Information Security (DI03016035)

Complete Lecture Deck – All 45 Lectures

Milav Dabgar

July 25, 2026

Table of Contents I

History of Information Security

- **Unit 1**: Introduction to Information Security
- Lecture 1
- **Course**: DI03016035

Agenda

- What is Information?
- The Early Days of Security
- ARPANET and First Vulnerabilities
- **The 1980s**: Viruses Emergence
- **The 1990s**: Internet Boom
- Modern Era of Cybersecurity

What is Information?

- Data that has been organized and processed.
- The most valuable asset of an organization.
- **Can exist in various forms**: printed, electronic, spoken.
- Needs protection from unauthorized access or alteration.

The Early Days of Security

- **Pre-1960s**: Security meant physical security.
- Focus on securing locations, filing cabinets, and mainframes.
- Only authorized personnel had physical access to computing facilities.
- Cryptography was primarily used by military and governments.

The 1960s: Timesharing and Multics

- Mainframes allowed multiple users via terminals.
- Need arose to protect user data from other users.
- MULTICS operating system introduced early security concepts.
- First conceptualization of access control and privilege levels.

The 1970s: ARPANET & First Vulnerabilities

- ARPANET developed by DoD for resilient communication.
- Focus was on connectivity, not security.
- Vulnerabilities discovered in remote access and file transfers.
- Introduction of early network security concepts.

The 1980s: Emergence of Viruses

- Personal computers became widespread.
- First PC viruses (e.g., Brain, Morris Worm) emerged.
- The Morris Worm (1988) crashed a significant portion of the internet.
- Formation of CERT (Computer Emergency Response Team).

The 1990s: The Internet Boom

- World Wide Web made internet accessible to the public.
- Explosion of e-commerce and digital transactions.
- Rise of firewalls, early antivirus software, and SSL/TLS.
- Hackers transitioned from hobbyists to financially motivated attackers.

The 2000s: Sophisticated Threats

- Rise of broadband internet meant always-on connections.
- Emergence of organized cybercrime and botnets.
- State-sponsored attacks and cyber warfare concepts began.
- Regulations (like HIPAA, SOX) introduced for data protection.

The Modern Era (2010s-Present)

- Cloud computing, IoT, and mobile devices expanded the attack surface.
- Ransomware became a multi-billion dollar criminal industry.
- AI and machine learning used for both defense and attacks.
- Zero Trust Architecture becomes the new standard.

Summary

- Security shifted from physical locks to network defense.
- The internet wasn't originally designed for security.
- Threats evolved from pranks to organized crime.
- Modern security requires continuous adaptation.

Need for Information Security

- **Unit 1**: Introduction to Information Security
- Lecture 2
- **Course**: DI03016035

Agenda

- What is Information Security?
- Business Continuity
- Protecting Data and Privacy
- Legal and Regulatory Compliance
- Reputation and Trust
- The Cost of a Breach

What is Information Security?

- The practice of protecting information from unauthorized access, use, disclosure, disruption, modification, or destruction.
- Applies to both physical and digital data.
- Aims to ensure business operations run smoothly.
- Involves people, processes, and technology.

Why Do We Need Security?

- To protect organizational assets.
- To ensure the organization can continue to operate (Business Continuity).
- To comply with laws and regulations.
- To maintain the trust of customers and partners.

Business Continuity

- Security incidents can cause system downtime.
- Downtime leads to lost revenue and productivity.
- Security ensures critical services remain available.
- Disaster Recovery Plans (DRP) help bounce back from incidents.

Protecting Data and Privacy

- Organizations hold sensitive customer data (PII, financial info).
- Intellectual property (trade secrets, designs) needs protection.
- A breach can lead to identity theft and financial loss for users.
- Privacy is increasingly viewed as a fundamental human right.

Legal and Regulatory Compliance

- Governments mandate strict data protection laws.
- **Examples**: GDPR (Europe), HIPAA (Healthcare), IT Act (India).
- Non-compliance results in massive fines and legal action.
- Organizations must prove they have adequate security controls.

