CYBR3000

Cybersecurity Overview

Dr Dan Kim

Associate Professor in Cybersecurity,
School of EECS
The 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
  - Computer Security Concepts
  - Threats, Attacks, and Assets
  - Security Functional Requirements
  - Fundamental Security Design Principles
  - Attack Surfaces and Attack Trees
  - Computer Security Strategy

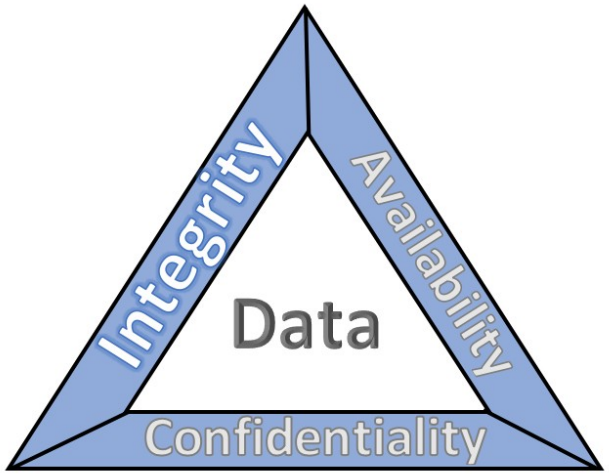
Computer Security definition?

- The NIST* Computer Security Handbook defines the term Computer Security as:
  - “The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources” (includes hardware, software, firmware, information/data, and telecommunications).

***National Institute of Standards and Technology**

The CIA triad

- represents the three pillars of information security: confidentiality, integrity, and availability, as follows.
  - Confidentiality – preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information
  - Integrity – guarding against improper information modification or destruction and ensuring information non-repudiation and authenticity
  - Availability – ensuring timely and reliable access to and use of information



https://www.nccoe.nist.gov/publication/1800-26/VolA/index.html

Security objectives (cont.)

An example

Internet

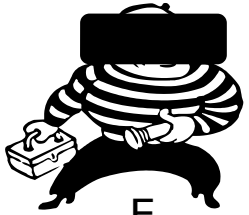
Alice in Brisbane

UGETGV

Bob in Sydney

by modern ciphers DES, 3DES, AES, *etc*

by a Classical Cipher named Caesar with *k=2*



E

The data has not been viewed by a 3rd party

C*onfidentiality*

Encryption

C*onfidentiality: the protection of transmitted data from passive attacks (release of message contents and traffic analysis)*

Security objectives (cont.)

Use Hashed message authentication code (HMAC), such as HMAC-MD5, HMAC-SHA1

Internet

Alice in Brisbane

Bob in Sydney

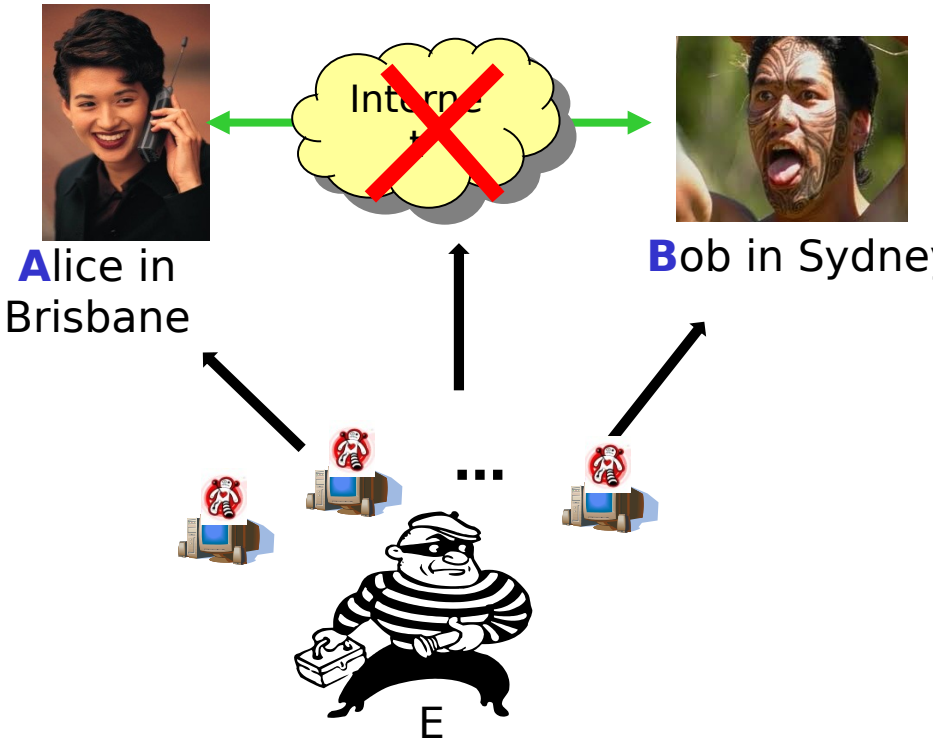
Unclassified code3000 - I hate you!

E

The data has not been modified in transit → *Integrity* ← Crytographic Hash func.

***I**ntegrity: the assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay)*

Security objectives (cont.)



Distributed Denial of Service (DDoS) attacks

For any information system to serve its purpose, the information must be **available** when it is needed

→ **A**vailability

Source: http://memeburn.com

Security objectives : summary

Internet

Alice in Brisbane

Bob in Sydney

Cryptography

The data has not been viewed by a 3rd party → ***C**onfidentiality* ← Encryption

The data has not been modified in transit → ***I**ntegrity* ← Hash func.

The data must be **available** when it is needed → **A**vailability

Additional Security Goals

- Authenticity:
  - The property of being genuine and being able to be verified and trusted
  - This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.
  - FIPS* PUB 199 (i.e., NIST standards for security categorization) includes authenticity under integrity.

- Accountability:
  - The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.
  - This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action.
  - truly secure systems aren't yet an achievable goal, we must be able to trace a security breach to a responsible party.

*The Federal Information Processing Standards **Publication** Series of the National Institute of. Standards and Technology (NIST)
Web: https://csrc.nist.gov/publications/detail/fips/199/final

University example?

