpose of this policy is to protect the assets of the organization by clearly informing staff of their roles and responsibilities for keeping the organization's information confidential.
Audience	ABC Co confidentiality agreement policy applies equally to all individuals granted access privileges to an ABC Co Information resources
Policy	This policy requires that staff sign a confidentiality policy agreement prior to being granted access to any sensitive information or systems. Agreements will be reviewed with the staff member when there is any change to the employment or contract, or prior to leaving the organization. The agreements will be provided to the employees by the Human Resource Dept.
Exceptions	At the discretion of the Information Security Officer, third parties whose contracts include a confidentiality clause may be exempted from signing individual confidentiality agreements.
Disciplinary Actions	Violation of this policy may result in disciplinary actions, which may include termination for employees and temporaries; a termination of employment relations in the case of contractors or consultants; or dismissal for interns and volunteers. Additionally, individuals are subject to civil and criminal prosecution.

Example: Policy for Password Use

Policy:

- All users must have a unique user ID and password that conforms to the company password standard
- Users must not share their password with anyone regardless of title or position
- Passwords must not be stored in written or any readable form
- If a compromise is suspected, it must be reported to the help desk and a new password must be requested



Standards:

- Minimum of 8 upper- and lowercase alphanumeric characters
- Must include a special character
- Must be changed every 30 days
- Password history of 24 previous passwords will be used to ensure passwords aren't reused

Example: Policy for Password Use

The Guideline:

- Take a phrase
  Up and At 'em at 7!
- Convert to a strong password
  Up&atm@7!
- To create other passwords from this phrase, change the number, move the symbol, or change the punctuation mark

Procedure for changing a password

1. Press Control, Alt, Delete to bring up the log in dialog box
2. Click the "change password" button
3. Enter your current password in the top box
4. …

The OSI Security Architecture

- **Security Attack:** An action that compromises the security of information owned by an organization.
- **Security Mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent or recover from a security attack.
- **Security Service:** A processing or communication service that enhances the security of the data processing systems and the information transfer of an organization.

Threat and Attack (RFC 2828)

- **Threat:** A potential for violation of security, which exits when there is a circumstance, capability, action, or event that could breach security and cause harm.
- **Attack:** An assault on system security that derives from an intelligent threat.

  i.e., an intelligent act that is a deliverable attempt to evade security services and violet the security policy of a system

Threats

Threats: Potential violation of security.

Attacks: Actions that could cause violation of security.

Threats can be divided into four broad classes:

- **Disclosure:** unauthorized access of information.
- **Deception:** acceptance of false data.
- **Disruption:** Interruption or prevention of correct operation.
- **Usurpation:** Unauthorized control of some part of a system.

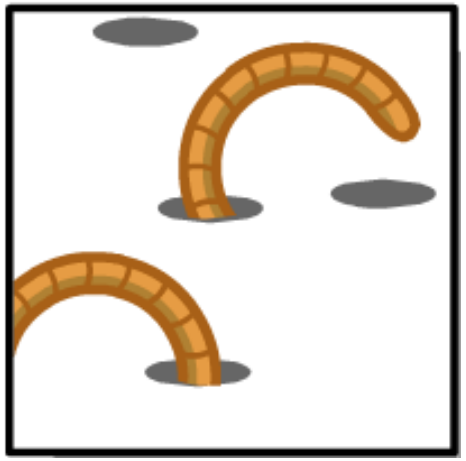
Threats

Some Important threats are:

- **Snooping:** unauthorized interception of information.
  - ✓ Passive wiretapping: Listening to communications or browsing files or system information.
  - ✓ Active wiretapping: Modification or alteration of information, e.g., the man-in the middle attack.
- **Masquerading or spoofing:** Impersonation of one entity by another. Delegation occurs when one entity authorizes a second entity to perform functions on its behalf. Masquerading is a violation of security whereas delegation is not.
  - ✓Passive masquerading: does not attempt to authenticate the recipient but merely accesses it.
  - ✓ Active masquerading: Masquerader issues response to mislead the user about its identity.
- **Repudiation of origin:** A false denial that an entity sent (or created) something.
- **Denial of receipt:** a false denial that an entity received some information or message.
- **Delay:** a temporary inhibition of a service. This requires manipulation of system control structures, such as network components or server components.
- **Denial of Service (DoS):** a long term inhibition of a service. This an infinite delay.

Security Attacks

- **Virus:** A malicious software which attaches to another program to execute a specific unwanted function on a computer.

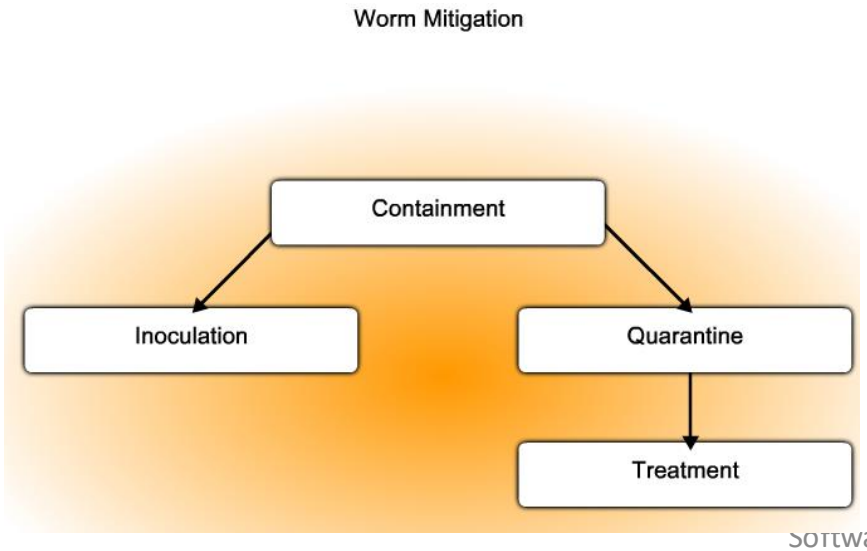


- **Worm:** executes arbitrary code and installs copies of itself in the memory of the infected computer, which then infects other hosts.

- **Trojan Horse:** An application written to look like something else. When a Trojan Horse is downloaded and opened, it attacks the end-user computer from within.

Security Attacks: Worm Attack

- Three major components to most worm attacks:
  - **Enabling vulnerability** - A worm installs itself using an exploit mechanism (email attachment, executable file, Trojan Horse) on a vulnerable system.
  - **Propagation mechanism** - After gaining access to a device, the worm replicates itself and locates new targets.
  - **Payload** - Any malicious code that results in some action. Most often this is used to create a backdoor to the infected host.



- Response to a worm infection:
  - **Containment** - A policy for checking the expansion of worm to other files or devices.
  - **Inoculation-** Increase computer's Immunity.
  - **Quarantine-** Separate infected files.
  - **Treatment-** disinfect the worm from the files.

Security Attacks: Trojan Horse

Classification of Trojan horse:

- **Remote-access** Trojan Horse (enables unauthorized remote access)
- **Data sending** Trojan Horse (provides the attacker with sensitive data such as passwords)
- **Destructive** Trojan Horse (corrupts or deletes files)
- **Proxy** Trojan Horse (user's computer functions as a proxy server)
- **FTP** Trojan Horse (opens port 21)
- **Security software disabler** Trojan Horse (stops anti-virus programs or firewalls from functioning)
- **Denial of Service** Trojan Horse (slows or halts network activity)

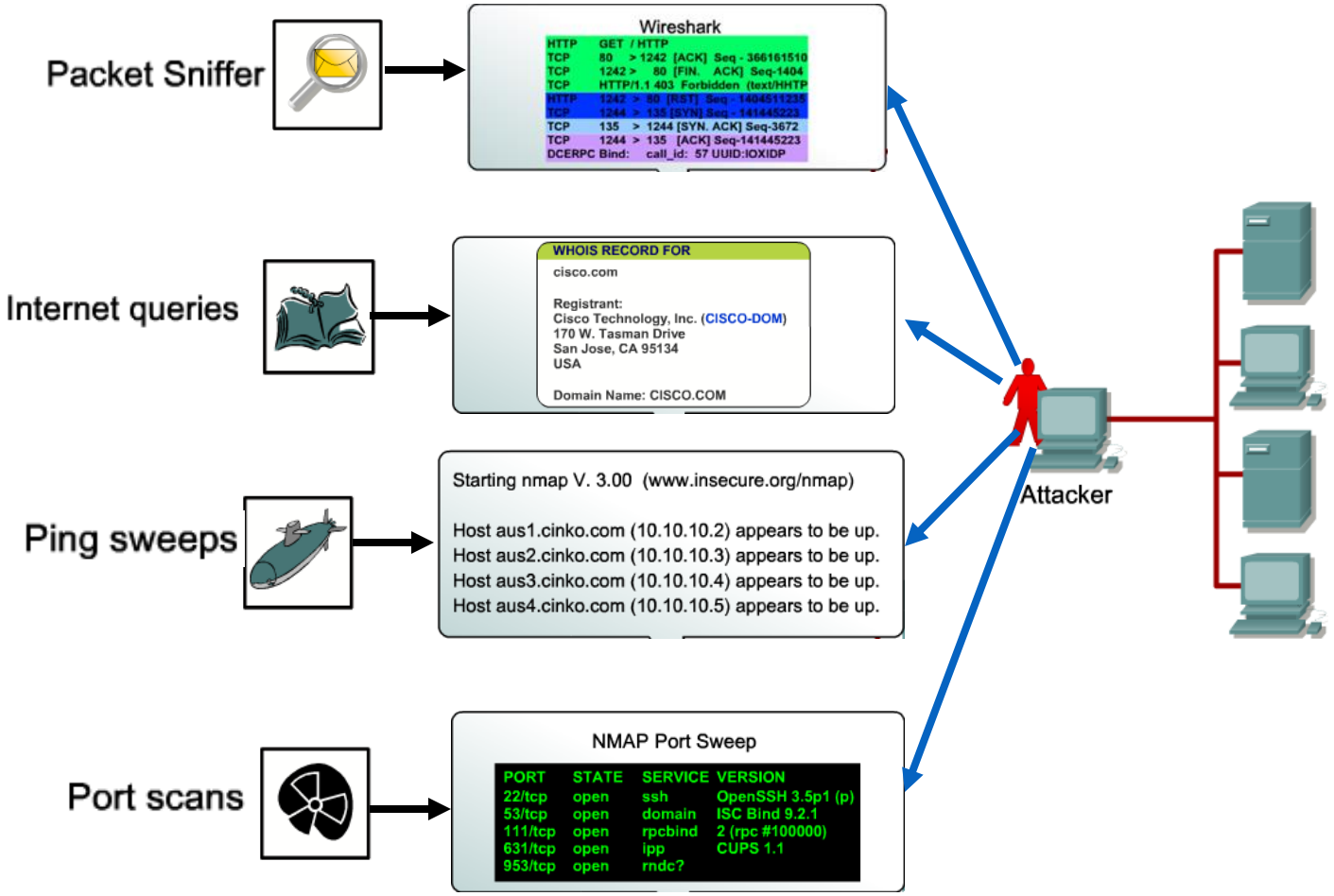
Attack Methodologies

- Reconnaissance Attacks
  - unauthorized discovery and mapping of systems, services, or vulnerabilities.
- Access Attacks
  - exploit known vulnerabilities in authentication services, FTP services, and web services.
- Denial of Service Attacks
  - send extremely large numbers of requests over a network or the Internet.

Reconnaissance Attack

- Reconnaissance attacks are the precursor to further attacks.

Reconnaissance



Access Attack

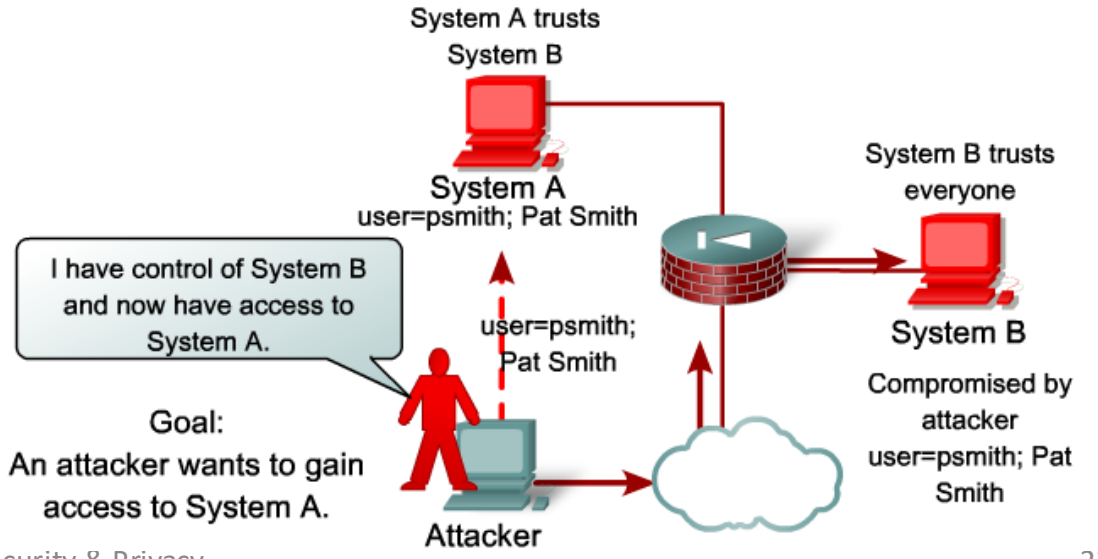
- Five types of access attacks:
  - ✓ Password attack
  - ✓ Trust exploitation
  - ✓ Port redirection
  - ✓ Man-in-the-middle attack
  - ✓ Buffer overflow