Reputation and Trust

- Trust takes years to build and seconds to destroy.
- A public data breach severely damages brand reputation.
- Customers leave companies they cannot trust.
- Stock prices often plummet following a major breach.

The Cost of a Breach

- **Direct costs**: Fines, forensic investigations, legal fees.
- **Indirect costs**: Lost business, increased insurance premiums.
- The global average cost of a data breach is in the millions of dollars.
- Small businesses often go bankrupt after a major cyberattack.

Security as a Business Enabler

- Good security enables organizations to adopt new technologies (Cloud, AI).
- It allows secure remote work and global collaboration.
- Provides a competitive advantage in the market.
- It is a foundational element of digital transformation.

Summary

- Information security protects assets, operations, and people.
- Compliance and legal requirements mandate security.
- The financial and reputational costs of a breach are devastating.
- Security enables modern digital business.

Key Concepts: The CIA Triad

- **Unit 1**: Introduction to Information Security
- Lecture 3
- **Course**: DI03016035

Agenda

- Introduction to the CIA Triad
- Confidentiality
- Mechanisms for Confidentiality
- Integrity
- Mechanisms for Integrity
- Availability
- Mechanisms for Availability
- Balancing the Triad

What is the CIA Triad?

- The core model guiding information security policies.
- Stands for Confidentiality, Integrity, and Availability.
- Every security control aims to enforce one or more of these principles.
- Forms the benchmark for evaluating information systems.

1. Confidentiality

- Ensuring that information is accessible only to those authorized to have access.
- Preventing unauthorized disclosure.
- Equivalent to privacy in the context of personal data.
- **Example**: A hacker stealing credit card numbers breaches confidentiality.

Mechanisms for Confidentiality

- **Encryption**: Scrambling data so it cannot be read without a key.
- **Access Control Lists (ACLs)**: Defining who can view files.
- **Authentication**: Verifying user identity (Passwords, Biometrics).
- **Physical Security**: Locked doors, secure server rooms.

2. Integrity

- Ensuring that data is accurate, complete, and trustworthy.
- Protecting data from unauthorized modification or deletion.
- Data must remain unchanged in transit and at rest.
- **Example**: A student changing their grades in the database breaches integrity.

Mechanisms for Integrity

- **Hashing**: Creating a unique mathematical fingerprint of data.
- **Digital Signatures**: Proving who created the data and that it hasn't changed.
- **Audit Logs**: Tracking who made changes and when.
- **Version Control**: Keeping history of file modifications.

3. Availability

- Ensuring that information and systems are accessible to authorized users when needed.
- A system that is secure but unusable is not a good system.
- Protecting against disruptions like hardware failures or cyberattacks.
- **Example**: A website crashing due to a DDoS attack breaches availability.

Mechanisms for Availability

- **Redundancy/Backups**: Having multiple copies of data and servers.
- **Load Balancing**: Distributing traffic across multiple servers.
- **Uninterruptible Power Supplies (UPS)**: Providing backup power.
- **DDoS Protection**: Filtering malicious traffic to keep services running.

Balancing the CIA Triad

- The three principles often conflict.
- High confidentiality might reduce availability (too many passwords).
- High availability might reduce confidentiality (easier access).
- Security is about finding the right balance based on risk.

The DAD Triad (The Opposite)

- **Disclosure**: The opposite of Confidentiality.
- **Alteration**: The opposite of Integrity.
- **Destruction/Denial**: The opposite of Availability.
- Attackers aim to achieve the DAD triad.

Summary

- CIA stands for Confidentiality, Integrity, and Availability.
- Confidentiality = Secrecy.
- Integrity = Accuracy.
- Availability = Accessibility.
- All security mechanisms map back to these three goals.

Threats and Vulnerabilities

- **Unit 1**: Introduction to Information Security
- Lecture 4
- **Course**: DI03016035

Agenda

- Defining the Terms
- What is a Threat?
- Categories of Threats
- What is a Vulnerability?
- Types of Vulnerabilities
- The Concept of Risk
- Threat Agents/Actors

Defining the Terms

- **Threat**: A potential cause of an incident that may result in harm.
- **Vulnerability**: A weakness in an asset or control that could be exploited.
- **Attack**: An intentional attempt to bypass security controls (Threat + Vulnerability).
- **Risk**: The likelihood and impact of a threat exploiting a vulnerability.