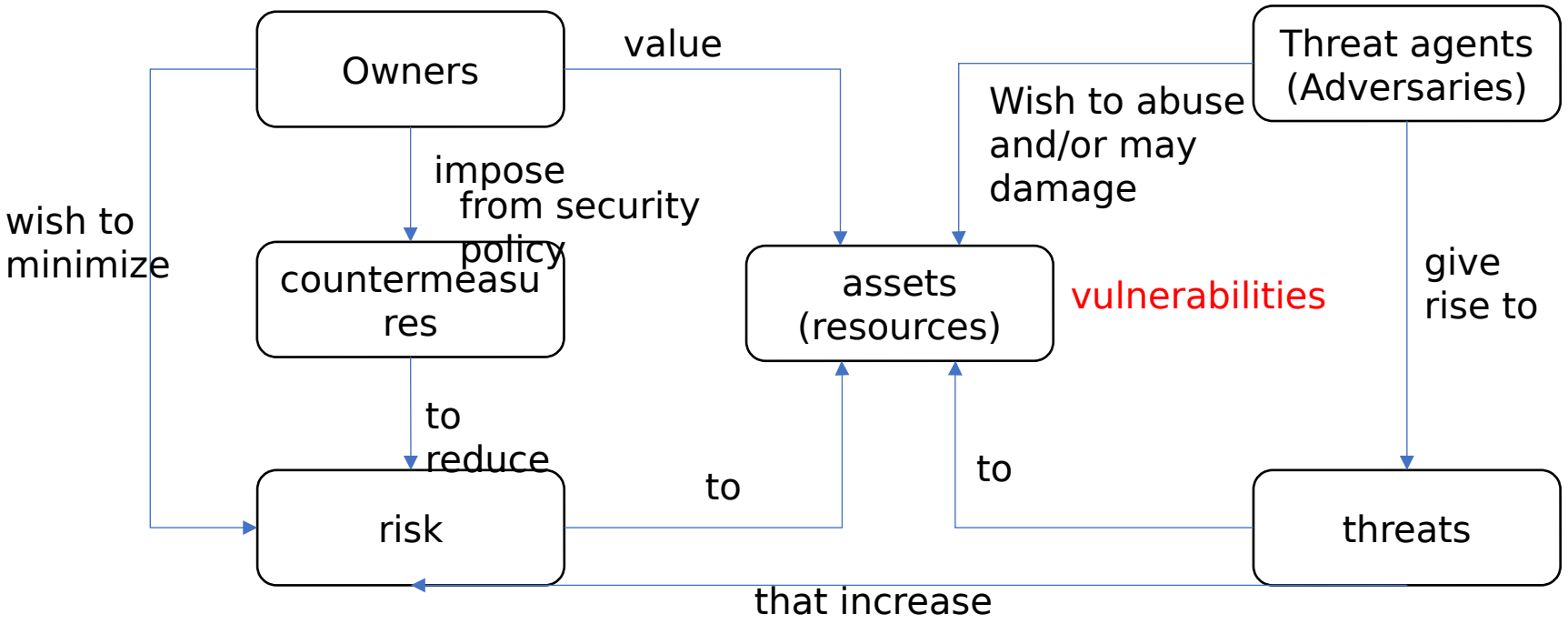
- Confidentiality
  - Student login password should not be improperly disclosed by others
  - e.g., ID: abc123 password: p@ssw0rd!
  - Q: any other examples?
- Integrity
  - Student name should not be modified improperly
  - e.g., John -> Jonathan
  - Q: any other examples?
- Availability
  - **Learn** (blackboard) system should be available when a student wants to download lecture slides.
  - Q: any other examples?

Availability - CrowdStrike outage

- considered the largest IT outage in history affecting millions of Windows systems around the world.
- occurred July 19, 2024, with millions of Windows systems failing and showing the infamous blue screen of death (BSOD).
- The flaw in CrowdStrike Falcon was inside of a sensor configuration update. The sensor is regularly updated -- sometimes multiple times daily -- to provide users with mitigation and threat protection.
- Microsoft estimated that approximately 8.5 million Windows devices were directly affected by the CrowdStrike logic error flaw. That's less than 1% of Microsoft's global Windows install base.
- Services affected include the following.
  - **Airlines and airports:** The outage grounded thousands of flights worldwide, leading to significant delays and cancellations of more than 10,000 flights around the world. In the United States, affected airlines included Delta, United and American Airlines. These airlines were forced to cancel hundreds of flights until systems were restored. Globally, multiple airlines and airports were affected, including KLM, Porter Airlines, Toronto Pearson International Airport, Zurich Airport and Amsterdam Schiphol Airport.
  - **Public transit:** Public transit in multiple cities was affected, including Chicago, Cincinnati, Minneapolis, New York City and Washington, D.C.
  - **Healthcare:** Hospitals and healthcare clinics around the world faced significant disruptions in appointment systems, leading to delays and cancellations. Some states also reported 911 emergency services being affected, including Alaska, Indiana and New Hampshire.
  - **Financial services:** Online banking systems and financial institutions around the world were affected by the outage. Multiple payment platforms were directly affected, and there were individuals who did not get their paychecks when expected.
  - **Media and broadcasting:** Multiple media and broadcast outlets around the world, including British broadcaster Sky News, were taken off the air by the outage.

https://www.techtarget.com/whatis/feature/Explaining-the-largest-IT-outage-in-history-and-whats-next

Security concepts and Relationships



Computer Security Terminology

- System Resource (Asset)
  - Data contained in an information system;
  - or a service provided by a system;
  - or system capability, such as processing power or communication bandwidth;
  - or an item of system equipment (i.e., a system component – hardware, firmware, software, or documentation);
  - or a facility that houses system operations and equipment
- Vulnerability
  - A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy
- Security policy
  - A set of rules and practices that specify or regulate how a system or organization provides security services to protect sensitive and critical system resources

Request for Comments (RFC) 4949, Internet Security Glossary, May 2000

Computer Security Terminology (cont.)

- Adversary (Threat agent)
  - An entity that attacks, or is a threat to, a system
- Threat
  - A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm; That is, a threat is a possible danger that might exploit a vulnerability
- Attack
  - An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system

Computer Security Terminology (cont.)

- Intrusion:
  - an attack that succeeds (or all the activities of violation of C.I.A in IDS)
- Countermeasure / control
  - An action, device, procedure, or technique that reduces a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken
- Risk
  - An expectation of loss expressed as the probability that a particular threat will exploit a particular vulnerability with a particular harmful result.