Three methods for password attacks

- ✓ Brute-force attacks
- ✓ Trojan Horse Programs
- ✓ Packet sniffers

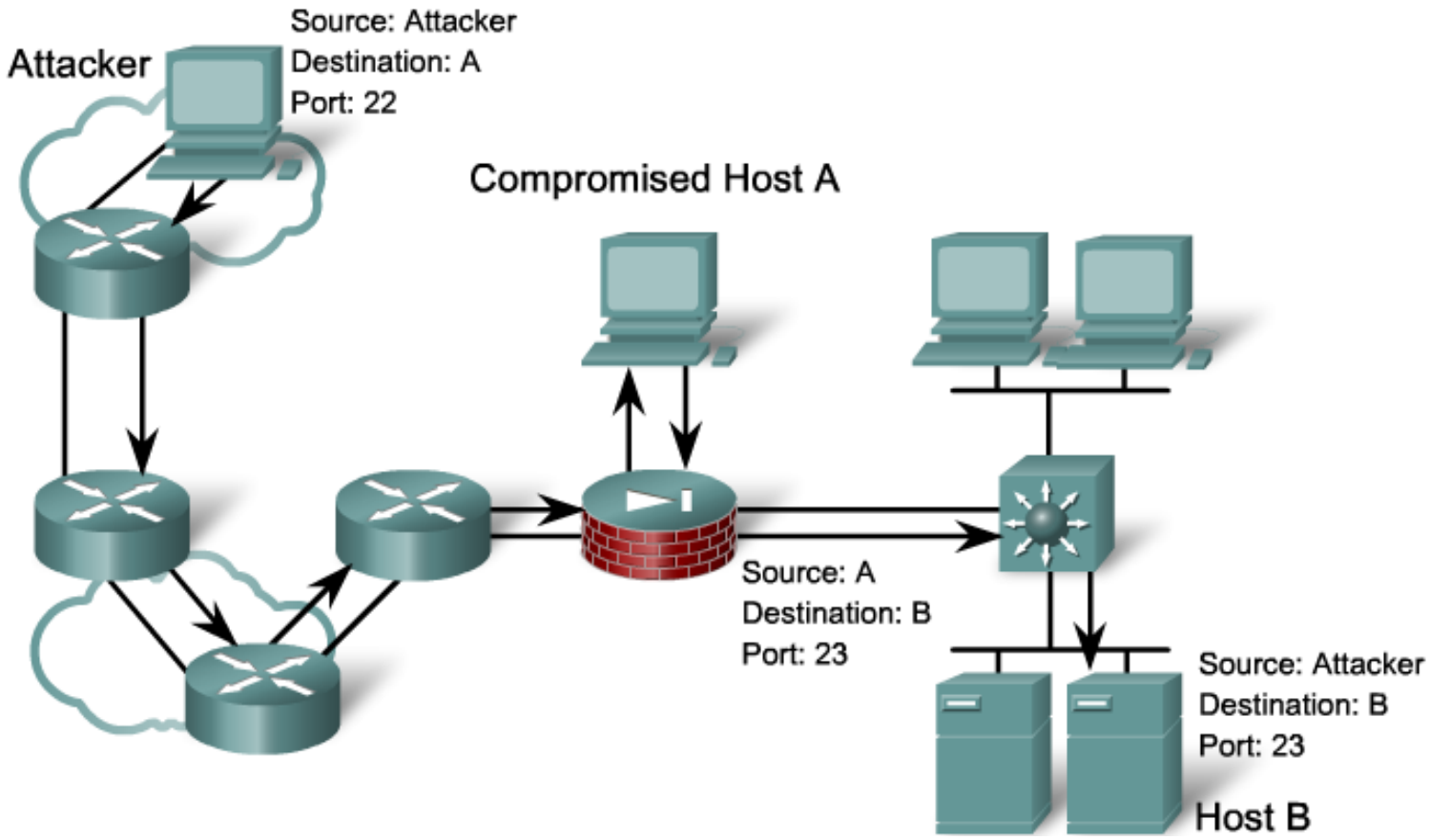
Network OS	Trust Models
Windows	Domains Active Directory (AD)
Linux and UNIX	Network File System (NFS) Network Information Service Plus (NIS+)

Trust exploitation

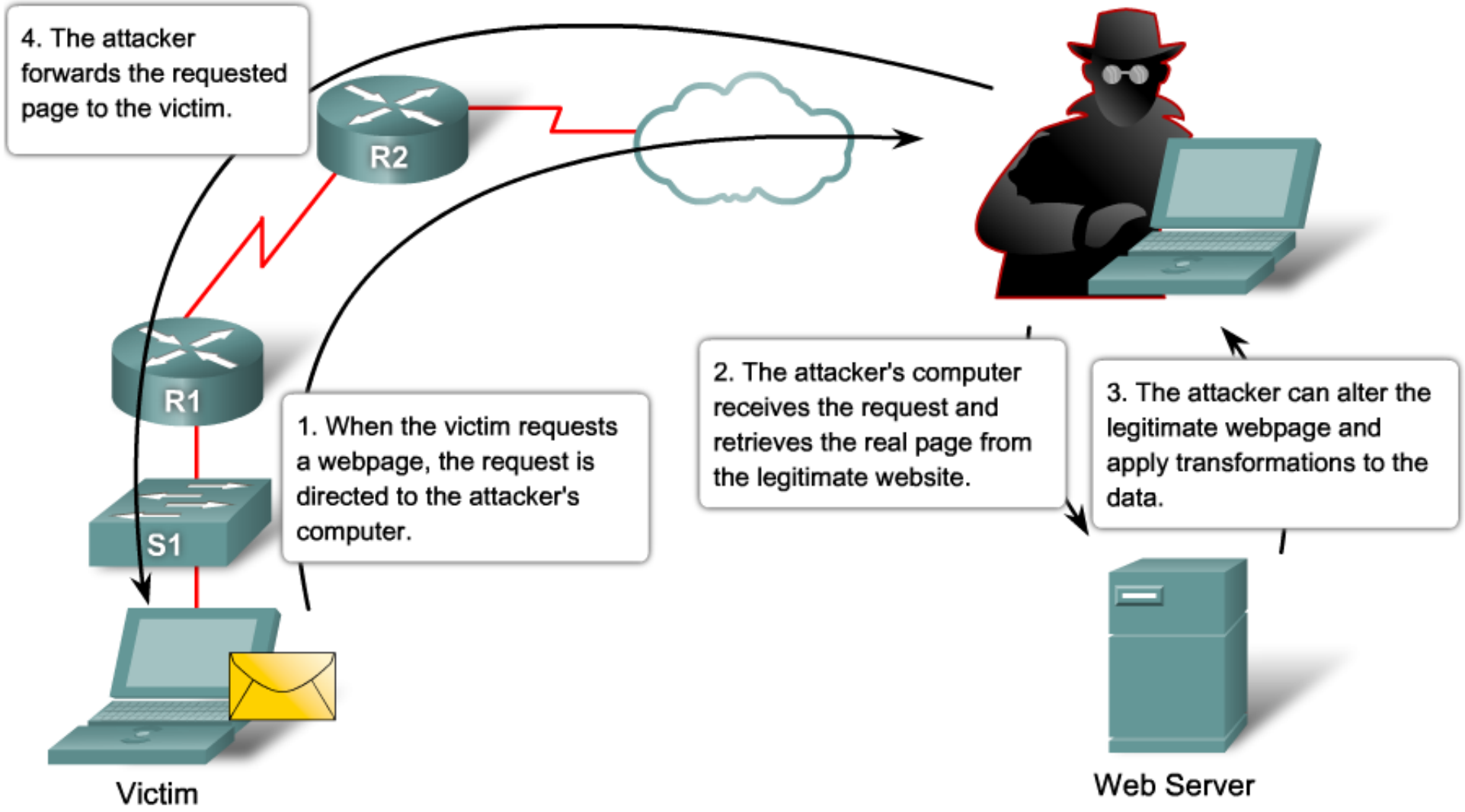


Access Attack: Port Redirection Attack

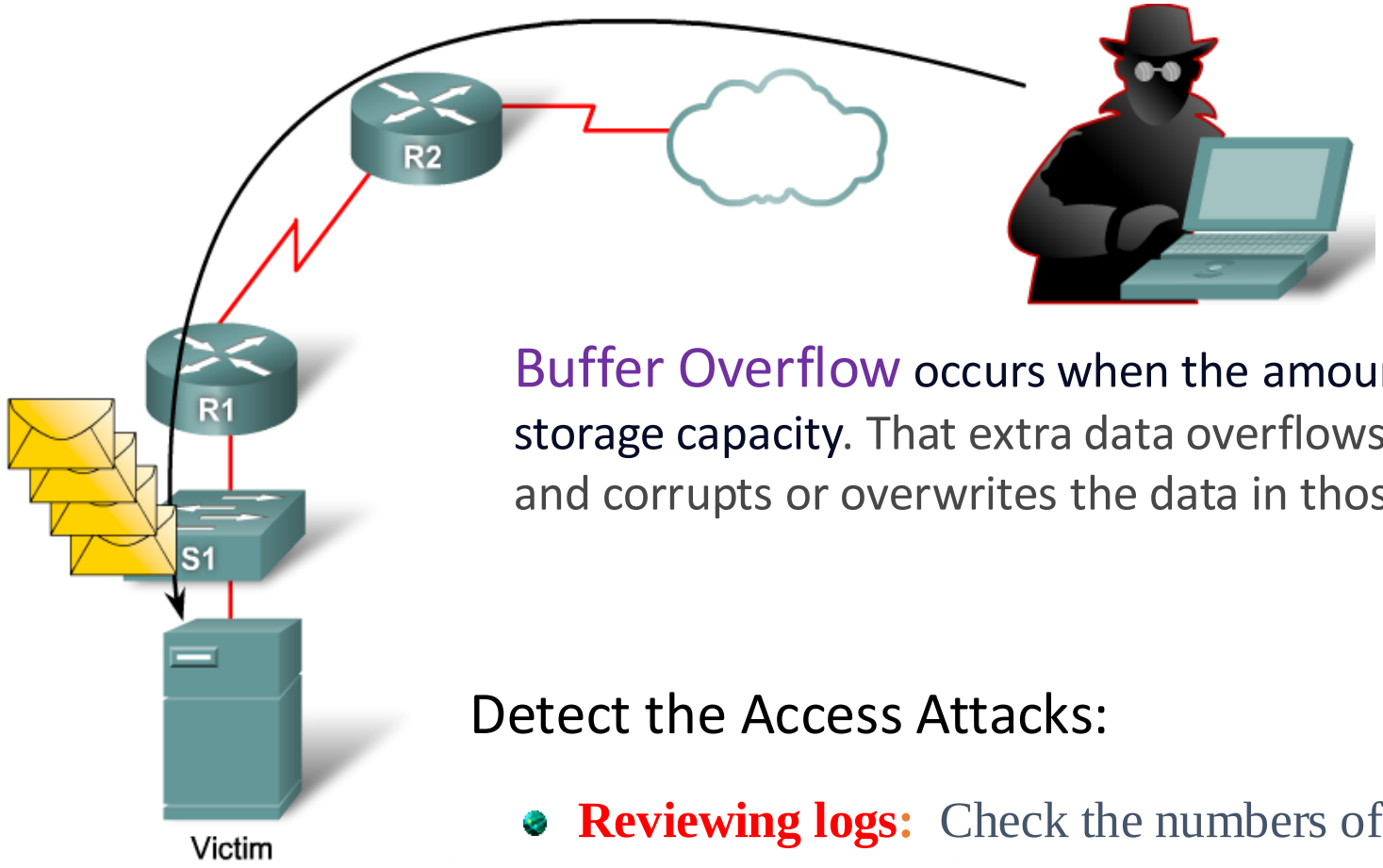
Port redirection is a type of trust-exploitation attack that uses a compromised host to pass traffic through a firewall that would otherwise be dropped. It is mitigated primarily through the use of proper trust models. Anti-virus software and host-based IDS can help detect and prevent an attacker installing port redirecting utilities on the host.



Access Attack: Man in the Middle Attack



Access Attack: Buffer Overflow Attack



Buffer Overflow occurs when the amount of data in the buffer exceeds its storage capacity. That extra data overflows into adjacent memory locations and corrupts or overwrites the data in those locations.

Detect the Access Attacks:

- **Reviewing logs:** Check the numbers of failed login attempts.
- **Bandwidth utilization:** Detect the Man-in-the-middle attacks.
- **Process loads:** Detect the buffer overflow attacks.

Access Attack: Mitigating

Strong password policy:

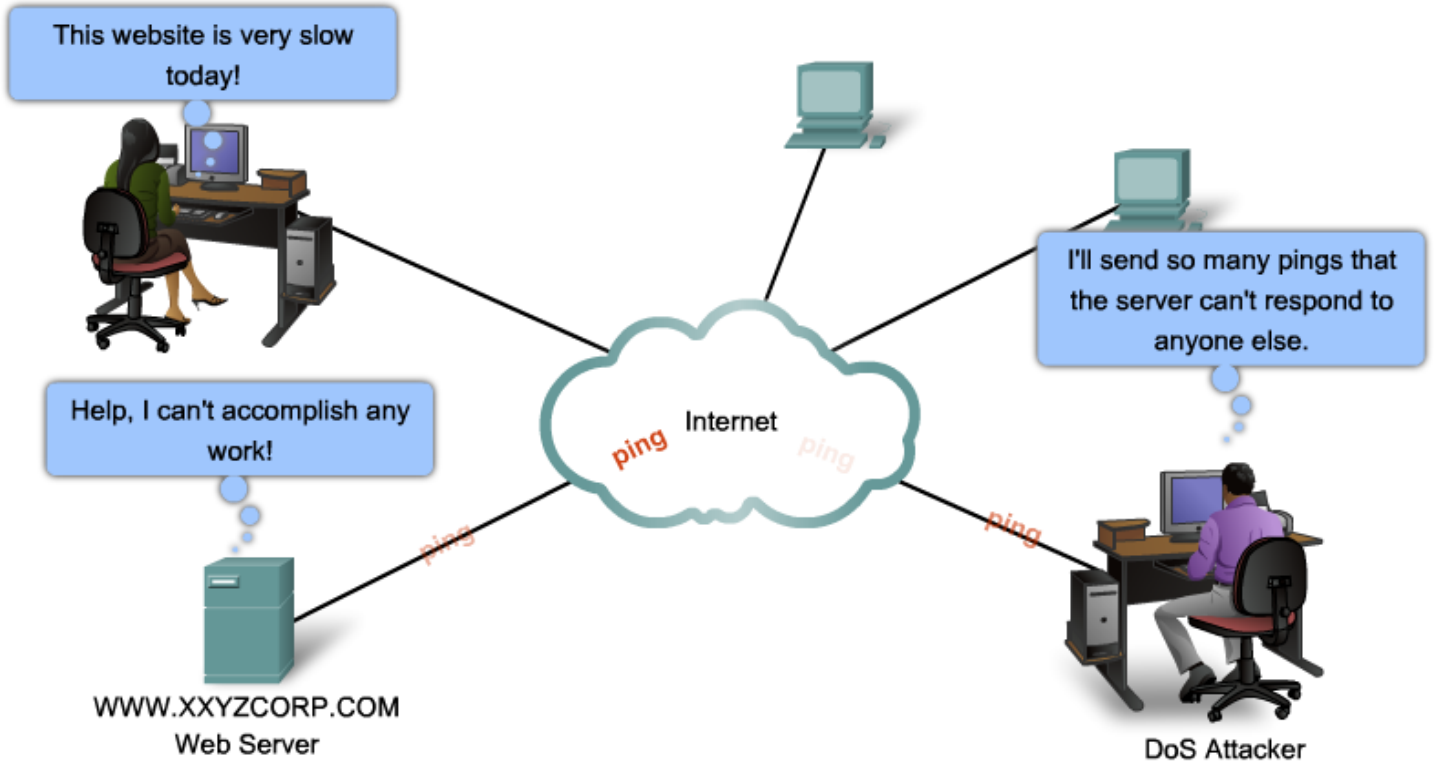
- Disabling accounts after a specific number of unsuccessful logins. This practice helps to prevent continuous password attempts.
- Not using plaintext passwords. Use either a one-time password (OTP) or encrypted password.
- Using strong passwords. Strong passwords are at least eight characters and contain uppercase letters, lowercase letters, numbers, and special characters.