What is a Threat?

- Anything that can disrupt the operation, function, or integrity of a system.
- Can be intentional (hackers) or accidental (human error).
- Can be natural (earthquakes) or human-made (malware).
- Threats are usually outside of our control.

Categories of Threats

- **Environmental/Natural**: Floods, fires, earthquakes, power outages.
- **Human Accidental**: Employee deleting files, spilling coffee on a server.
- **Human Intentional**: Hackers, malicious insiders, cyber terrorists.
- **Technological**: Hardware failures, software bugs.

What is a Vulnerability?

- A flaw or weakness in system design, implementation, or operation.
- Vulnerabilities are internal to the organization.
- Unlike threats, vulnerabilities CAN be fixed or mitigated.
- **Example**: Unpatched software, weak passwords, unlocked doors.

Types of Vulnerabilities

- **Software Vulnerabilities**: Buffer overflows, injection flaws (bugs in code).
- **Hardware Vulnerabilities**: Design flaws in CPUs (e.g., Meltdown/Spectre).
- **Network Vulnerabilities**: Open ports, unencrypted protocols.
- **Human Vulnerabilities**: Lack of awareness, susceptibility to phishing.

The Equation of Risk

- Risk = Threat x Vulnerability x Impact.
- If there is a threat but no vulnerability, Risk is low.
- If there is a vulnerability but no threat, Risk is low.
- High Risk occurs when a likely threat can exploit a severe vulnerability causing high impact.

Threat Agents / Actors

- The individuals or entities carrying out a threat.
- **Script Kiddies**: Unskilled attackers using pre-made tools.
- **Hacktivists**: Attacking for political or social causes.
- **Cybercriminals**: Attacking for financial gain.
- **Nation-State Actors**: Highly skilled, government-funded attackers (APTs).

Examples: Connecting the Dots

- **Threat**: Rainstorm. Vulnerability: Hole in the roof. Impact: Water damage.
- **Threat**: Hacker. Vulnerability: Weak password. Impact: Data breach.
- **Threat**: Power outage. Vulnerability: No backup generator. Impact: System downtime.

Summary

- Threats are potential harms (outside our control).
- Vulnerabilities are weaknesses (within our control).
- Attacks are threats exploiting vulnerabilities.
- Risk management involves minimizing vulnerabilities to lower overall risk.

Common Types of Attacks

- **Unit 1**: Introduction to Information Security
- Lecture 5
- **Course**: DI03016035

Agenda

- What is an Attack?
- Malware Overview
- Viruses vs. Worms vs. Trojans
- Ransomware
- Social Engineering
- Denial of Service (DoS)
- Man-in-the-Middle (MitM)

What is an Attack?

- The realization of a threat.
- An intentional act attempting to violate the security policy of a system.
- Attacks compromise Confidentiality, Integrity, or Availability.
- Can target networks, hosts, applications, or humans.

Malware (Malicious Software)

- Software specifically designed to disrupt, damage, or gain unauthorized access.
- An umbrella term encompassing many types of malicious code.
- Often delivered via email attachments, malicious downloads, or infected USBs.

Viruses, Worms, and Trojans

- **Virus**: Attaches to a legitimate file and requires human interaction to spread.
- **Worm**: Self-replicating malware that spreads across networks automatically.
- **Trojan Horse**: Disguises itself as legitimate software but contains malicious code; creates backdoors.

Ransomware

- A type of malware that encrypts a victim's files.
- The attacker demands a ransom (usually cryptocurrency) for the decryption key.
- **Double Extortion**: Attackers also steal data and threaten to release it if not paid.
- Highly disruptive to businesses and hospitals.

Social Engineering

- Manipulating people into giving up confidential information.
- Exploits human psychology rather than technical vulnerabilities.
- **Phishing**: Fraudulent emails claiming to be from a reputable source.
- **Spear Phishing**: Targeted phishing at specific individuals.