Attacks and their classifications

Alter?

- Passive:
  - attempt to learn or make use of information from the system that does not affect system resources; eavesdropping on, or monitoring of, transmissions;
  - Two types: release of message contents; traffic analysis
- Active
  - attempt to alter system resources or affect their operation
  - Four categories: Replay, Masquerade, Modification of messages, Denial of service

Origin?

- Inside
  - initiated by an entity inside the security perimeter
- Outside
  - initiated from outside the perimeter

Countermeasure to Security Vulnerabilities and Threats
in terms of Security Requirements
(FIPS PUB 200)

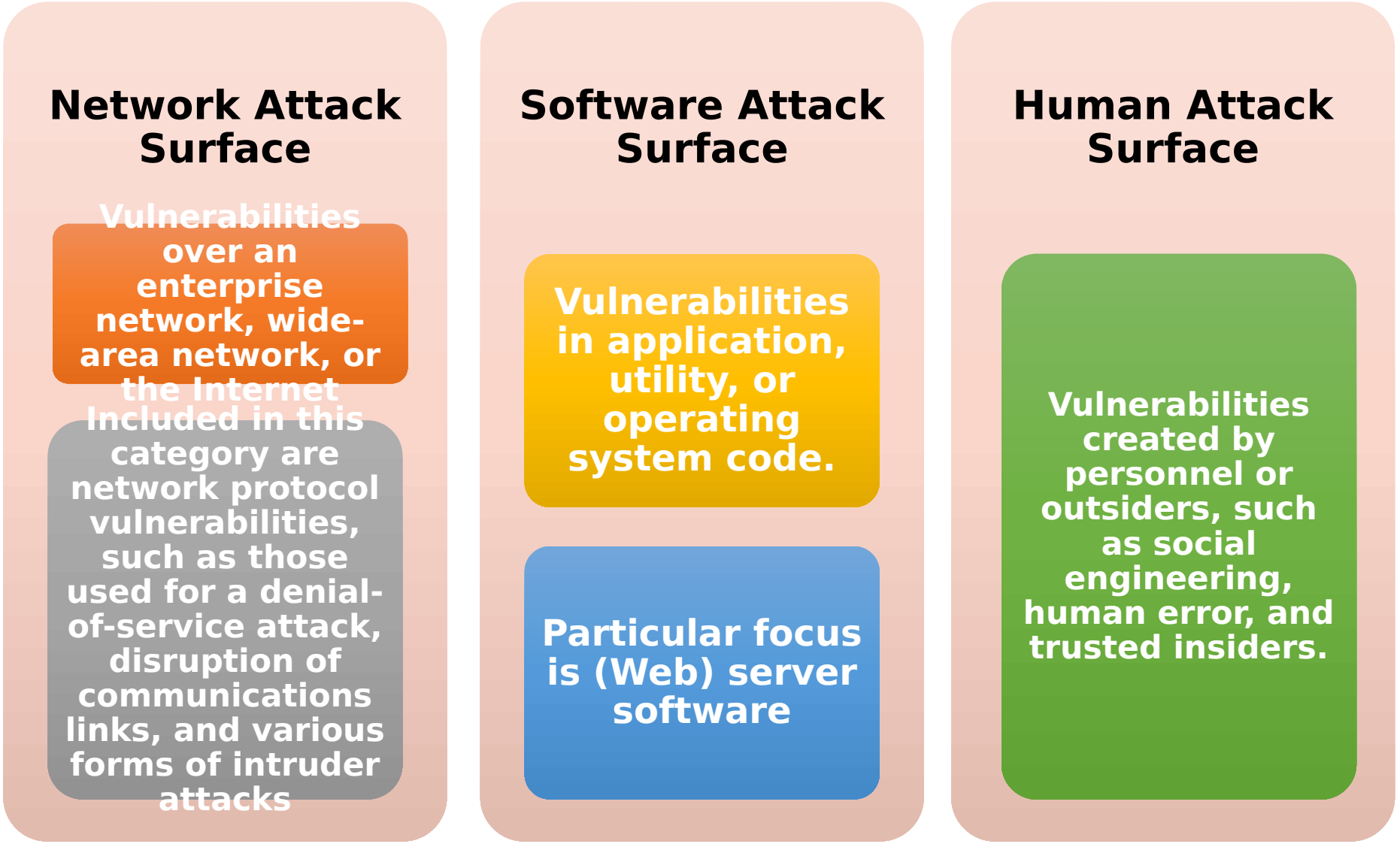
*Minimum Security Requirements for Federal Information Processing Standards Publications (17 security-related areas w.r.t. protecting C.I.A.)

- Technical measures
  - Access control; identification & authentication; system & communication protection; system & information integrity
- Management controls and procedures
  - Awareness & training; audit & accountability; certification, accreditation, & security assessments; contingency planning; maintenance; physical & environmental protection; planning; personnel security; risk assessment; systems & services acquisition
- Overlapping technical and management
  - Configuration management; incident response; media protection (both digital & paper)

Attack Surfaces

- consists of the reachable and exploitable vulnerabilities in a system
- Examples
  - Open ports on outward facing Web and other servers, and code listening on those ports
  - Services available on the inside of a firewall
  - Code that processes incoming data, email, XML, office documents, and industry-specific custom data exchange formats Interfaces, SQL, and Web forms
  - **An employee** with access to sensitive information vulnerable to a social engineering attack

Attack Surfaces: categories



Attack Surfaces vs. Attack Vectors

- Attack Surface = **Where** an attacker can get in
  - It refers to all the possible points in a system that could be exploited.
  - Focuses on the total exposure or set of vulnerabilities.
  - Examples:
    - ✓ Open ports
    - ✓ Public-facing APIs
    - ✓ Login pages

- Attack Vector = **How** an attacker gets in
  - It refers to the **method or pathway** used to exploit a vulnerability.
  - Focuses on the technique or entry point.
  - Examples:
    - ✓ Phishing email
    - ✓ Malware download
    - ✓ SQL injection
    - ✓ Exploiting weak passwords