Techniques Available for Access Attack Mitigation Include:

- Strong password security
- Principle of minimum trust
- Cryptography
- Applying operating system and application patches

Denial of Service (DoS) Attack

A DoS attack is a network attack



compromise the availability of a network, host, or application

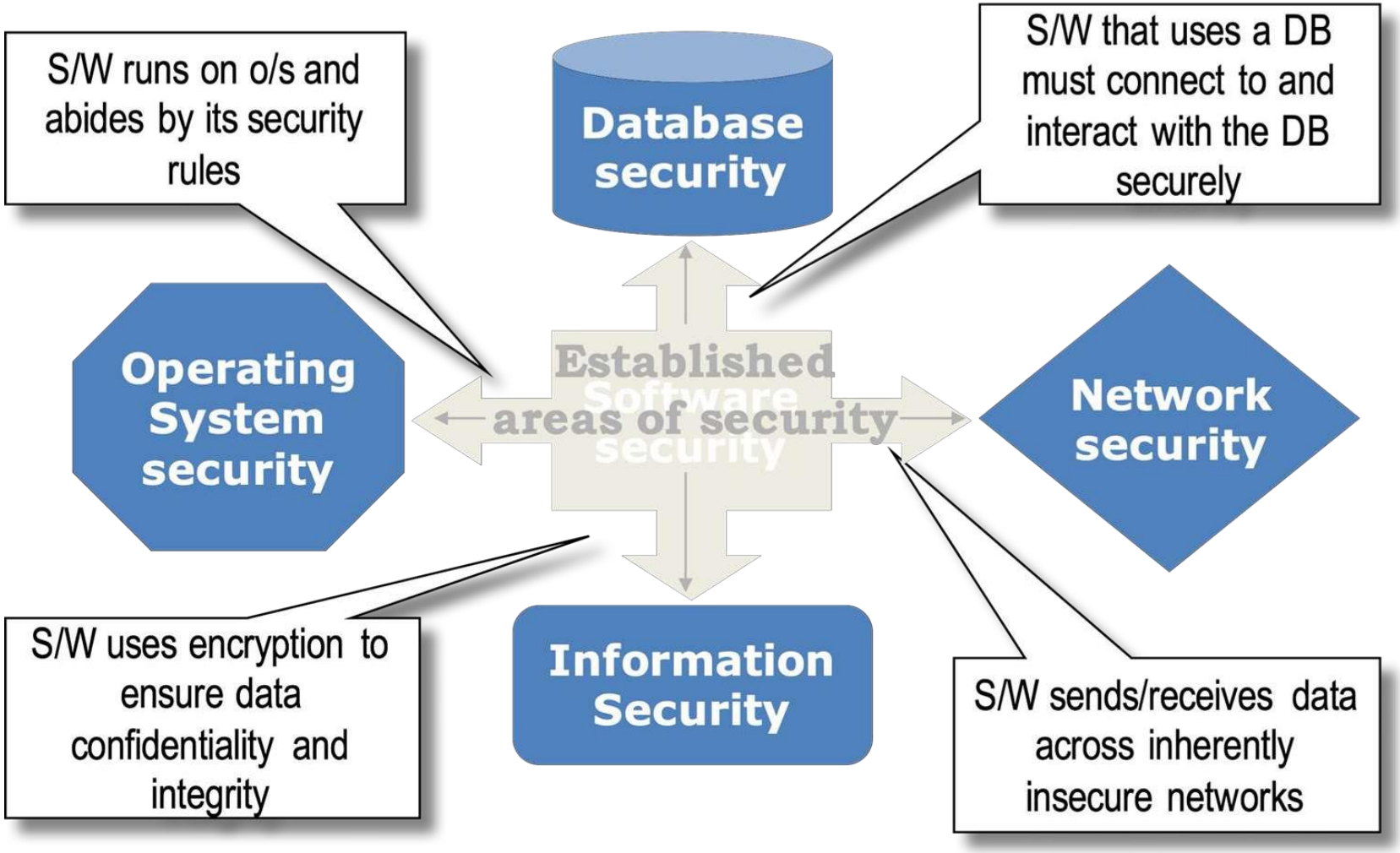
There are two major reasons a DoS attack occurs:

- A host or application fails to handle an unexpected condition.
- A network, host, or application is unable to handle an enormous quantity of data.

Social Engineering Attack

- Hacker-speak for tricking a person into revealing some confidential information
- An attack based on deceiving users or administrators at the target site
- Done to gain illicit access to systems or useful information
- The goals of social engineering are fraud, network intrusion, industrial espionage, identity theft, etc.

Where Does Software Security Fit?

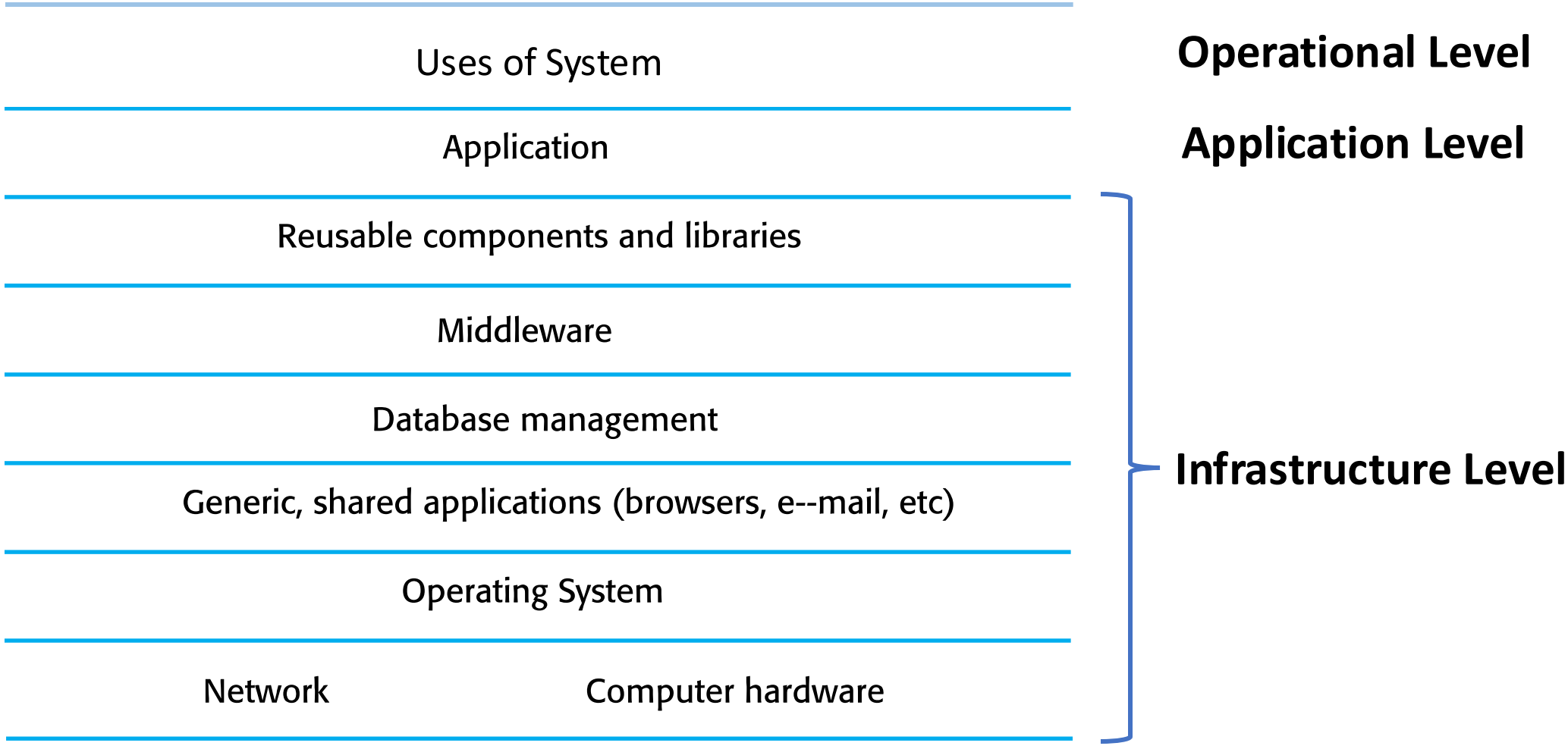


Why is Software Security a Problem?

Vulnerability: A weakness in a system is exploitable by an attacker to realize a threat, e.g., an online bank server storing user passwords in a publicly accessible server.

- **Many Vulnerabilities are being exploited**
  - ➢ Strong incentives for finding and exploiting vulnerabilities
    - ✓ Financial (black market for vulnerabilities / malware)
    - ✓ Political / Espionage (Cyber warefare / intelligence)
- **Large number of software vulnerabilities are being discovered**
  - ➢ Made worse by increasing software
    - ✓ Complexity (millions of code lines)
    - ✓ Connectivity (more potential threats, zero-day vulnerabilities)
    - ✓ Extendibility (online updates)

Software Security Levels



Software Security Levels

- **Infrastructure security**:
  - ✓ concerned with maintaining the security of all systems and networks that provide an infrastructure and a set of shared services to the organization.
  - ✓ System management problems where the infrastructure is configured to resists the attacks.
- **Application security**:
  - ✓ concerned with the security of individual application systems or related groups of systems.
  - ✓ Software Engineering problem where the system is designed to resists the attacks.
- **Operational security**:
  - ✓ concerned with the secure operation and use of the organization's systems.

System Security Management

- ❖User and permission management (Authentication & Authorization)
  - ✓ Adding and removing users from the system and setting up appropriate permissions for users
- ❖Software deployment, maintenance & Infrastructure Management
  - ✓ Installing application software and middleware and configuring these systems so that vulnerabilities are avoided.
- ❖Attack monitoring, detection and recovery
  - ✓ Monitoring the system for unauthorized access, design strategies for resisting attacks and develop backup and recovery strategies.
- ❖ Backup
  - ✓ Backup policies should be implemented to ensure that you keep undamaged copies of program and data files. These can then be restored after an attack

Security Terminologies

Term	Definition
Asset	Something of value which has to be protected. The asset may be the software system itself or data used by that system.
Attack	An exploitation of a system's vulnerability. Generally, this is from outside the system and is a deliberate attempt to cause some damage.
Control	A protective measure that reduces a system's vulnerability. Encryption is an example of a control that reduces a vulnerability of a weak access control system
Exposure	Possible loss or harm to a computing system. This can be loss or damage to data, or can be a loss of time and effort if recovery is necessary after a security breach.
Threat	Circumstances that have potential to cause loss or harm. You can think of these as a system vulnerability that is subjected to an attack.
Vulnerability	A weakness in a computer-based system that may be exploited to cause loss or harm.

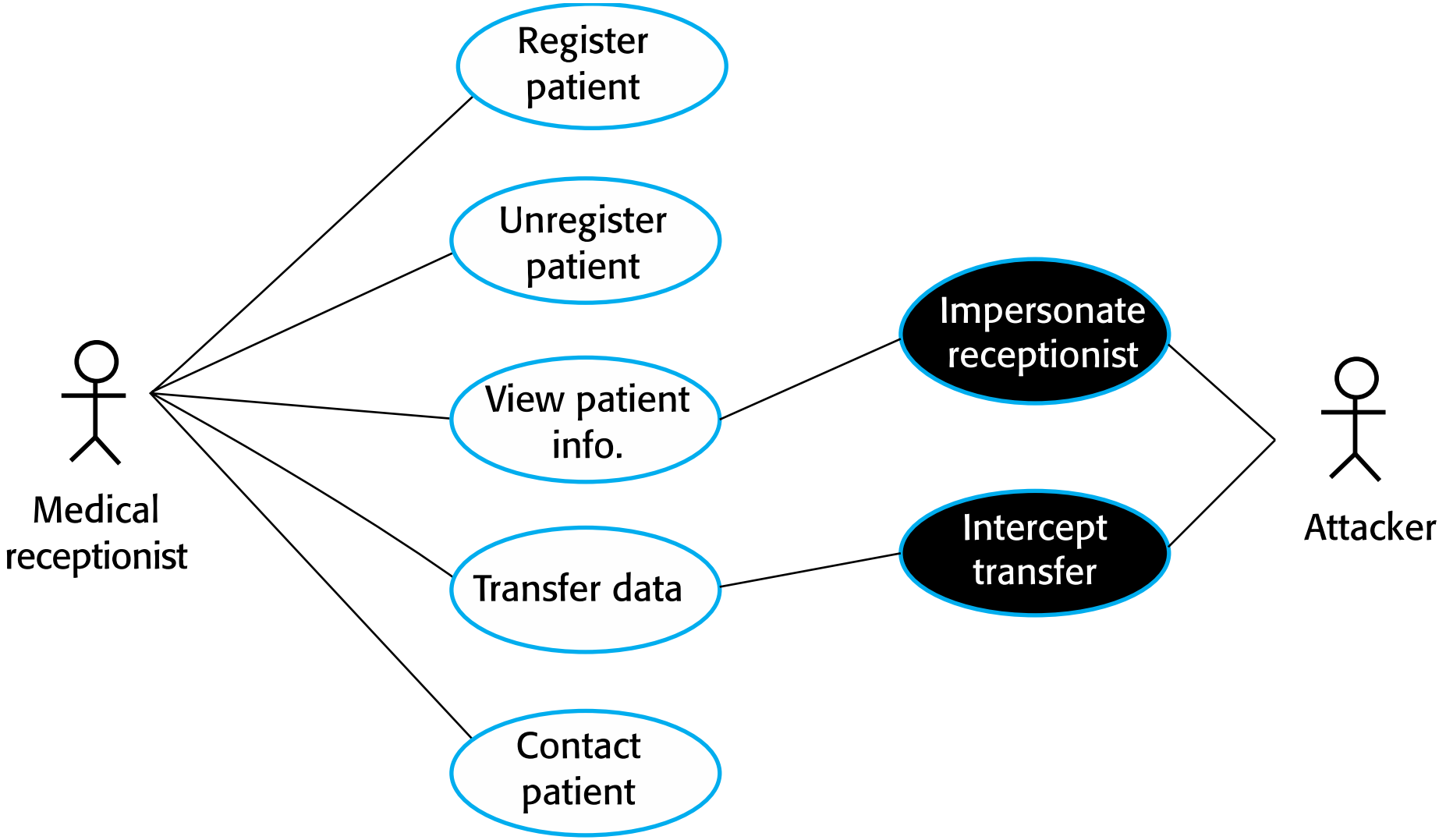
Example of Security Terminologies (MentCare)

Unauthorized access to the Mentcare system

Clinic staff log on to the Mentcare system using a username and password. The system requires passwords to be at least eight letters long but allows any password to be set without further checking. A criminal finds out that a well-paid sports star is receiving treatment for mental health problems. He would like to gain illegal access to information in this system so that he can blackmail the star.