Denial of Service (DoS & DDoS)

- **DoS**: Overwhelming a system with traffic to make it unavailable to legitimate users.
- Targets the Availability pillar of the CIA triad.
- **DDoS (Distributed DoS)**: Using thousands of compromised computers (a botnet) to attack a single target.
- Extremely difficult to defend against without specialized infrastructure.

Man-in-the-Middle (MitM) Attack

- Attacker secretly intercepts and relays communication between two parties.
- The parties believe they are directly communicating with each other.
- Used to steal login credentials, financial data, or alter messages.
- Prevented using strong encryption (like HTTPS).

Password Attacks

- **Brute Force**: Trying every possible combination of characters.
- **Dictionary Attack**: Trying a list of common words and passwords.
- **Credential Stuffing**: Using passwords stolen from one site on another site.
- **Mitigation**: Strong passwords, Multi-Factor Authentication (MFA).

Summary

- Attacks can be highly technical (Malware, MitM) or psychological (Social Engineering).
- Ransomware and DDoS are major threats to modern businesses.
- Multi-Factor Authentication and encryption are key defenses.

Security Services and Mechanisms

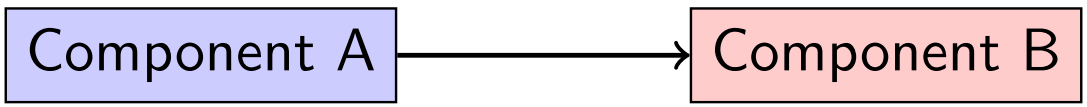
- **Unit 1**: Introduction to Information Security
- Lecture 6
- **Course**: DI03016035

Agenda

- ITU-T X.800 Security Architecture
- What is a Security Service?
- The 5 Key Security Services
- What is a Security Mechanism?
- Specific vs. Pervasive Mechanisms
- Relating Services to Mechanisms

The X.800 Security Architecture

- A systematic approach to defining requirements for security.
- Developed by ITU-T for the OSI network model.
- Divides security into three components: Security Attacks, Services, and Mechanisms.
- Provides a standard framework for implementing security in networks.



What is a Security Service?

- A processing or communication service that enhances the security of systems.
- It is the 'WHAT' needs to be done.
- Provides a specific type of protection for resources.
- **Example**: We need a service that ensures a message isn't altered.

The 5 Key Security Services (X.800)

- **1. Authentication**: Assurance that the communicating entity is the one it claims to be.
- **2. Access Control**: Prevention of unauthorized use of a resource.
- **3. Data Confidentiality**: Protection of data from unauthorized disclosure.
- **4. Data Integrity**: Assurance that data received is exactly as sent by an authorized entity.
- **5. Non-repudiation**: Protection against denial by one of the entities involved in a communication.

Deep Dive: Non-Repudiation

- Prevents either sender or receiver from denying a transmitted message.
- **Origin Non-repudiation**: Proves the message was sent by the specified party.
- **Destination Non-repudiation**: Proves the message was received by the specified party.
- Crucial for legal and financial transactions (e.g., digital contracts).

What is a Security Mechanism?

- A process (or device) that is designed to detect, prevent, or recover from a security attack.
- It is the 'HOW' a security service is implemented.
- Multiple mechanisms may be needed to provide one service.
- One mechanism may support multiple services.

Specific Security Mechanisms

- Mechanisms implemented in a specific OSI layer.
- **Encipherment**: Using cryptography to hide data.
- **Digital Signatures**: Cryptographic proof of origin and integrity.
- **Access Controls**: Hardware/software verifying permissions.
- **Data Integrity Mechanisms**: Hash functions appended to data.

Pervasive Security Mechanisms

- Mechanisms not specific to any particular OSI layer; applied generally.
- **Trusted Functionality**: Hardware/software proven to be secure.
- **Security Labels**: Marking data with sensitivity levels (e.g., Top Secret).
- **Event Detection**: Intrusion detection systems (IDS).
- **Security Audit Trails**: Logging activity for review.