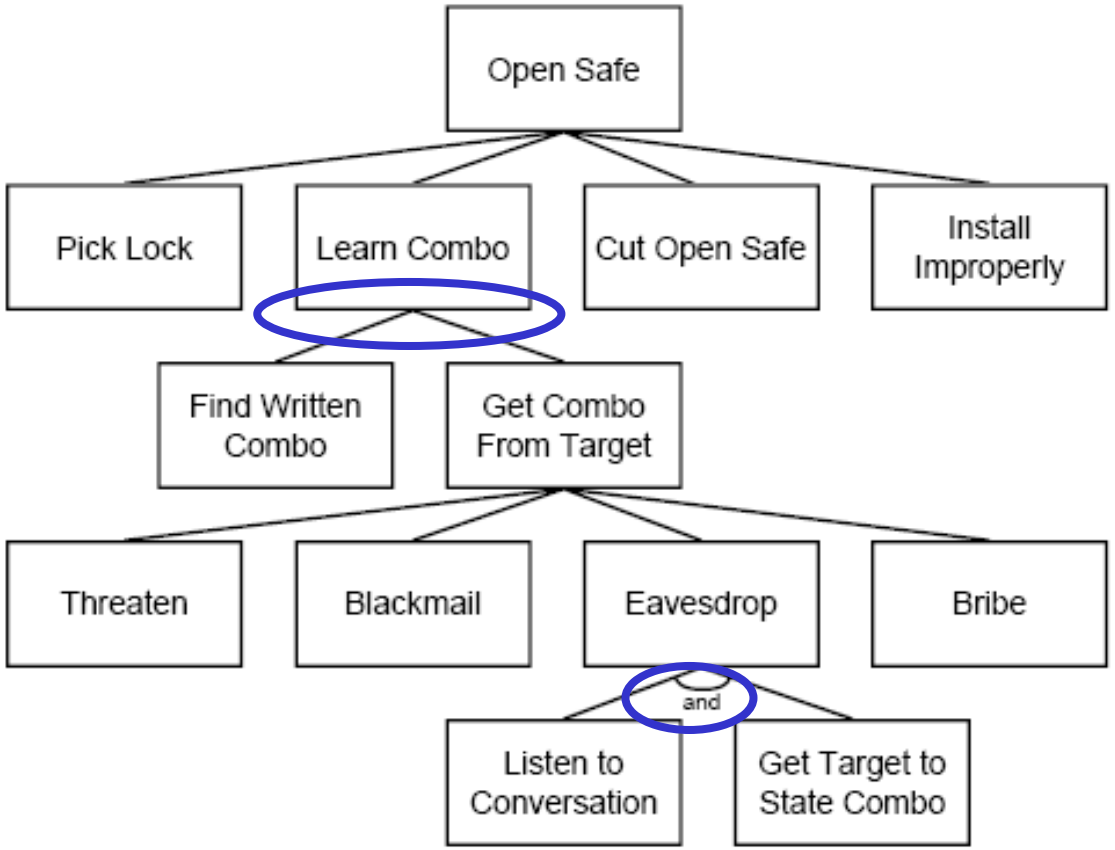
Attack trees

- A branching, hierarchical data structure that represents a set of potential vulnerabilities (events)

- Objective: to effectively exploit the info available on attack patterns
  - published on the US CERT* or similar forums
  - Security analysts can use the tree to guide design and strengthen countermeasures

*Cyber Emergency Response Team; aka, Computer Emergency Response Team (CERT) in the past.

An Attack Tree for “open safe”



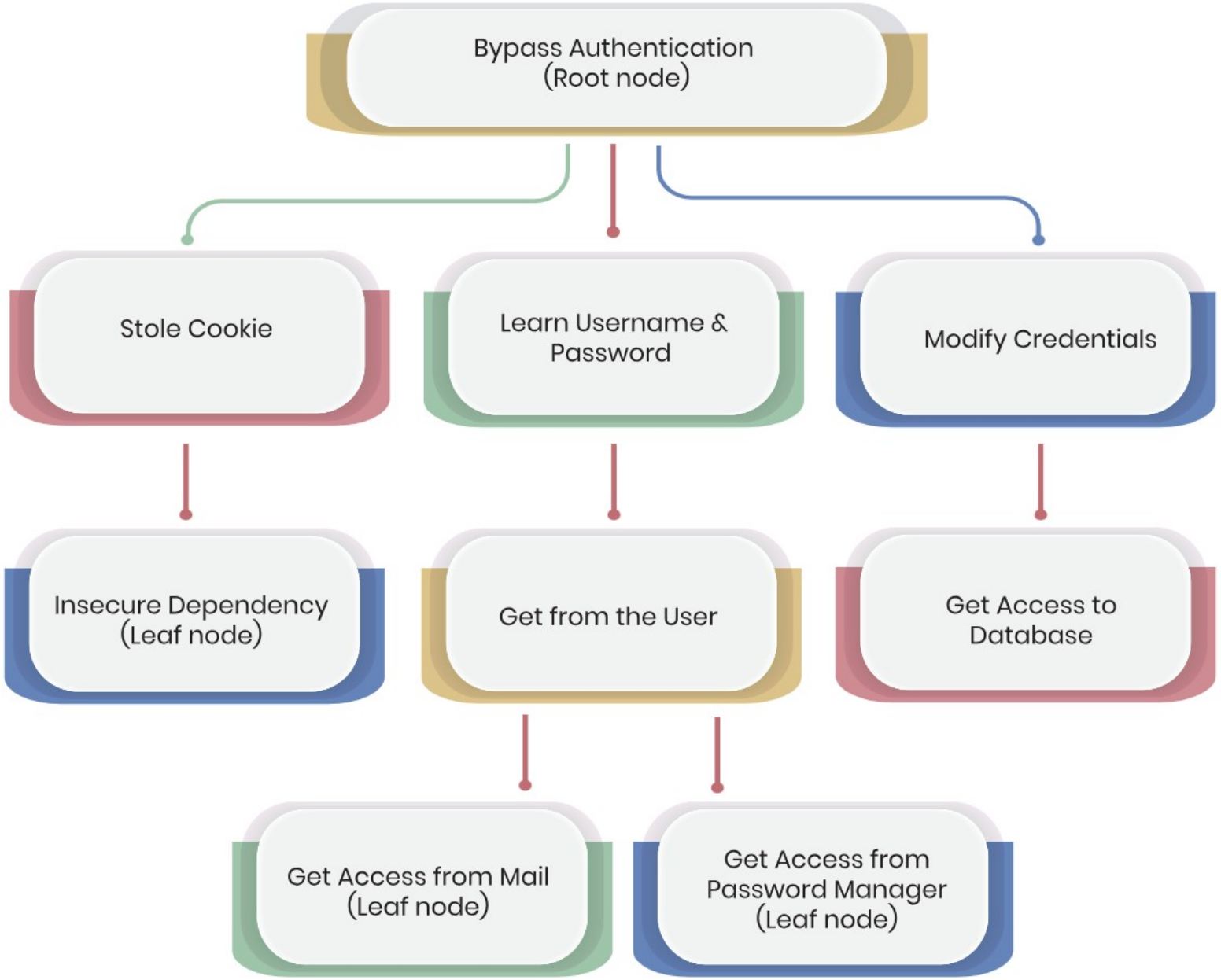
Source: B. Schneier, https://www.schneier.com/academic/archives/1999/12/attack_trees.html