By posing as a concerned relative and talking with the nurses in the mental health clinic, he discovers how to access the system and personal information about the nurses and their families. By checking name badges, he discovers the names of some of the people allowed access. He then attempts to log on to the system by using these names and systematically guessing possible passwords, such as the names of the nurses' children.

Example of Threats Through Misuse



Example of Security Terminologies (MentCare)

Term	Example
Asset	The records of each patient that is receiving or has received treatment.
Exposure	Potential financial loss from future patients who do not seek treatment because they do not trust the clinic to maintain their data. Financial loss from legal action by the sports star. Loss of reputation.
Vulnerability	A weak password system which makes it easy for users to set guessable passwords. User ids that are the same as names.
Attack	An impersonation of an authorized user.
Threat	An unauthorized user will gain access to the system by guessing the credentials (login name and password) of an authorized user.
Control	A password checking system that disallows user passwords that are proper names or words that are normally included in a dictionary.

MentCare Use Case – Data Transfer

Mentcare system: Transfer data	
Actors	Medical receptionist, Patient records system (PRS)
Description	A receptionist may transfer data from the Mentcare system to a general patient record database that is maintained by a health authority. The information transferred may either be updated personal information (address, phone number, etc.) or a summary of the patient's diagnosis and treatment.
Data	Patient's personal information, treatment summary.
Stimulus	User command issued by medical receptionist.
Response	Confirmation that PRS has been updated.
Comments	The receptionist must have appropriate security permissions to access the patient information and the PRS.

MentCare Misuse Case – Intercept Transfer

Mentcare system: Intercept transfer (Misuse case)	
Actors	Medical receptionist, Patient records system (PRS), Attacker
Description	A receptionist transfers data from his or her PC to the Mentcare system on the server. An attacker intercepts the data transfer and takes a copy of that data.
Data (assets)	Patient's personal information, treatment summary
Attacks	A network monitor is added to the system and packets from the receptionist to the server are intercepted. A spoof server is set up between the receptionist and the database server so that receptionist believes they are interacting with the real system.
Mitigations	All networking equipment must be maintained in a locked room. Engineers accessing the equipment must be accredited. All data transfers between the client and server must be encrypted. Certificate-based client-server communication must be used
Requirements	All communications between the client and the server must use the Secure Socket Layer (SSL). The https protocol uses certificate based authentication and encryption.

Threat Types

- **Interception threats** that allow an attacker to gain access to an asset.
  - ✓ A possible threat to the Mentcare system might be a situation where an attacker gains access to the records of an individual patient.
- **Interruption threats** that allow an attacker to make part of the system unavailable.
  - ✓ A possible threat might be a denial of service attack on a system database server so that database connections become impossible.
- **Modification threats** that allow an attacker to tamper with a system asset.
  - ✓ In the Mentcare system, a modification threat would be where an attacker alters or destroys a patient record.
- **Fabrication threats** that allow an attacker to insert false information into a system.
  - ✓ This is perhaps not a credible threat in the Mentcare system but would be a threat in a banking system, where false transactions might be added to the system that transfer money to the perpetrator's bank account.

Security Assurance

- Vulnerability avoidance
  - ✓ The system is designed so that vulnerabilities do not occur. For example, if there is no external network connection then external attack is impossible
- Attack detection and elimination
  - ✓ The system is designed so that attacks on vulnerabilities are detected and neutralised before they result in an exposure. For example, virus checkers find and remove viruses before they infect a system
- Exposure limitation and recovery
  - ✓ The system is designed so that the adverse consequences of a successful attack are minimised. For example, a backup policy allows damaged information to be restored

Security and Dependability

- ***Security and reliability***
  - ✓ If a system is attacked and the system or its data are corrupted as a consequence of that attack, then this may induce system failures that compromise the reliability of the system.
- ***Security and availability***
  - ✓ A common attack on a web-based system is a denial of service attack, where a web server is flooded with service requests from a range of different sources. The aim of this attack is to make the system unavailable.
- ***Security and safety***
  - ✓ Safety checks rely on analysing the source code of safety critical software and assume the executing code is a completely accurate translation of that source code. If this is not the case, safety-related failures may be induced and the safety case made for the software is invalid.
- ***Security and resilience***
  - ✓ Resilience is a system characteristic that reflects its ability to resist and recover from damaging events. The most probable damaging event is a cyberattack so most of the work now done in resilience is aimed at deterring, detecting and recovering from such attacks.

Security in an Organization

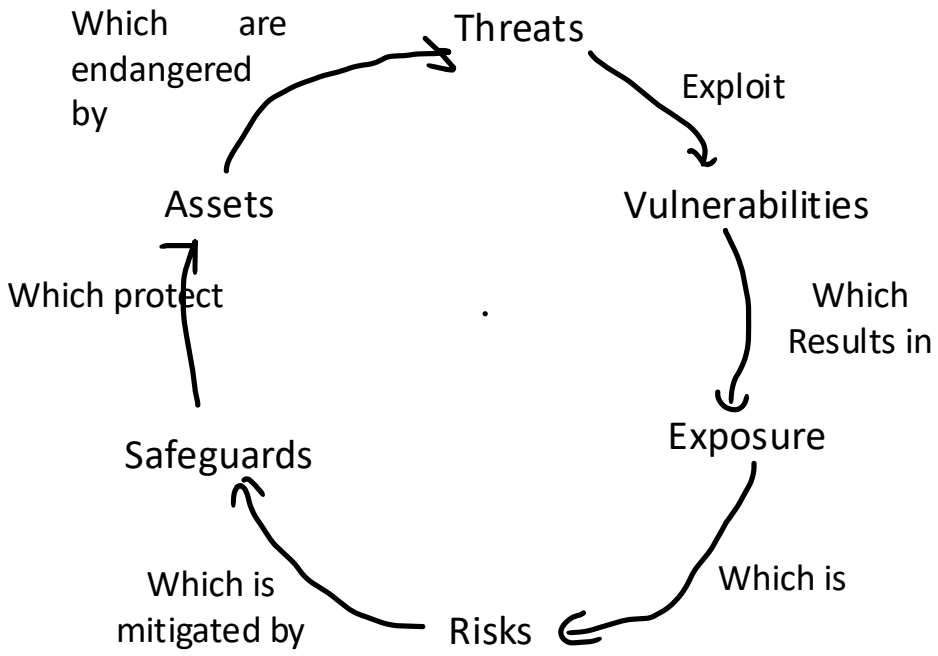
- ***Security is expensive and it is important that security decisions are made in a cost-effective way***
  - ✓ There is no point in spending more than the value of an asset to keep that asset secure.
- ***The level of protection that is required for different types of asset***
  - ✓ It is not cost-effective to apply stringent security procedures to all organizational assets. Many assets are not confidential and can be made freely available.
  - ✓ For sensitive personal information, a high level of security is required; for other information, the consequences of loss may be minor so a lower level of security is adequate.
- ***The responsibilities of individual users, managers and the organization***
  - ✓ The security policy should set out what is expected of users e.g. strong passwords, log out of computers, office security, etc.

Risk Assessment and Management

- **Threat**: Any potential occurrence that may cause an undesirable or unwanted outcome for an organization.
- **Vulnerability**: The weakness in an asset or a safeguard is a vulnerability. A vulnerability is a flaw, loophole, oversight, error, limitation, frailty, or susceptibility that enables a threat to cause harm.
- **Exposure**: Exposure is being susceptible to asset loss because of a threat; there is the possibility that a vulnerability can or will be exploited by a threat agent or event.
- **Risk**: Risk is the possibility or likelihood that a threat will exploit a vulnerability to cause harm to an asset and the severity of damage that could result.

Risk = Threat * vulnerability or

Risk = probability of harm * severity of harm



The cyclical Relationship of risk elements

Risk Assessment and Management

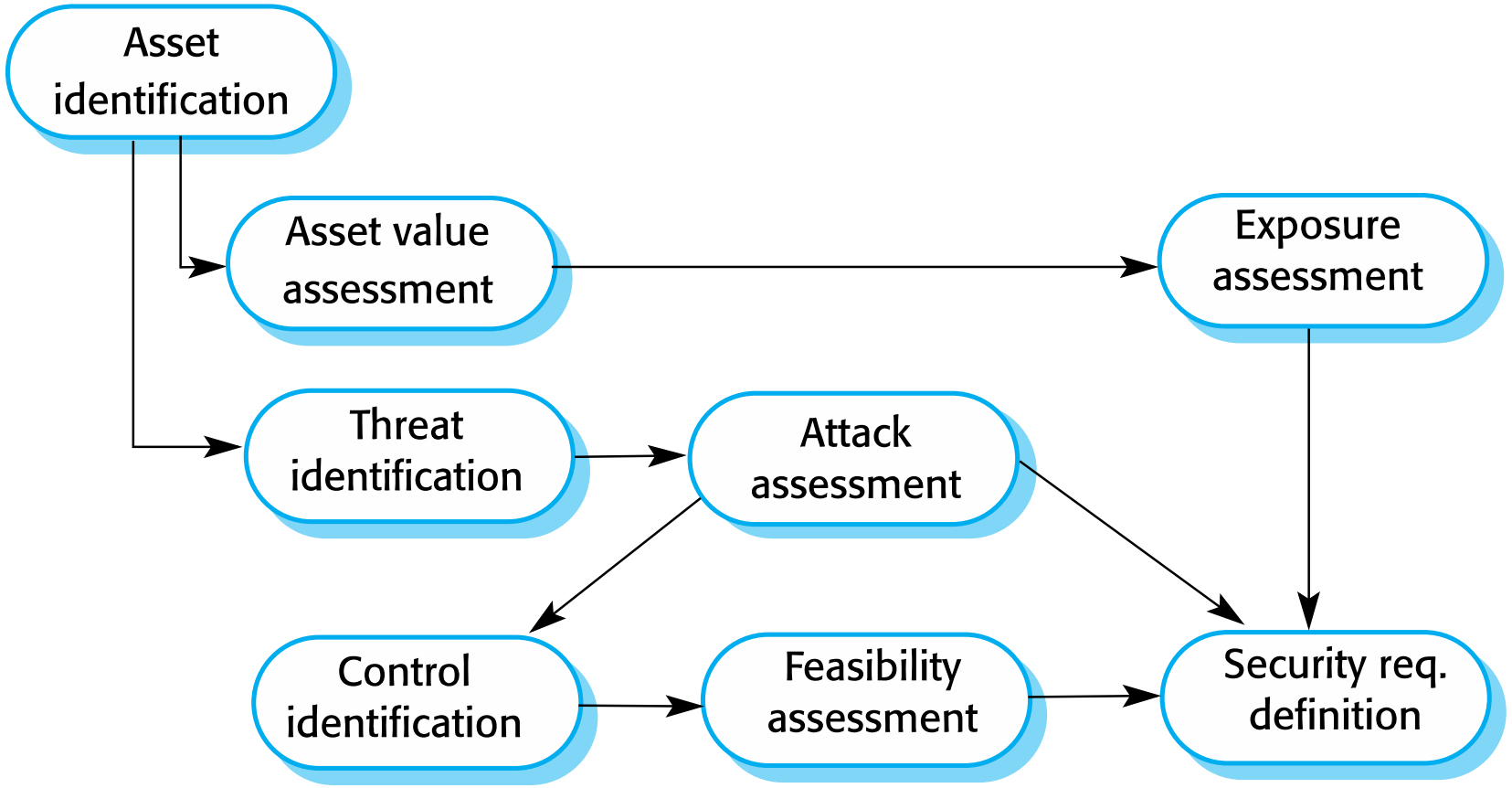
- **Safeguards**: A safeguard, security control, protection mechanism, or countermeasure is anything that removes or reduces a vulnerability or protects against one or more specific threats. This concept is also known as a risk response.
- **Asset**: An asset is a person, place or anything, whether tangible or intangible.

Asset Valuation: Asset valuation is value assigned to an asset based on several factors, including importance to the organization, use in critical process, actual cost, and nonmonetary expenses/costs (such as time, attention, productivity, and research and development).

The cyclical Relationship of risk elements

- If an asset has no value, there is no need to provide protection for it. It makes no sense to spend $100,000 protecting an asset that is worth only $1,000. As a rule, the annual costs of safeguards should not exceed the potential annual cost of asset value loss.

Risk Assessment



Risk Assessment

There are two primary risk assessment methodologies: qualitative and quantitative.

- ✓ Quantitative risk analysis assigns real dollar figures to the loss of an asset and is based on mathematical calculations.
- ✓ Qualitative risk analysis assigns subjective and intangible values to the loss of an asset and takes into account perspectives, feelings, intuition, preferences, ideas, and gut reactions.

The method of combining quantitative and qualitative analysis into a final assessment of organizational risk is known as hybrid assessment or hybrid analysis.

Qualitative Risk Assessment

Scenarios Method

- A scenario is a written description of a single major threat. The scenarios are limited to one page of text to keep them manageable.
- The analysis participants then assign to the scenario a threat level, a loss potential, and the advantages of each safeguard. These assignments can be simple—such as High, Medium, and Low, or a basic number scale of 1 to 10—or they can be detailed essay responses.
- The responses from all participants are then compiled into a single report that is presented to upper management.
- The usefulness and validity of a qualitative risk analysis improves as the number and diversity of the participants in the evaluation increases.

Delphi Technique

- The Delphi technique is simply an anonymous feedback-and-response process used to enable a group to reach an anonymous consensus.
- The participants are usually gathered into a single meeting room. To each request for feedback, each participant writes down their response on paper or through digital messaging services anonymously. The results are compiled and presented to the group for evaluation. The process is repeated until a consensus is reached.