Relating Services to Mechanisms

- **Service**: Confidentiality -¿ Mechanism: Encipherment (Encryption).
- **Service**: Integrity -¿ Mechanism: Hashing / Digital Signatures.
- **Service**: Authentication -¿ Mechanism: Digital Signatures / Passwords.
- **Service**: Non-repudiation -¿ Mechanism: Digital Signatures.

Summary

- The X.800 architecture standardizes security design.
- Security Services define the goals (Authentication, Confidentiality, etc.).
- Security Mechanisms are the tools used to achieve the goals (Encryption, Hashing, etc.).

Passive vs Active Attacks

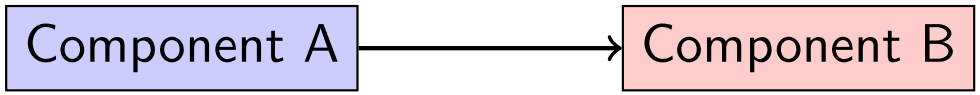
- **Unit 1**: Introduction to Information Security
- Lecture 7
- **Course**: DI03016035

Agenda

- Network Attack Classification
- Passive Attacks Defined
- Types of Passive Attacks
- Challenges of Passive Attacks
- Active Attacks Defined
- Types of Active Attacks
- Comparing Passive and Active
- Defense Strategies

Network Attack Classification

- Network security attacks are broadly categorized into two types based on their nature.
- **Passive Attacks**: Eavesdropping on, or monitoring of, transmissions.
- **Active Attacks**: Modification of the data stream or the creation of a false stream.
- Both pose severe risks, but require different defense strategies.



Passive Attacks Defined

- **Goal**: To obtain information being transmitted.
- The attacker does not alter the data or affect system resources.
- The sender and receiver are unaware they are being monitored.
- Targets the Confidentiality of information.

Types of Passive Attacks

- **1. Release of Message Contents**: Reading email, phone conversations, or unencrypted data packets.
- **2. Traffic Analysis**: Analyzing the frequency, length, and origin/destination of encrypted messages to guess the nature of the communication.
- Traffic analysis works even if the data itself is encrypted!

Challenges of Passive Attacks

- They are extremely difficult to detect because there is no alteration of data.
- No system alerts are triggered.
- Therefore, the focus in dealing with passive attacks is PREVENTION rather than detection.
- **Primary defense**: Strong Encryption.

Active Attacks Defined

- **Goal**: To alter system resources or affect their operation.
- Involves some modification of the data stream or creation of false data.
- Targets Integrity and Availability.
- Often causes noticeable disruptions.

Types of Active Attacks (Part 1)

- **1. Masquerade**: One entity pretends to be a different entity (e.g., Spoofing an IP address to bypass access control).
- **2. Replay**: Passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (e.g., replaying a valid login sequence).

Types of Active Attacks (Part 2)

- **3. Modification of Messages**: A portion of a legitimate message is altered, delayed, or reordered (e.g., changing 'Deposit $10'$ *to*$'$ *Deposit* 1000').
- **4. Denial of Service (DoS)**: Prevents or inhibits the normal use or management of communications facilities.

Challenges of Active Attacks

- It is quite difficult to prevent active attacks entirely due to physical, network, and software vulnerabilities.
- However, they are easier to DETECT than passive attacks.
- Focus is on DETECTION and RECOVERY.
- **Primary defenses**: Hashing, Digital Signatures, Firewalls, IDS.

Summary Comparison

- **Passive**: Goal is eavesdropping. Affects Confidentiality. Hard to detect. Focus on Prevention (Encryption).
- **Active**: Goal is alteration/disruption. Affects Integrity/Availability. Easier to detect. Focus on Detection/Recovery.
- Both require a comprehensive security strategy encompassing the CIA triad.

Unit 1 Summary

- We've covered the history and need for security.
- We defined the core CIA Triad.
- We explored threats, vulnerabilities, attacks, and risk.
- We looked at the X.800 architecture and compared Passive vs. Active attacks.
- You are now ready to learn the Cryptographic tools that provide these security services.