Attack Trees? How practical?

- National Cyber Security Centre (NCSC) in UK used Attack Trees to **identify cyber security risks** in their security analysis for the UK telecoms sector.
- This involved identifying higher level impacts or outcomes and linking these to lower-level methods or exploitation routes that could contribute to such events occurring.

https://www.ncsc.gov.uk/collection/risk-management/using-attack-trees-to-understand-cyber-security-risk

https://www.ncsc.gov.uk/collection/risk-management/using-attack-trees-to-understand-cyber-sec



https://www.ncsc.gov.uk/collection/risk-management/using-attack-trees-to-understand-cyber-sec

Fundamental Security Design Principles to guide the development of protection mechanisms

- Economy of mechanism
- Fail-safe defaults
- Complete mediation
- Open design
- Separation of privilege
- Least privilege
- Least common mechanism
- Psychological acceptability
- Isolation
- Encapsulation
- Modularity
- Layering
- Least astonishment

Fundamental Security Design Principles (1/3)

- Economy of Mechanism:
  - Security mechanism should be as simple as possible.
  - Example: simple login mechanisms - a login system that uses only username and password
- Fail-Safe Defaults:
  - Systems should default to a secure state, denying access by default and granting access only when explicit permission is given.
  - Examples: most file access permissions work this way;
    - ✓ Windows access control list (ACL), Linux/Unix permissions
    - ✓ Firewalls (in the FW figure later)
- Complete mediation
  - **Every** access to **every** resource should be checked to ensure it is allowed.
  - Example: the operating system checks the user requesting access against the file's ACL.
- Open design
  - Security of a mechanism should not depend upon secrecy of its design or implementation.
  - Should be open for scrutiny (critical observation or examination) by the community
  - Examples: cryptography and openness
    - ✓ TLS (Transport Layer Security) is an open protocol with publicly available specifications. Its security is based on strong cryptographic algorithms, not hidden designs.
    - ✓ Open-source encryption libraries like OpenSSL are reviewed by many researchers, increasing trust and accountability.

Fundamental Security Design Principles (2/3)

- Separation of privilege
  - System should not grant permission based on single condition; Access should be based on multiple conditions or credentials
  - Access to objects should depend on more than one condition being satisfied
  - Examples: a company checks over $75,000 to be signed by two officers.; **Two-person control** used in launching nuclear weapons or accessing top-secret cryptographic materials.
- Least privilege
  - Entity (users and systems) should be given only those privilege needed to finish a task
  - Example: The Unix sudo command allows users to temporarily elevate privileges.
- Least Common Mechanism:
  - Mechanisms used to access resources should be minimized to avoid sharing that might lead to vulnerabilities.
  - Examples: In web hosting, using separate database connections for each tenant in a multi-tenant application instead of a single shared connection; In a system where multiple users process sensitive data, using separate logging mechanisms prevents one user's logs from leaking into another's.
- Psychological acceptability
  - the idea that the security mechanisms of a computer system should align as closely as possible to the functional expectations of system users.
  - By providing security mechanisms that do not burden or inconvenience users, architects can achieve security without alienation users or encouraging them to find ways to avoid security mechanisms.
  - Examples: Single Sign-On (SSO): Allows users to authenticate once and access multiple services, reducing password fatigue; Biometric login (e.g., fingerprint or facial recognition) on smartphones provides strong security without requiring users to type complex passwords frequently.

Fundamental Security Design Principles (3/3)

- Isolation
  - Public access should be isolated from critical resources (no connection between public and critical information) (e.g., net separation for critical infrastructure)
  - Examples: One user's files should be isolated from one another (except when desired); security mechanism should be isolated (i.e., preventing access to those mechanisms) (FW, IDS, access control could be targets of attacks)
- Encapsulation
  - A design principle that separates the internal workings of a component from its external interface, improving modularity and security; hiding the internal details.
  - Example: a web application hides its database implementation details behind an API, preventing direct access to the database and allowing only controlled interactions.
- Modularity
  - Systems should be composed of separate, self-contained components that can be independently developed, tested, and replaced.
  - dividing a system into separate components or modules that are independent and interchangeable.
  - Example: A smartphone app separates its messaging, contacts, and settings features into independent modules that can be updated without affecting each other.
- Layering (defense in depth):
  - The use of multiple layers of security controls to protect resources, ensuring that the failure of one control does not lead to a complete compromise.
  - Example: a corporate network uses multiple security layers: a firewall, intrusion detection system (IDS), antivirus software, and role-based access control. If one layer fails, others still protect the system.
  - Layering uses multiple stacked controls to provide defense in depth while Separation of Privilege requires multiple conditions for access.
- Least astonishment
  - a program or interface should always respond in a way that is least likely to astonish a user
  - Examples: A password reset feature that sends a confirmation email rather than immediately changing the password aligns with user expectations and avoids surprising or insecure behavior.
  - Anti-example: A "Delete" button that immediately erases data without any confirmation or undo option, causing unexpected loss.