Quantitative Risk Assessment

Major steps in Quantitative Risk Analysis:

1. Inventory assets and assign asset value (AV)
2. For each asset-threat pairing, calculate the exposure factor (EF).
3. For each asset-threat pairing, calculate the single loss expectancy (SLE)
4. Assess the annualized rate of occurrence (ARO)
5. Derive the annualized loss expectancy (ALE)
6. Perform cost/benefit analysis of countermeasures

Quantitative Risk Assessment

Exposure Factor (EF)

- The EF indicates the expected overall asset value loss because of a single realized risk. The EF is expressed as a percentage.
- The EF is usually small for assets that are easily replaceable, such as hardware and can be very large for assets that are irreplaceable or proprietary, such as product designs or a database of customers.
- The EF is determined by using historical internal data, performing statistical analysis, consulting public or subscription risk ledgers/registers, working with consultants, or using a risk management software solution.

Single Loss Expectancy (SLE)

- The single-loss expectancy (SLE) is the potential loss associated with a single realized threat against a specific asset.
- SLE = asset value (AV) * exposure factor (EF)

Quantitative Risk Assessment

Annualized Rate of Occurrence (ARO)

- The annualized rate of occurrence (ARO) is the expected frequency with which a specific threat or risk will occur within a single year.
- The ARO can range from a value of 0.0 (zero), indicating that the threat or risk will never be realized, to a very large number, indicating that the threat or risk occurs often.
- The ARO can be derived by reviewing historical internal data, performing statistical analysis, consulting public or subscription risk ledgers/registers, working with consultants, or using a risk management software solution.

Annualized Loss Expectancy (ALE)

- The annualized loss expectancy (ALE) is the possible yearly loss of all instances of a specific realized threat against a specific asset.

ALE = single loss expectancy (SLE) * Annualized rate of occurrence (ARO)

= Asset value (AV) * Exposure factor (EF) *
Annualized rate of occurrence (ARO)

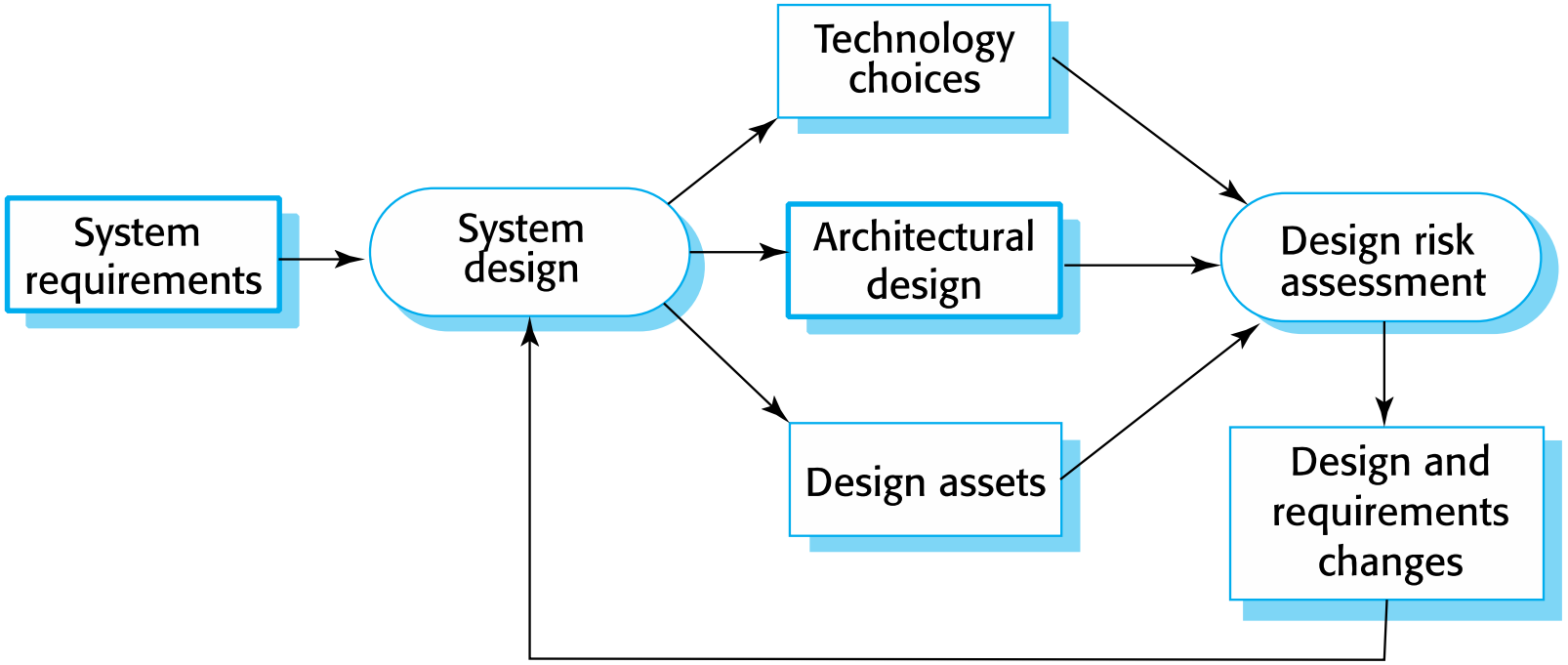
Quantitative Risk Assessment

Concept	Formula or meaning
Asset Value (AV)	$
Exposure factor or severity of damage (EF)	%
Single loss expectancy (SLE)	SLE = AV * EF
Annualized rate of occurrence (ARO)	# / year
Annualized loss expectancy (ALE)	ALE = SLE * ARO or ALE = AV * EF * ARO
Annual cost of the safeguard (ACS)	$ / year
Value or benefit of a safeguard (i.e., cost/benefit equation)	(ALE1 – ALE2) – ACS

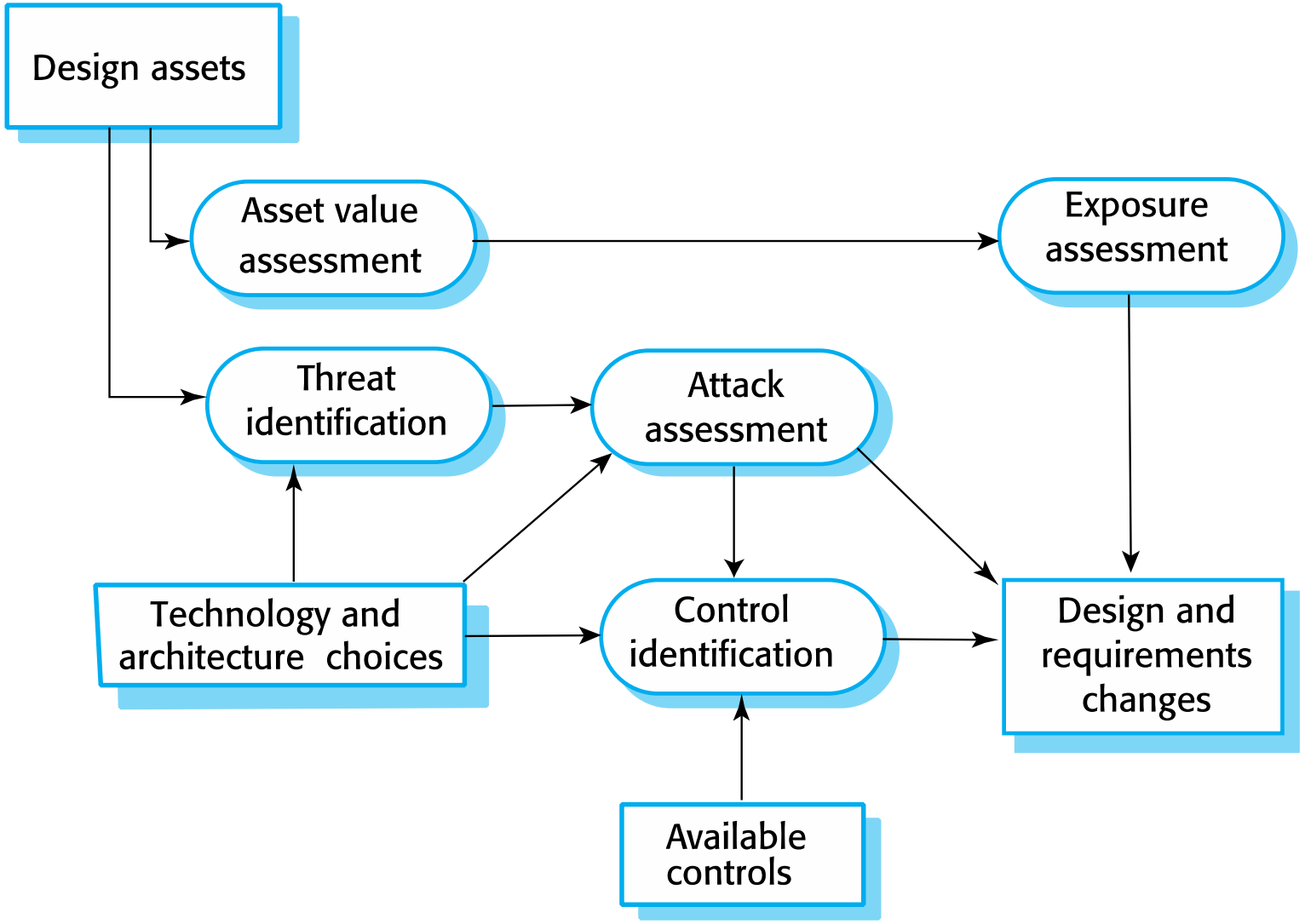
Mathematical Problems:

1. If an asset is valued at $200,000 and it has an EF of 45 percent for a specific threat, then what is the SLE of the threat for that asset?
2. If the SLE of an asset is $90,000 and the ARO for a specific threat (such as total power loss) is 0.5, then what is the ALE? If the ARO for a specific threat (such as compromised user account) is 15 for the same asset, then what is the ALE?

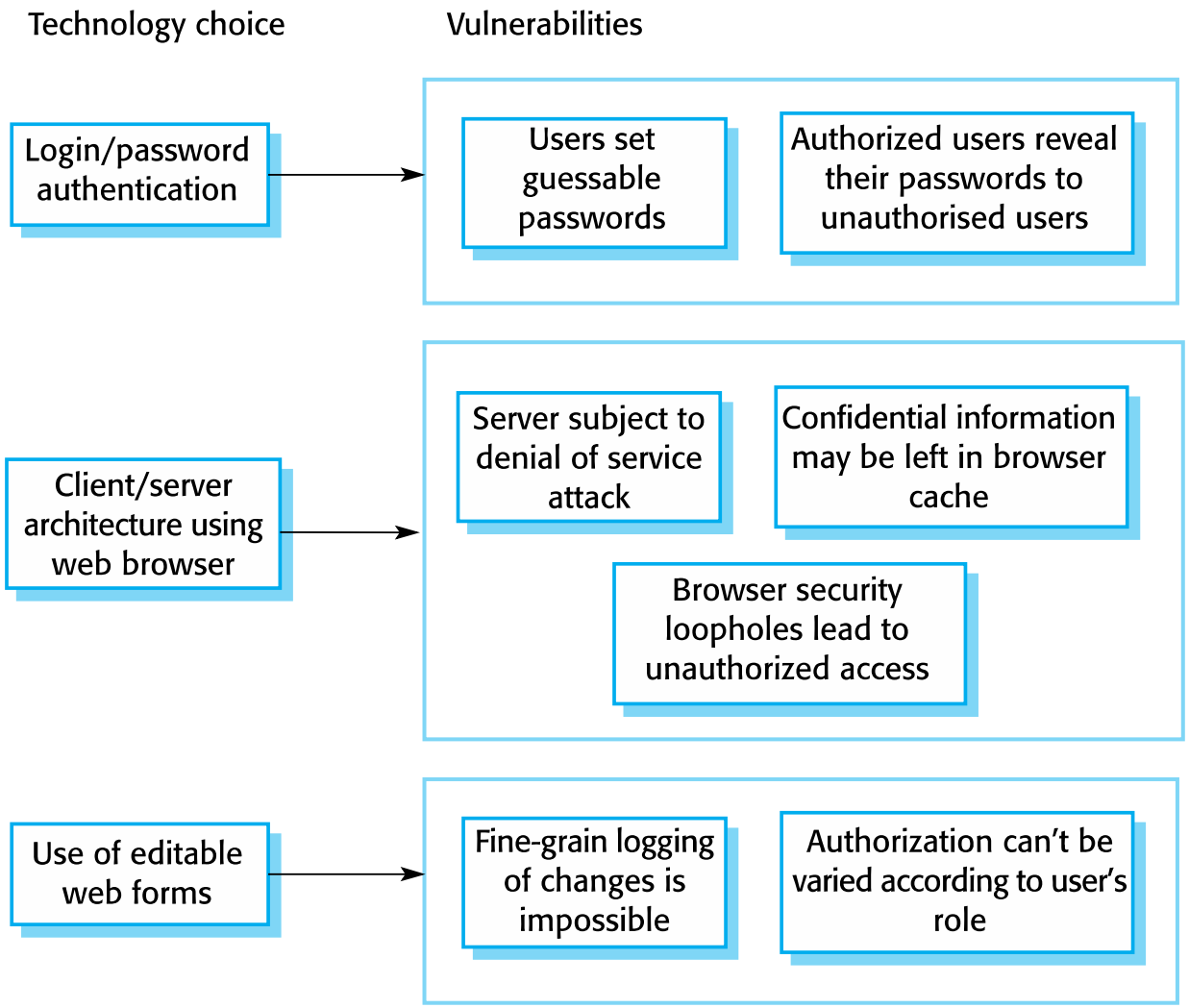
Design and Risk Assessment



Design and Risk Assessment



Vulnerabilities Associated with Technology Choices



System Requirements:

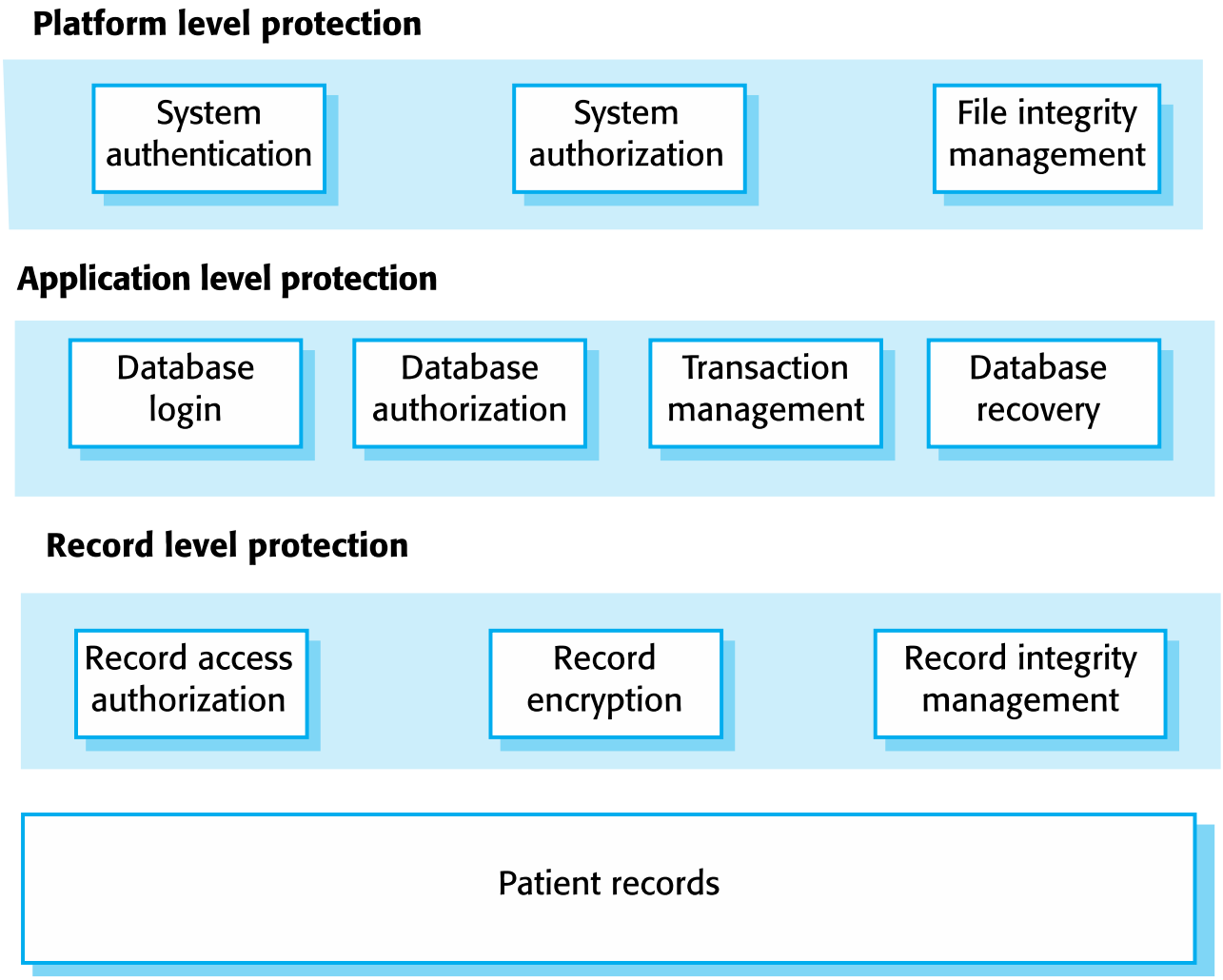
- ✓ A password checker shall be made available and shall be run daily. Weak passwords shall be reported to system administrators.
- ✓ Access to the system shall only be allowed by approved client computers.
- ✓ All client computers shall have a single, approved web browser installed by system administrators.

Architectural Design

- Two fundamental issues have to be considered when designing an architecture for security.
  - ✓Protection
    - How should the system be organized so that critical assets can be protected against external attack?
  - ✓Distribution
    - How should system assets be distributed so that the effects of a successful attack are minimized?
- These are potentially conflicting
  - ✓If assets are distributed, then they are more expensive to protect. If assets are protected, then usability and performance requirements may be compromised.

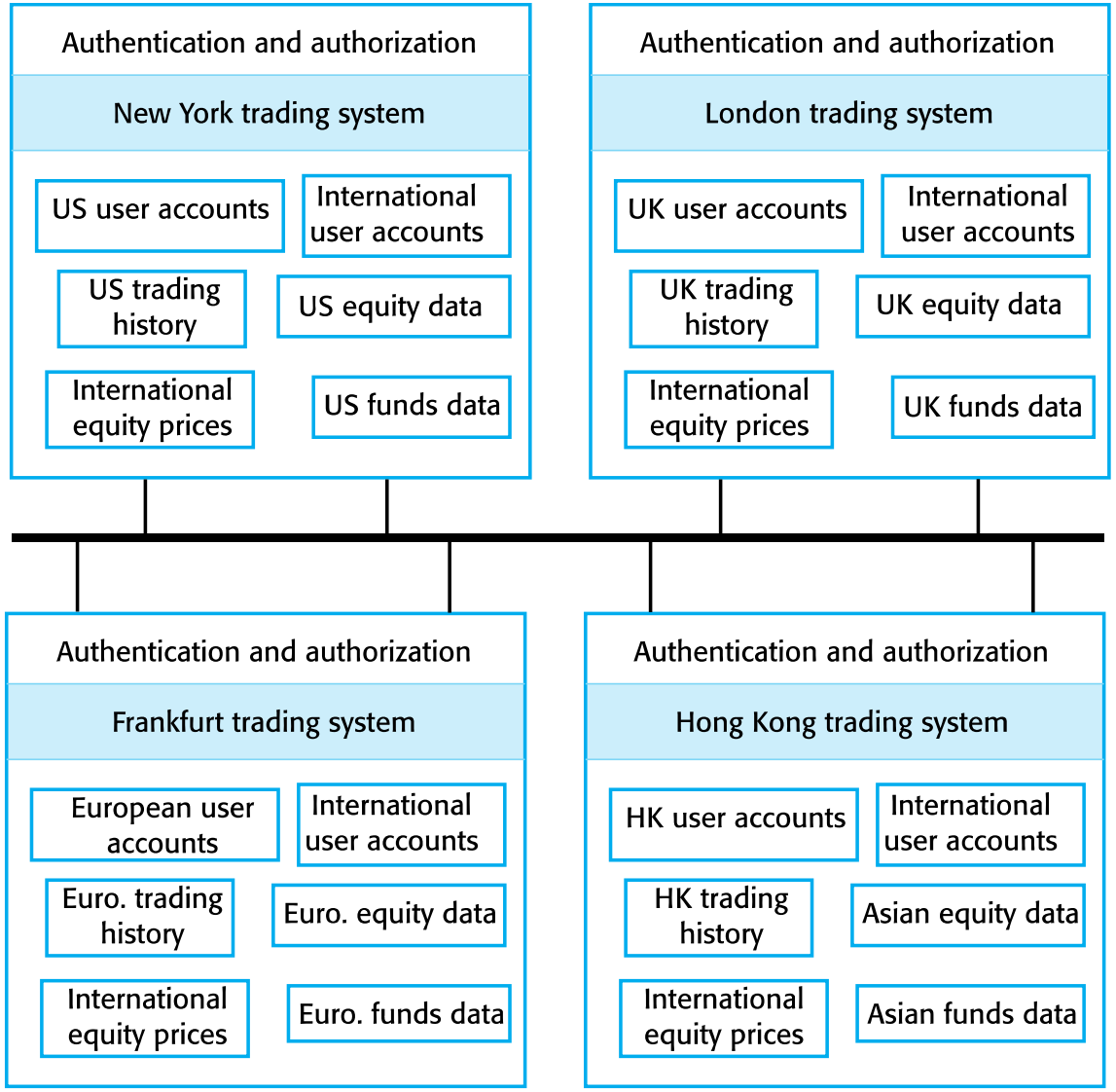
Architectural Design: Layered Protection

- Platform-level protection
  - ✓ Top-level controls on the platform on which a system runs.
- Application-level protection
  - ✓ Specific protection mechanisms built into the application itself e.g. additional password protection.
- Record-level protection
  - ✓ Protection that is invoked when access to specific information is requested



Architectural Design: Distribution

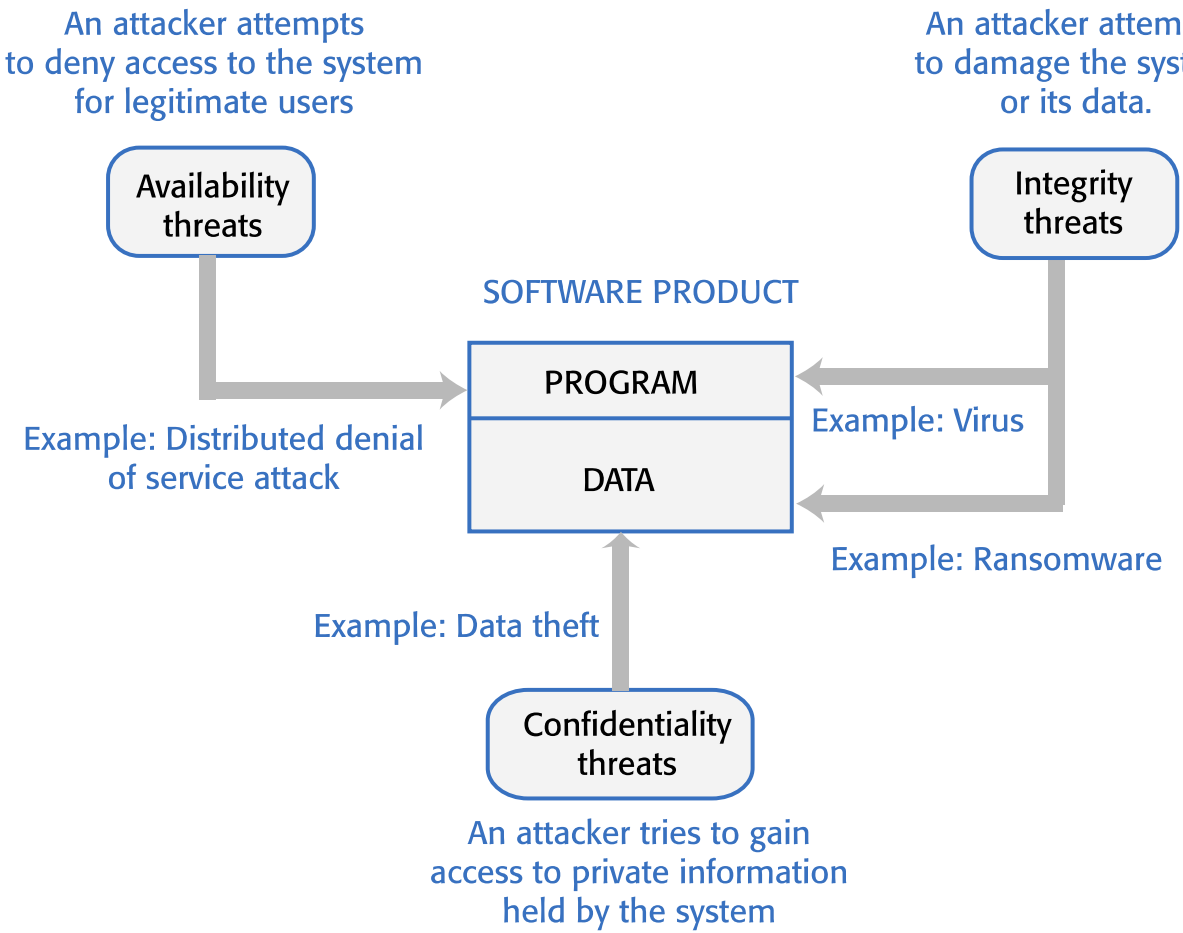
- Distributing assets means that attacks on one system do not necessarily lead to complete loss of system service
- Each platform has separate protection features and may be different from other platforms so that they do not share a common vulnerability
- Distribution is particularly important if the risk of denial of service attacks is high



Operational Security

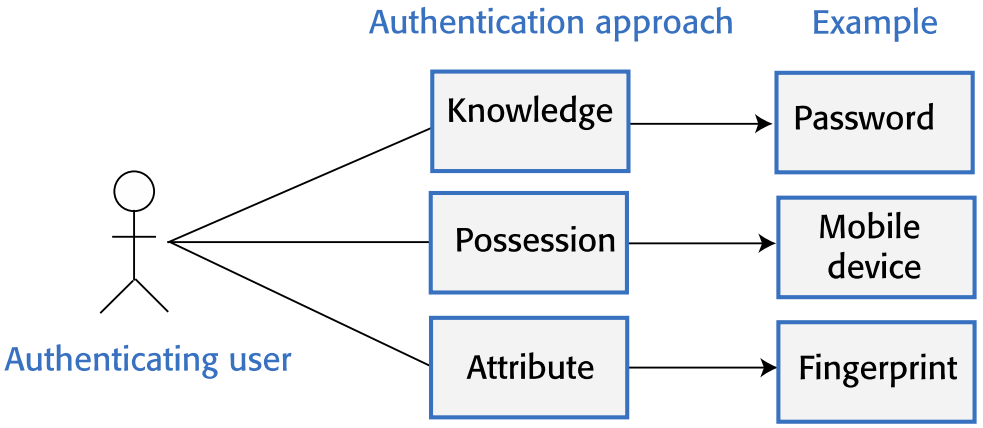
- It focuses on helping users to maintain security
- User attacks try to trick users into disclosing their credentials or accessing a website that includes malware, such as a key-logging system
- Operational security procedures and practices
  - **Auto-logout,** which addresses the common problem of users forgetting to logout from a computer used in a shared space
  - **User command logging**, which makes it possible to discover actions taken by users that have deliberately or accidentally damaged some system resources
  - **Multi-factor authentication**, which reduces the chances of an intruder gaining access to the system using stolen credentials

Security Threats: Based on CIA Triad



Authentication

- Authentication is the process of ensuring that a user of your system is who they claim to be
- You need authentication in all software products that maintain user information so that only the providers of that information can access and change it
- You also use authentication to learn about your users so that you can personalize their experience of using your product



Authentication Methods

- Knowledge-based authentication
  - ✓ The user provides secret, personal information when they register with the system. Each time they log on, the system asks them for this information
- Possession-based authentication
  - ✓ This relies on the user having a physical device (such as a mobile phone) that can generate or display information that is known to the authenticating system. The user inputs this information to confirm possession of the device
- Attribute-based authentication
  - ✓ This is based on a unique biometric attribute of the user, such as a fingerprint, which is registered with the system
- Multi-factor authentication
  - ✓ combines these approaches and requires users to use more than one authentication method