Introduction to Cryptography and Basic Terminology

- **Unit 2**: Cryptography Basics
- Lecture 8
- **Course**: DI03016035

Definition

This is a key concept in Information Security.

Agenda

- What is Cryptography?
- The Evolution of Cryptography
- **Basic Terminology**: Plaintext & Ciphertext
- **Basic Terminology**: Encryption & Decryption
- Ciphers and Keys
- Kerckhoffs's Principle
- Symmetric vs. Asymmetric Cryptography
- Cryptography vs. Steganography
- Summary
- Next Lecture Preview

What is Cryptography?

- Derived from Greek words 'kryptos' (hidden) and 'graphein' (to write).
- The practice and study of techniques for secure communication in the presence of adversarial behavior.
- Transforms readable data into an unreadable format and vice-versa.
- Aims to ensure confidentiality, integrity, authenticity, and non-repudiation.

The Evolution of Cryptography

- **Classical Cryptography**: Relied on pen-and-paper techniques and simple mechanical aids.
- **Medieval & Renaissance**: Introduction of polyalphabetic ciphers.
- **World War I & II**: Rise of complex electro-mechanical rotor machines (e.g., Enigma).
- **Modern Era**: Based on computational hardness, mathematics, and digital computers.

Basic Terminology: Plaintext & Ciphertext

- **Plaintext (or Cleartext)**: The original, intelligible message or data that needs to be protected.
- **Ciphertext**: The scrambled, unreadable output produced by the encryption process.
- The goal of an adversary is usually to recover the plaintext from the ciphertext without authorization.
- **Example**: 'HELLO' (Plaintext) -¿ 'KHOOR' (Ciphertext).

Basic Terminology: Encryption & Decryption

- **Encryption (Enciphering)**: The process of converting plaintext into ciphertext using an algorithm and a key.
- **Decryption (Deciphering)**: The reverse process of converting ciphertext back into plaintext.
- **Formally**: $C = E(K, P)$ and $P = D(K, C)$.
- Both processes typically require a specific secret parameter (the key).

Ciphers and Keys

- **Cipher**: A cryptographic algorithm used for encryption and decryption.
- **Key**: A piece of information (usually a string of bits) that determines the functional output of a cryptographic algorithm.
- The security of the ciphertext should rely entirely on the secrecy of the key, not the secrecy of the algorithm.
- Keys must be randomly generated and securely distributed.

Kerckhoffs's Principle

- Formulated by Auguste Kerckhoffs in the 19th century.
- **Principle**: A cryptographic system should be secure even if everything about the system, except the key, is public knowledge.
- Security through obscurity (hiding the algorithm) is strongly discouraged.
- Allows algorithms to be peer-reviewed and standardized (e.g., AES).

Symmetric vs. Asymmetric Cryptography

- **Symmetric Cryptography**: The same key is used for both encryption and decryption. Fast, but requires secure key distribution.
- **Asymmetric Cryptography**: Uses a pair of keys (Public Key to encrypt, Private Key to decrypt). Slower, but solves the key distribution problem.
- **Hybrid Systems**: Modern systems often use asymmetric crypto to exchange a symmetric key, then use symmetric crypto for the actual data payload.

Cryptography vs. Steganography

- **Cryptography**: Scrambles the message. The existence of the message is known, but its meaning is hidden.
- **Steganography**: Hides the very existence of the message (e.g., embedding text within an image file).
- **Often used together**: A message is encrypted first, then embedded via steganography for a defense-in-depth approach.
- Steganography alone is easily broken if the hiding method is discovered.

Summary

- **Recap of today's core concepts**:
- Cryptography secures communication in the presence of adversaries.
- Modern cryptography is highly mathematical compared to historical methods.
- Plaintext is readable data; Ciphertext is encrypted data.
- Encryption hides the message; decryption reveals it using a key.
- ...and more.

Q&A and Next Steps

- Any Questions?
- Please review the notes and textbook reading for this module.
- Next lecture will build upon these foundational concepts.