Do I memorise every principle?
Fundamental Security Design Principles - Question

- **Question 1:**
  Which security design principle states that a system should grant access only after multiple conditions are met, rather than relying on a single condition?
  A) Least Privilege
  B) Separation of Privilege
  C) Economy of Mechanism
  D) Fail-Safe Defaults
  E) Defense in Depth
- Answer: B) Separation of privilege

Fundamental Security Design Principles - Question

- **Question 2:**
  Which principle emphasizes that security mechanisms should be as simple as possible to reduce errors and ease maintenance?
  A) Open Design
  B) Psychological Acceptability
  C) Economy of Mechanism
  D) Least Common Mechanism
  E) Fail-Safe Defaults

- **Answer:** C) Economy of Mechanism

Fundamental Security Design Principles - Question

- **Question 3**
  Which of the following best illustrates the principle of **Layering (Defense in Depth)**?
  A) Using a single firewall to protect a network
  B) Relying only on antivirus software for malware protection
  C) Implementing a firewall, intrusion detection system, and access controls together
  D) Giving all users full admin access to simplify management
  E) Encrypting data at rest without network security controls

- **Answer:** C) Implementing a firewall, intrusion detection system, and access controls together

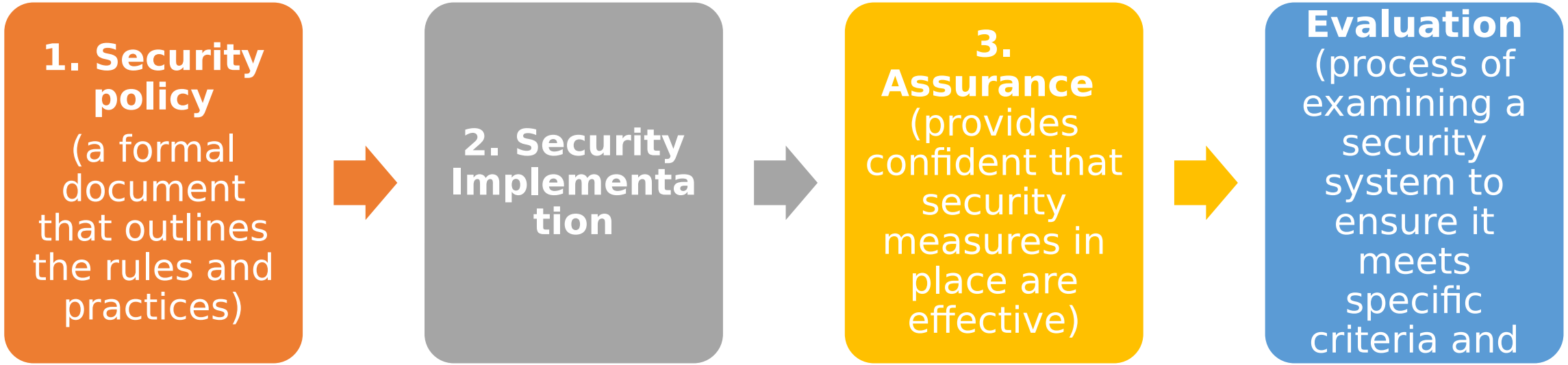
Fundamental Security Design Principles - Question

- **Question 4**
- Which scenario is an example of **poor modularity** in system design? Choose one answer only.
  A) Developing independent services for user login, payment, and notifications
  B) Combining all application features into one large, tightly coupled codebase
  C) Designing components with clear, well-defined interfaces
  D) Breaking program logic into smaller, reusable functions
  E) Allowing separate teams to work on well-defined modules independently

- Answer: B)

Computer Security Strategy

to devise security services and mechanisms (simple view)



Computer Security Strategy
to devise security services and mechanisms (textbook version)

A security manager needs to consider the following factors:

- The value of the assets being protected
- The vulnerabilities of the system
- Potential threats and the likelihood of attacks

An ideal security scheme is one in which no attack is successful Absolute protection is not feasible, but it is practical to detect security attacks

If security mechanisms detect an ongoing attack, the system may be able to respond in such a way as to halt the attack and prevent further damage.

Assurance deals with the questions, “Does the security system design meet its requirements?” and “Does the security system implementation meet its specifications?”

Assurance is about gaining confidence *before or during deployment* that controls work correctly

Evaluation involves testing and may also involve formal analytic or mathematical techniques.

Evaluation is about *monitoring and reviewing* **after** implementation to ensure controls remain effective.

Example: Applying Policy → Implementation → Assurance → Evaluation

1. Security Policy:

- *Defines the rule or requirement.*
- "All users must authenticate using multi-factor authentication (MFA) before accessing corporate email."

2. Security Implementation:

- *Describes the technical and operational steps taken to enforce the policy.*
- Deploy an MFA system integrated with the corporate email server.
- Configure the email platform to require MFA for all logins.

3. Assurance:

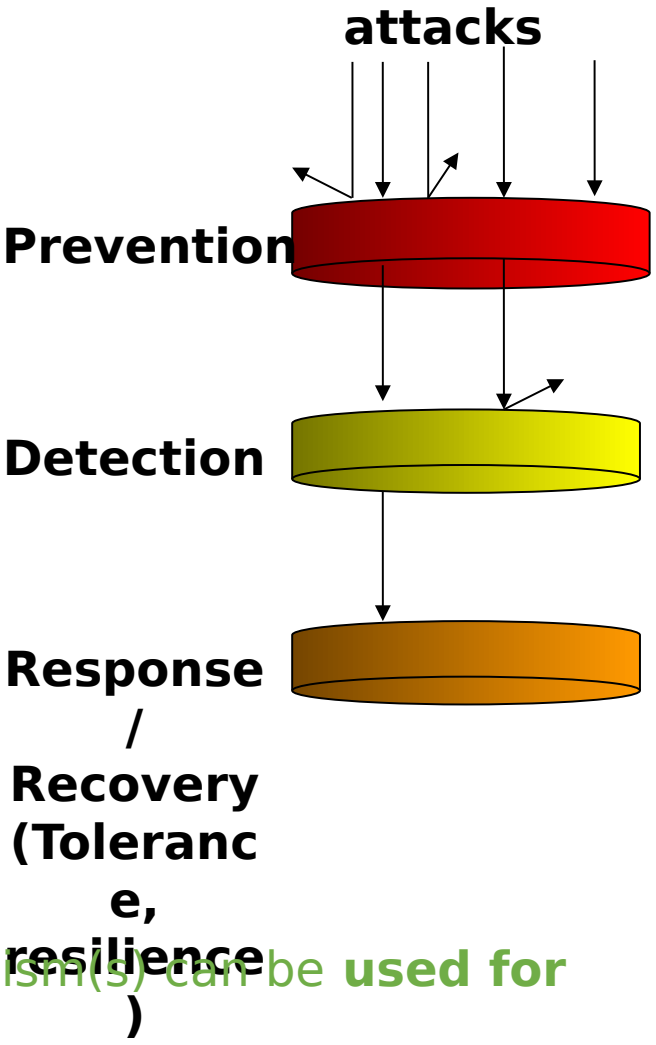
- *Builds confidence that the implementation is secure and functions as intended.*
- Perform testing and verification to confirm MFA is enforced on all email accounts.
- Conduct penetration testing to ensure no bypass is possible.