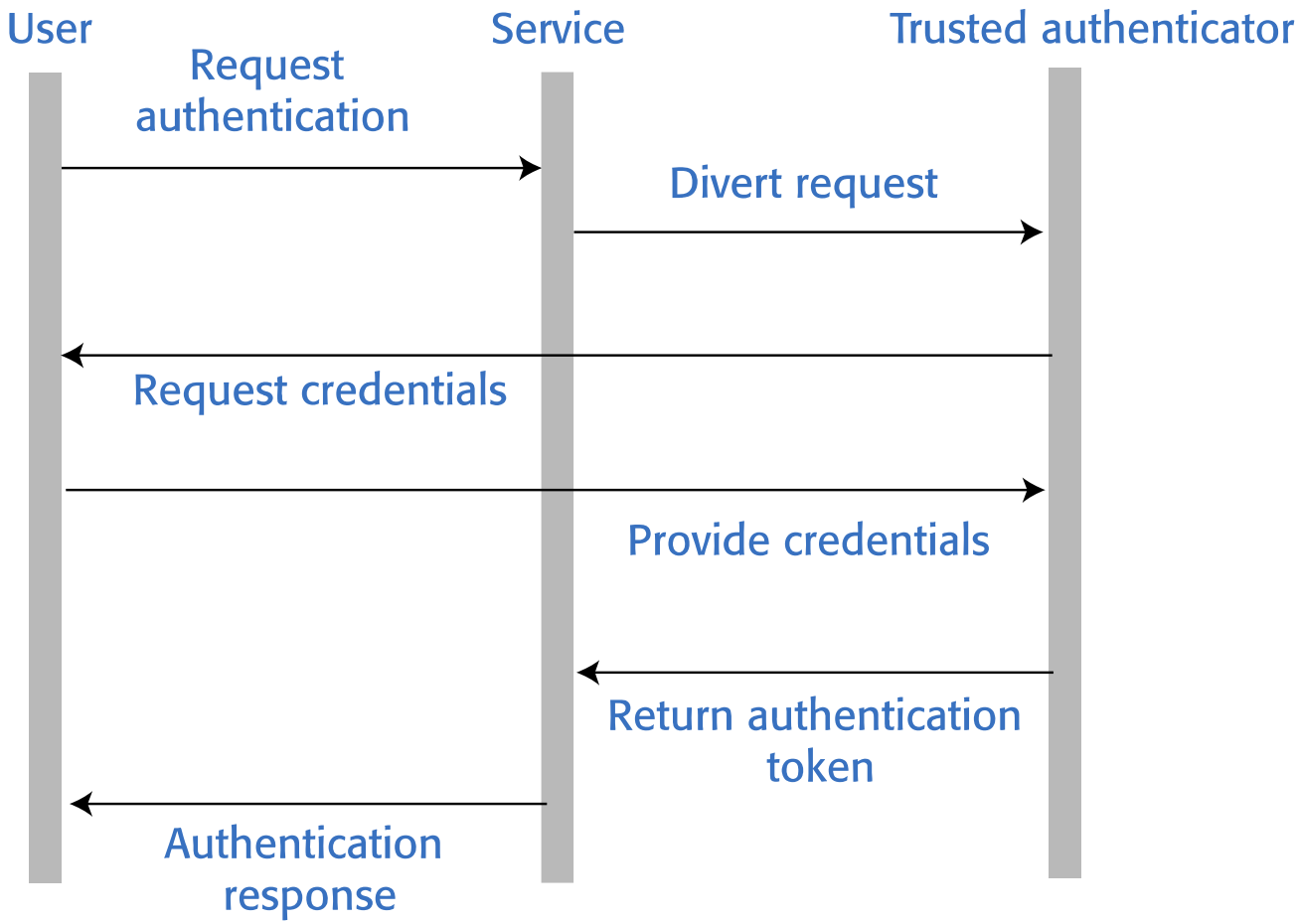
Weakness of Password Based Authentication

- Insecure passwords
  - ✓ Users choose passwords that are easy to remember. It is easy for attackers to guess/generate these passwords, either dictionary or brute force attack
- Phishing attacks
  - ✓ Users click on an email link that points to a fake site that tries to collect their login and password details
- Password reuse
  - ✓ Users use the same password for several sites. If there is a security breach at one of these sites, attackers then have passwords they can try on other sites
- Forgotten passwords
  - ✓ Users regularly forget their passwords, so you need to set up a password recovery mechanism to allow these to be reset. This can be a vulnerability if users' credentials have been stolen and attackers use them to reset their passwords.

Federated Identity Based Authentication

- Federated identity is an approach to authentication where you use an external authentication service
- ‘Login with Google’ and ‘Login with Facebook’ are widely used examples of authentication using federated identity
- The advantage of federated identity for a user is that they have a single set of credentials that are stored by a trusted identity service
- Instead of logging into a service directly, a user provides their credentials to a known service that confirms their identity to the authenticating service
- They don’t have to keep track of different user IDs and passwords. Because their credentials are stored in fewer places, the chances of a security breach where these are revealed are reduced.

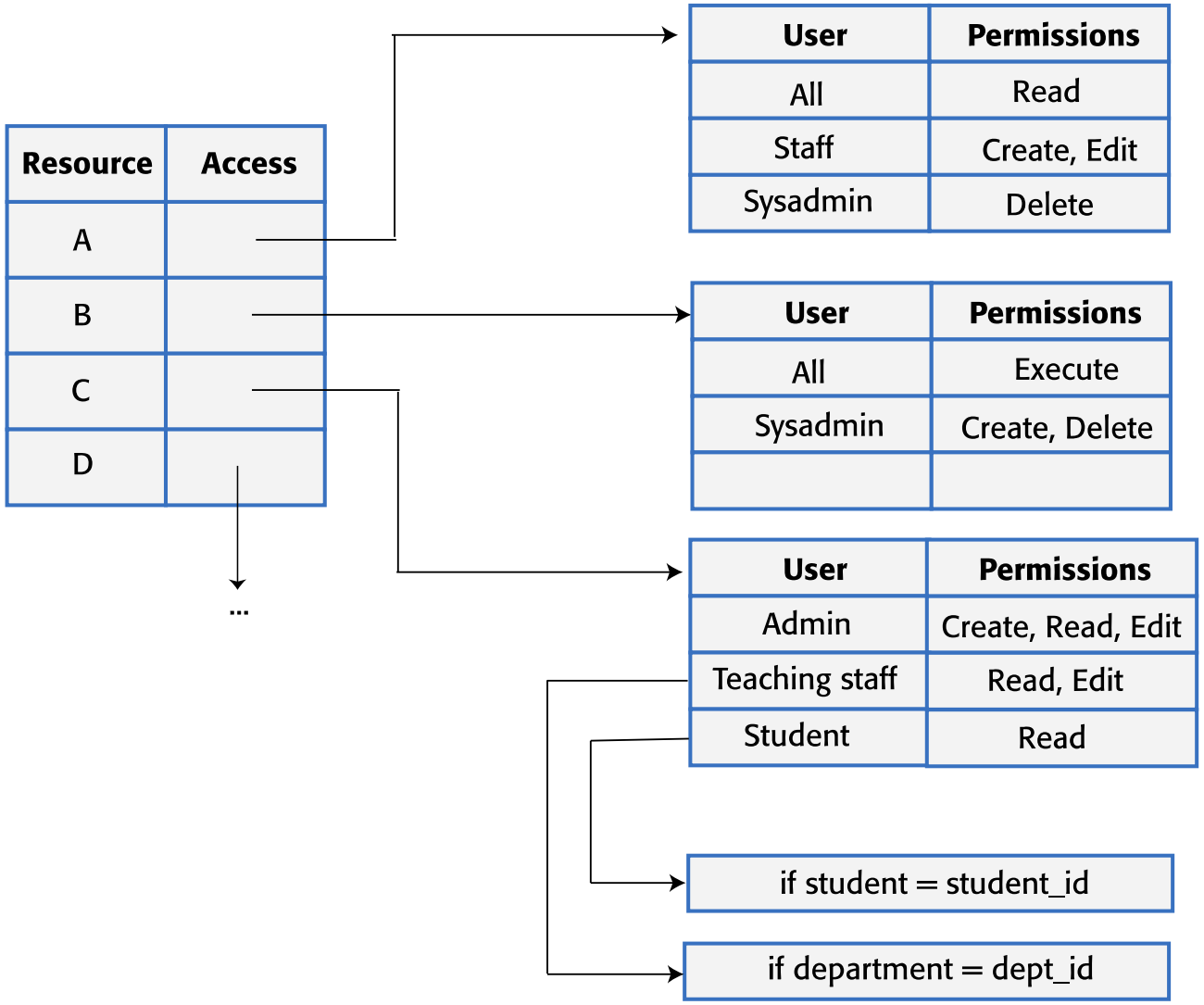
Federated Identity Based Authentication



Authorization

- Authentication involves a user proving their identity to a software system
- Authorization is a complementary process in which that identity is used to control access to software system resources
- For example, if you use a shared folder on Dropbox, the folder's owner may authorize you to read the contents of that folder, but not to add new files or overwrite files in the folder
- When a business wants to define the type of access that users get to resources, this is based on an access control policy
- This policy is a set of rules that define what information (data and programs) is controlled, who has access to that information, and the type of access that is allowed

Access Control Lists



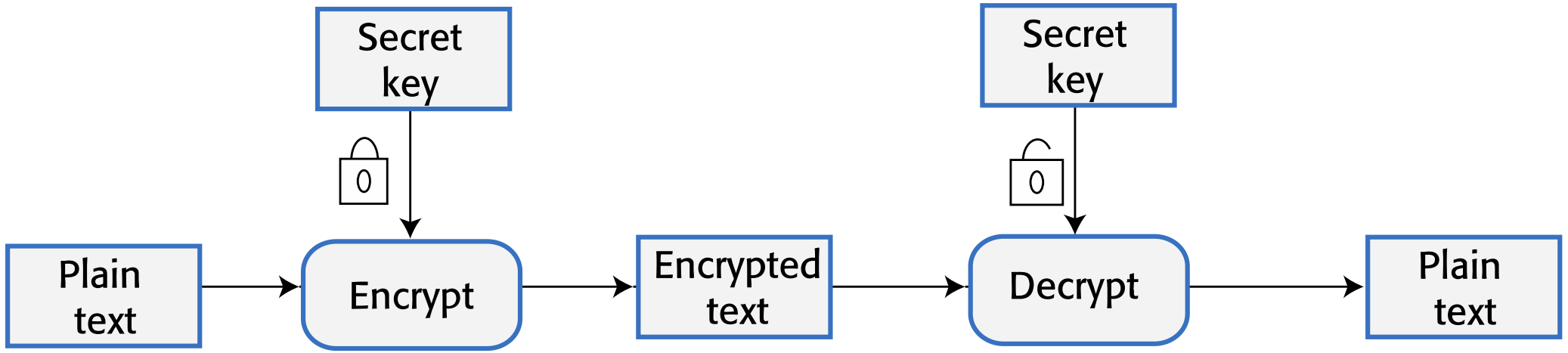
Access Control Lists

- Access control lists (ACLs) are used in most file and database systems to implement access control policies
- Access control lists are tables that link users with resources and specify what those users are permitted to do.
  - For example, for a book, I would like to be able to set up an access control list to a book file that allows reviewers to read that file and annotate it with comments. However, they are not allowed to edit the text or delete the file
- If access control lists are based on individual permissions, then these can become very large
- However, you can dramatically cut their size by allocating users to groups and then assigning permissions to the group

Encryption

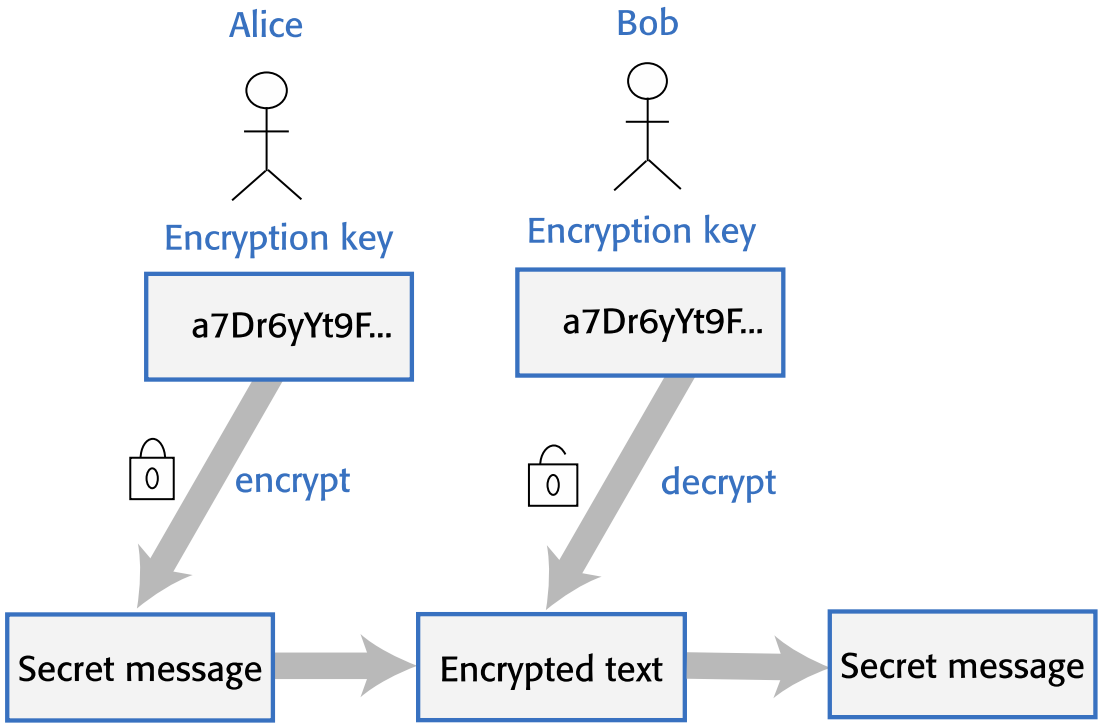
- Encryption is the process of making a document unreadable by applying an algorithmic transformation to it
- A secret key is used by the encryption algorithm as the basis of this transformation. You can decode the encrypted text by applying the reverse transformation