Cryptanalysis and Classical Ciphers

- **Unit 2**: Cryptography Basics
- Lecture 9
- **Course**: DI03016035

Agenda

- What is Cryptanalysis?
- Types of Cryptanalytic Attacks (Part 1)
- Types of Cryptanalytic Attacks (Part 2)
- Unconditional vs. Computational Security
- Classical Substitution Ciphers
- Monoalphabetic vs. Polyalphabetic
- Classical Transposition Ciphers
- Rotor Machines and the Enigma
- Summary
- Next Lecture Preview

What is Cryptanalysis?

- The study of mathematical techniques for attempting to defeat cryptographic techniques.
- Also known as codebreaking. The goal is to recover the plaintext or the secret key.
- Evaluates the strength of cryptographic algorithms against various attacks.
- A cipher is 'broken' if a cryptanalyst finds a method faster than a brute-force search of the key space.

Types of Cryptanalytic Attacks (Part 1)

- **Ciphertext-Only Attack (COA)**: The attacker only has access to a set of ciphertexts. The hardest type of attack.
- **Known-Plaintext Attack (KPA)**: The attacker has access to one or more plaintext-ciphertext pairs.
- **Goal of KPA**: To deduce the key or an algorithm to decrypt any future ciphertexts.
- Historically common (e.g., guessing standard message headers).

Types of Cryptanalytic Attacks (Part 2)

- **Chosen-Plaintext Attack (CPA)**: The attacker can choose arbitrary plaintexts and obtain their corresponding ciphertexts.
- **Chosen-Ciphertext Attack (CCA)**: The attacker can choose ciphertexts and obtain their corresponding decrypted plaintexts.
- Modern ciphers must be secure against CPA and CCA.
- **Adaptive attacks**: The attacker chooses subsequent inputs based on previous results.

Unconditional vs. Computational Security

- **Unconditional Security**: The cipher cannot be broken even with infinite computational resources (e.g., One-Time Pad).
- **Computational Security**: The cipher cannot be broken with available computing resources within a reasonable timeframe.
- **Criteria for Computational Security**: Cost of breaking exceeds value of data, or time to break exceeds lifespan of data.
- Most modern cryptography provides computational security, not unconditional.

Classical Substitution Ciphers

- **Substitution Cipher**: Each element in the plaintext (bit, letter, word) is mapped into another element.
- **Caesar Cipher**: A simple substitution cipher where each letter is shifted by a fixed number of positions (e.g., shift of 3).
- **Vulnerable to Brute Force**: Only 25 possible keys in English.
- **Vulnerable to Frequency Analysis**: The frequency of letters in the ciphertext matches the natural language frequency.

Monoalphabetic vs. Polyalphabetic

- **Monoalphabetic**: Uses a single fixed substitution rule for the entire message. Extremely vulnerable to frequency analysis.
- **Polyalphabetic**: Uses multiple substitution alphabets. The rule changes continuously depending on a key.
- **Vigenère Cipher**: A famous polyalphabetic cipher using a keyword to determine the shift for each letter.
- Flattens the frequency distribution, making simple frequency analysis much harder.

Classical Transposition Ciphers

- **Transposition (Permutation) Cipher**: The elements of the plaintext are rearranged (shuffled) rather than substituted.
- **Rail Fence Cipher**: Write plaintext diagonally, read row by row.
- **Row Transposition**: Write plaintext in a rectangle, rearrange columns according to a key, read column by column.
- Letter frequencies remain exactly the same as the original plaintext, just in a different order.

Rotor Machines and the Enigma

- Electro-mechanical devices used primarily in the mid-20th century.
- Consisted of multiple rotating cylinders (rotors) wired to substitute letters.
- As a letter is typed, the rotors advance, continuously changing the substitution alphabet (complex polyalphabetic).
- The German Enigma machine is the most famous example, broken by Polish and British cryptanalysts (Alan Turing).

Summary

- **Recap of today's core concepts**:
- Cryptanalysis is the science of breaking codes.
- Attack models are defined by what information the attacker possesses.
- CPA and CCA give the attacker more power to find vulnerabilities.
- Practical cryptography relies on computational security.
- ...and more.