4. Evaluation:

- Ensures ongoing effectiveness and identifies areas for improvement.
- Continuously monitor authentication logs to detect any failed or suspicious login attempts.
- Periodically audit compliance reports to confirm all users have MFA enabled.
- Review user feedback and incident reports to assess effectiveness and usability.

Security mechanisms/implementation

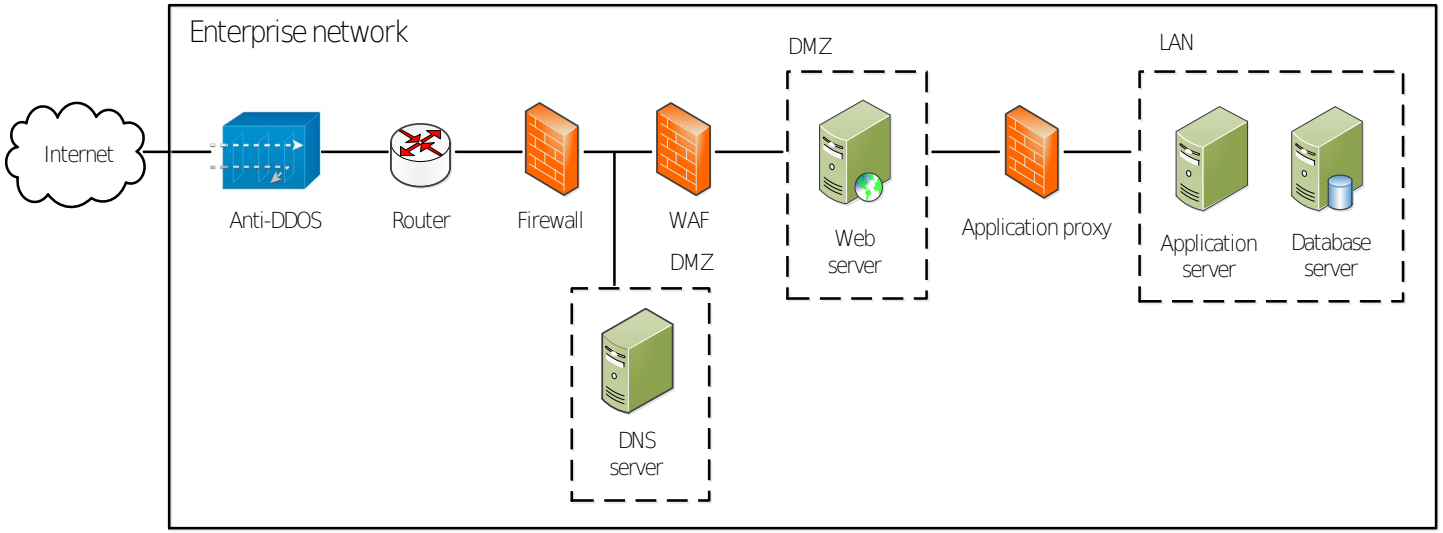
- Prevention
  - ✓ Example: encryption to prevent unauthorized access to data, access control (e.g., firewall, password/fingerprint)
- Detection
  - ✓ Example: Auditing and intrusion detection (e.g., Intrusion Detection System, forensics)
- Response
  - ✓ Example: halt the detected attack and prevent further damage
- Recovery
  - ✓ Data backup and reload correct copy of data
  - ✓ intrusion tolerance (e.g., Intrusion Tolerance System), backup



Q: What prevention, detection, response and recovery mechanism(s) can be **used for Multi Factor Authentication (MFA)?**

Q: What prevention, detection, response and recovery mechanism(s) can be used against a specific attack (e.g., DDoS attacks, Ransomware)?

An example of security mechanisms



- DDoS: Distributed Denial of Service attacks
- WAF: Web Application Firewall
- DMZ: Demilitarized Zone

Q: What security mechanisms were applied to this network?

1.Anti-DDoS Protection – Mitigates Distributed Denial of Service attacks before they reach the network.

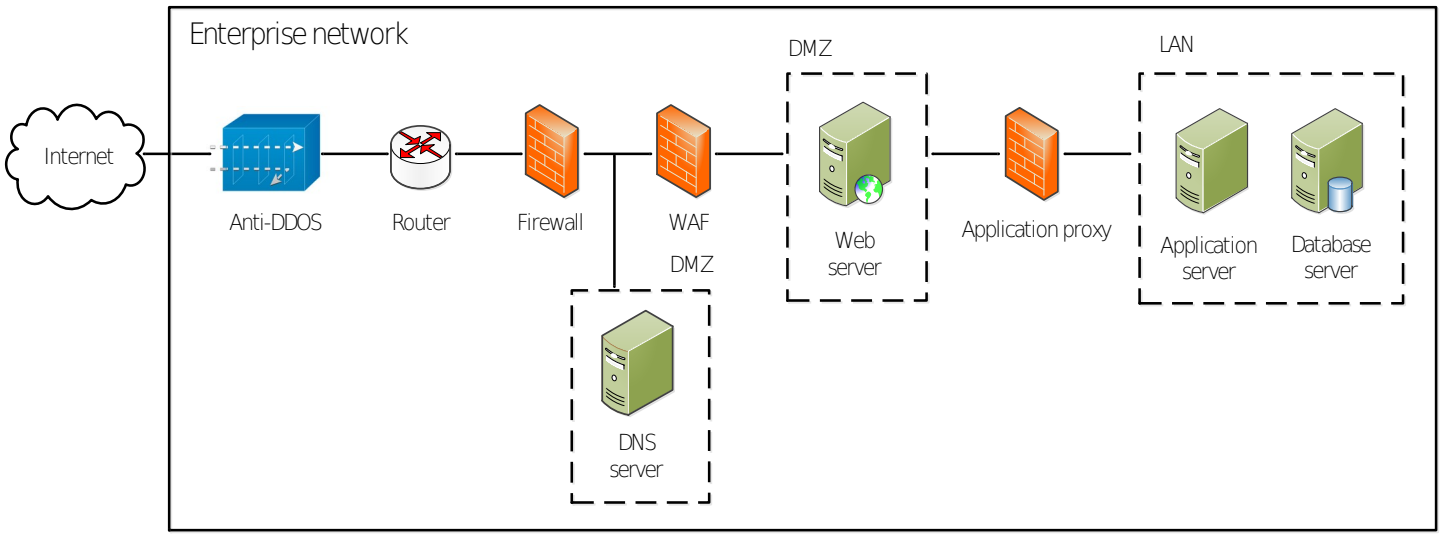
2.Firewall – Controls incoming and outgoing traffic based on security rules.