Encryption and Decryption



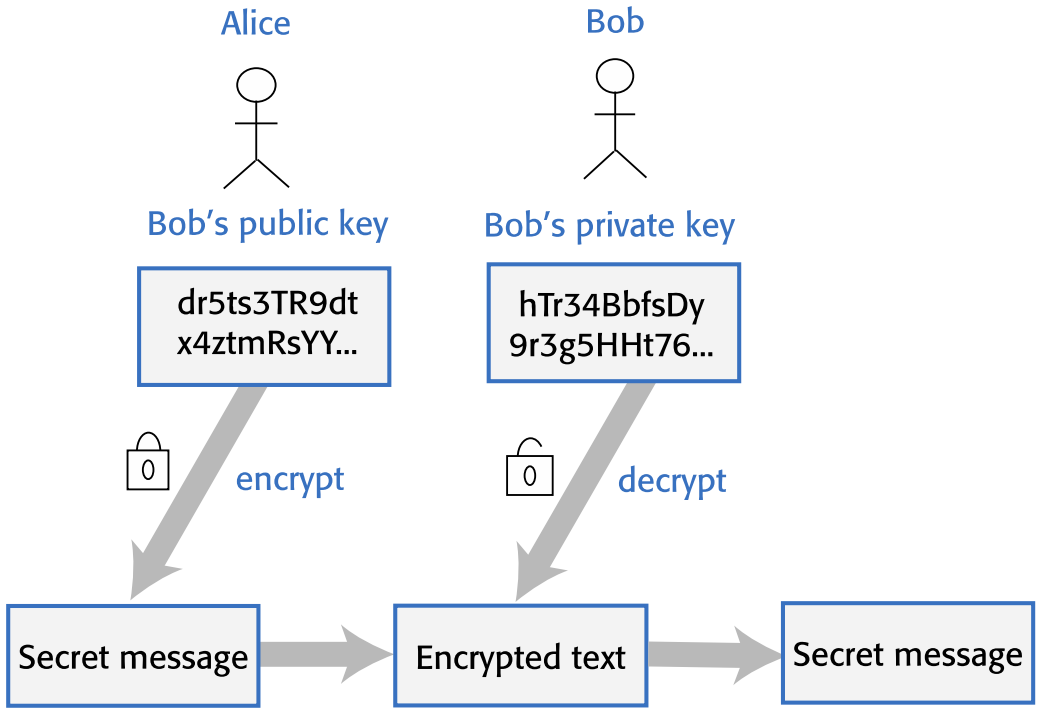
Symmetric Encryption

- In a symmetric encryption scheme, the same encryption key is used for encoding and decoding the information that is to be kept secret
- If Alice and Bob wish to exchange a secret message, both must have a copy of the encryption key. Alice encrypts the message with this key. When Bob receives the message, he decodes it using the same key to read its contents
- The fundamental problem with a symmetric encryption scheme is securely sharing the encryption key
- If Alice simply sends the key to Bob, an attacker may intercept the message and gain access to the key. The attacker can then decode all future secret communications

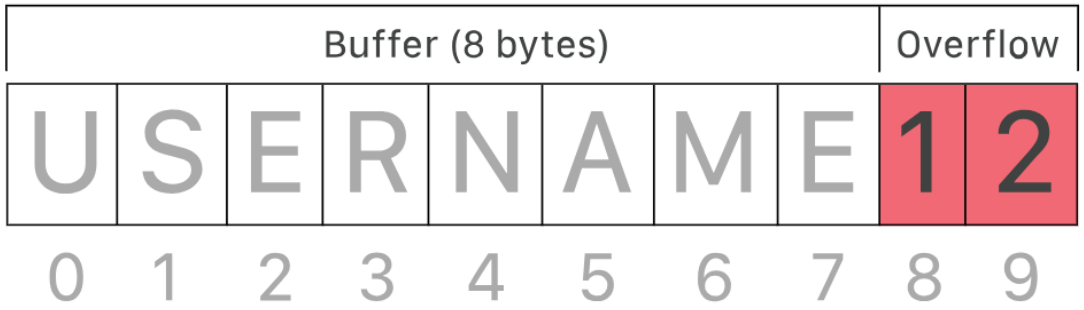


Asymmetric Encryption

- Asymmetric encryption, does not require secret keys to be shared
- An asymmetric encryption scheme uses different keys for encrypting and decrypting messages
- Each user has a public and a private key. Messages may be encrypted using either key but can only be decrypted using the other key
- Public keys may be published and shared by the key owner. Anyone can access and use a published public key
- However, messages can only be decrypted by the user's private key, so is only readable by the intended recipient



Buffer Overflow Attack

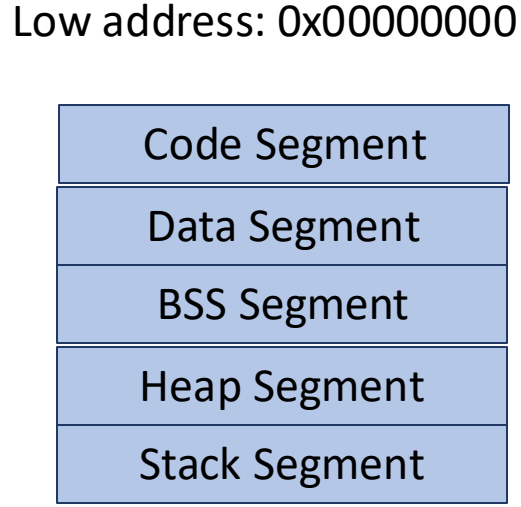


- It is possible when systems are programmed in C or C++
  - ✓ These languages do not automatically check that an assignment to an array element is within the array bounds
  - ✓ You can declare a buffer as an array of a specific size, but the run-time system does not check whether an input exceeds the length of that buffer
  - ✓ An attacker who understands how the system memory is organized can carefully craft an input string that includes executable instructions
  - ✓ This overwrites the memory, and if a function return address is also overwritten, control can then be transferred to the malicious code
- Most programming languages check for buffer overflows at run time and reject long, malicious inputs
  - ✓ Operating systems and libraries are often written in C or C++
  - ✓ However. If inputs are passed directly from your system to an underlying system function, buffer overflow is a possibility

Buffer Overflow Attack

Memory Layout

- **Text (code) segment:** program code and fixed program constants
- **Data segment:** initialized global and static variables
- **BSS (Block Started by Symbol) segment:** uninitialized global and static variables
- **Heap segment:** dynamic program data (e.g., malloc)
- **Stack Segment:** function local variables, arguments, context (calling function return address/stack frame) – can also contain code!



Buffer Overflow Attack

```c
int x =100;

int main()
{
    int a = 2;
    float b = 2.5;
    static int y;

    int *ptr = (int *) malloc (2*sizeof(int));

    ptr[0]=5;
    ptr[1]=6;

    free(ptr);

    return 1;
}
```

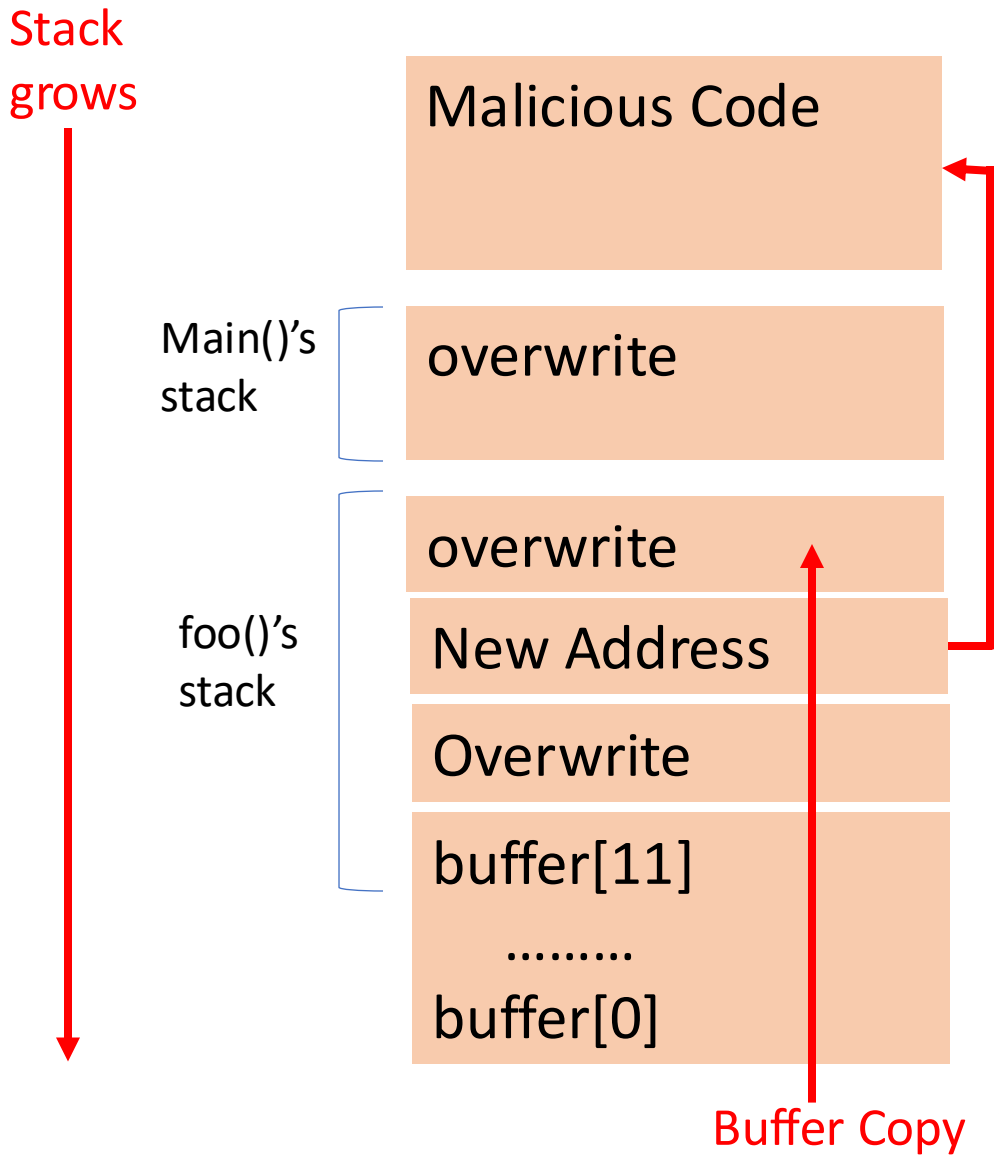
Data Segment: x

BSS Segment: y

Stack Segment: a, b, ptr

Heap Segment: ptr[0], ptr[1]

Buffer Overflow Attack: An Example




```
void foo (char *str){
        char buffer[12];
        strcpy(buffer, str);
}
```

```
int main (){
        char *str = "Some text to overflow buffer
                    [new address] some other text
                    [malicious code]" ;
        foo(str);
}
```

Injection Attack

- Injection attacks are a type of attack where a malicious user uses a valid input field to input malicious code or database commands
- These malicious instructions are then executed, causing some damage to the system
- Code can be injected that leaks system data to the attackers
- Common types of injection attacks include buffer overflow attacks and SQL poisoning attacks

```
SELECT * FROM AccountHolders WHERE accountnumber = '34200645'
```

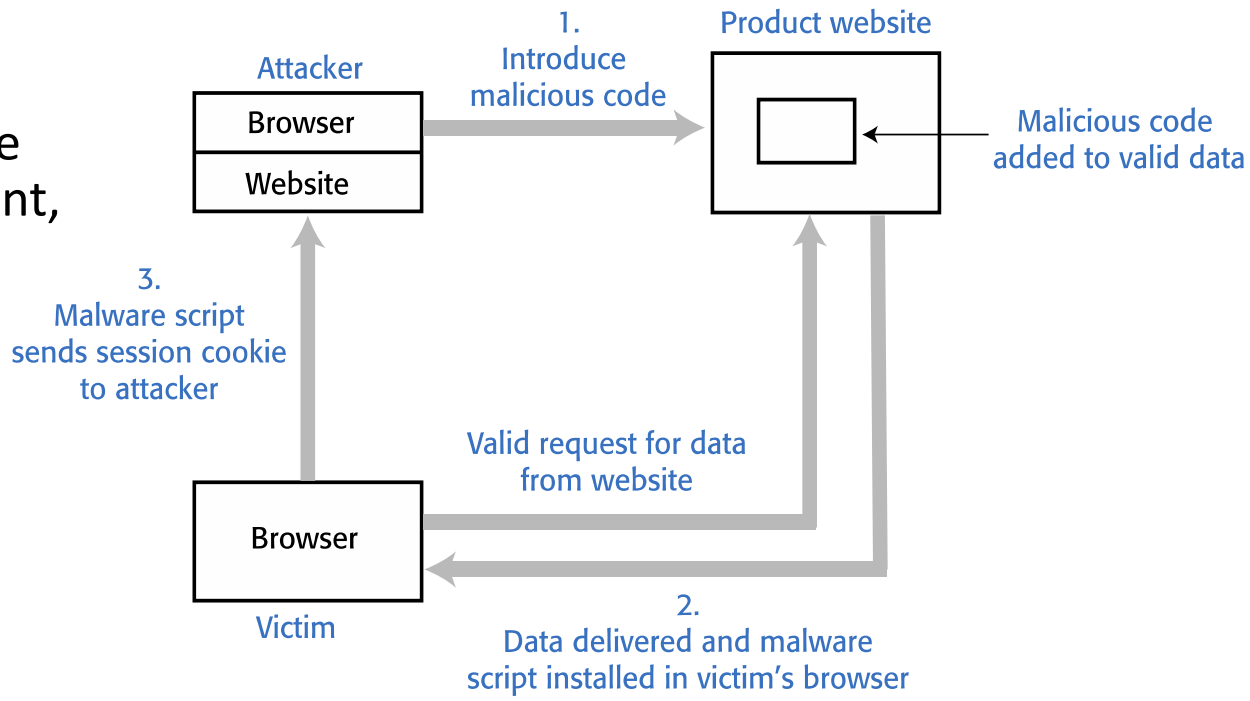
```
accNum = getAccountNumber ()
SQLstat = "SELECT * FROM AccountHolders WHERE accountnumber = '"
+ accNum + "';"
database.execute (SQLstat)
```

```
SELECT * from AccountHolders WHERE accountnumber = '10010010' OR '1' = '1';
```

```
SELECT * from AccountHolders
```

Cross-site Scripting Attack

- Cross-site scripting attacks are another form of injection attack
- An attacker adds malicious Javascript code to the web page that is returned from a server to a client, and this script is executed when the page is displayed in the user's browser
- The malicious script may steal customer information or direct them to another website
- This may try to capture personal data or display advertisements
- Cookies may be stolen, which makes a session-hijacking attack possible
- As with other types of injection attacks, cross-site scripting attacks may be avoided by input validation



Session Hijacking Attack

- When a user authenticates with a web app, a session is created
  - A session is a time period during which the user's authentication is valid. They don't have to re-authenticate for each interaction with the system
  - The authentication process places a session cookie on the user's device
- Session hijacking is a type of attack where an attacker gets hold of a session cookie and uses this to impersonate a legitimate user
- An attacker can find out the session cookie value
  - In cross-site scripting, the installed malware sends session cookies to the attackers
  - In traffic monitoring, attackers capture the traffic between the client and server. The session cookie can be identified by analyzing the data exchanged