3.WAF (Web Application Firewall) – Protects web applications by filtering and monitoring HTTP traffic.

4.DMZ (Demilitarized Zone) – Isolates public-facing servers (Web and DNS) from the internal network.

5.Application Proxy – Acts as an intermediary for requests from clients seeking resources from servers.

An example of security mechanisms



Q: What fundamental Security principles were applied to this network?

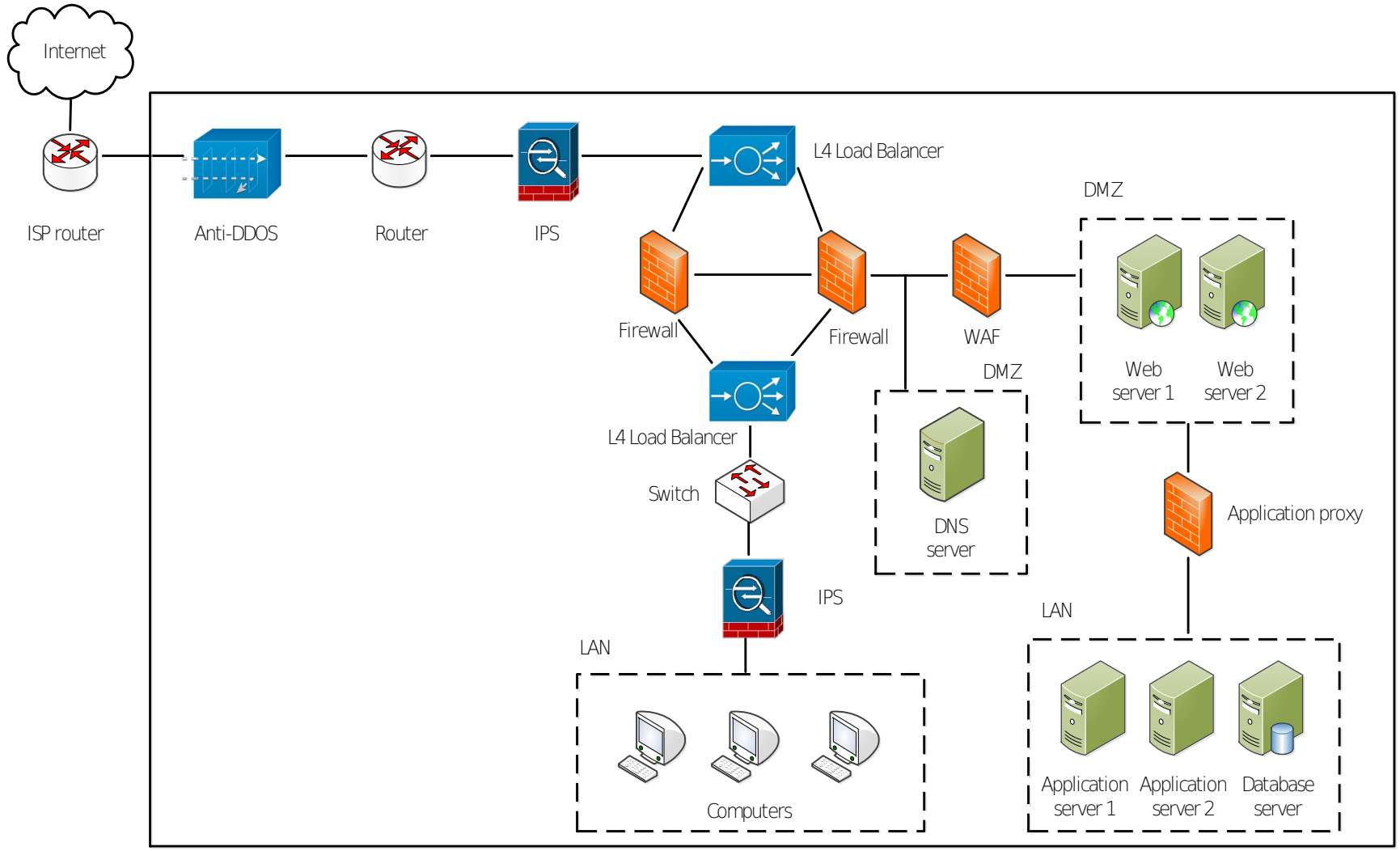
A: Layering (**Defense in Depth)**
Multiple layers of security (Anti-DDoS, Firewall, WAF, DMZ) provide redundancy and resilience against different types of threats.

A: Isolation:

- The **DMZ** isolates public-facing servers (Web and DNS) from the internal LAN, reducing the risk of external threats reaching sensitive systems.
- The **Application Proxy** acts as a boundary, isolating external users from direct access to internal application and database

- Economy of mechanism
- Fail-safe defaults
- Complete mediation
- Open design
- Separation of privilege
- Least privilege
- Least common mechanism
- Psychological acceptability
- Isolation
- Encapsulation
- Modularity
- Layering
- Least astonishment

An example of security mechanisms (cont.)



- IPS: intrusion prevention system

Summary

- Security concepts
- Terminologies
- Functional requirements
- Security design principles
- Security strategy
- Security mechanisms