Q&A and Next Steps

- Any Questions?
- Please review the notes and textbook reading for this module.
- Next lecture will build upon these foundational concepts.

Symmetric Key Cryptography Concepts

- **Unit 2**: Cryptography Basics
- Lecture 10
- **Course**: DI03016035

Agenda

- Principles of Symmetric Key Cryptography
- Stream Ciphers vs. Block Ciphers
- Stream Ciphers in Detail
- **Block Ciphers**: Block Size and Padding
- The Feistel Cipher Structure
- **Feistel Cipher**: Reversibility
- Confusion and Diffusion
- Introduction to Modes of Operation
- Summary
- Next Lecture Preview

Principles of Symmetric Key Cryptography

- Also known as secret-key or single-key cryptography.
- Parties share a single secret key used for both encryption and decryption.
- Requires a secure channel for initial key distribution.
- Highly efficient and suitable for encrypting large volumes of data.
- **Examples**: DES, 3DES, AES, RC4.

Stream Ciphers vs. Block Ciphers

- **Symmetric ciphers are divided into two main categories**: Stream and Block.
- **Stream Ciphers**: Encrypt a continuous stream of data one bit or byte at a time.
- **Block Ciphers**: Encrypt data in fixed-size chunks (blocks) at a time.
- Stream ciphers are faster and have lower hardware complexity, while block ciphers are more common for general data encryption.

Stream Ciphers in Detail

- Generates a continuous pseudo-random keystream based on the secret key.
- Encryption is performed by XORing the plaintext bits with the keystream bits.
- Decryption uses the exact same keystream XORed with the ciphertext.
- **Crucial requirement**: Keystream must never be reused (avoid two-time pad).
- **Notable example**: RC4 (now considered insecure).

Block Ciphers: Block Size and Padding

- Block ciphers process plaintext in blocks (e.g., 64 bits or 128 bits).
- If the plaintext length is not a multiple of the block size, padding must be added.
- **Padding methods**: PKCS#7 (adds bytes with the value of the number of padding bytes), Zero padding.
- A fixed key is used to transform the entire block via multiple rounds of operations.

The Feistel Cipher Structure

- Proposed by Horst Feistel at IBM; forms the basis of many block ciphers (e.g., DES).
- Inputs: A plaintext block and a key.
- The plaintext block is divided into two halves: Left (L) and Right (R).
- Operates in multiple identical rounds.
- In each round, the right half is transformed by a function 'F' using a subkey, and XORed with the left half.

Feistel Cipher: Reversibility

- A key feature of the Feistel structure is that the 'F' function does not need to be invertible.
- Decryption is exactly the same process as encryption, just applying the round subkeys in reverse order.
- This drastically reduces hardware and software implementation costs (one algorithm for both).
- Security depends on block size, key size, number of rounds, and the design of the 'F' function.

Confusion and Diffusion

- Introduced by Claude Shannon to frustrate statistical cryptanalysis.
- **Confusion**: Makes the relationship between the ciphertext and the symmetric key as complex as possible (usually achieved by substitution).
- **Diffusion**: Dissipates the statistical structure of plaintext into long-range statistics of ciphertext (usually achieved by permutation).
- A small change in plaintext should cause a massive change in ciphertext (Avalanche Effect).

Introduction to Modes of Operation

- A block cipher algorithm only encrypts a single block. How do we encrypt a large file?
- **Electronic Codebook (ECB)**: Encrypt each block independently. Identical plaintext blocks produce identical ciphertext blocks (insecure for patterns).
- **Cipher Block Chaining (CBC)**: XOR each plaintext block with the previous ciphertext block before encrypting.
- CBC requires an Initialization Vector (IV) for the first block.

Definition

This is a key concept in Information Security.

Summary

- **Recap of today's core concepts**:
- Symmetric cryptography uses one shared secret key.
- Ciphers operate on data either as a continuous stream or in fixed blocks.
- Stream ciphers XOR plaintext with a pseudo-random keystream.
- Block ciphers require data to be padded to fit fixed block sizes.
- ...and more.

Q&A and Next Steps

- Any Questions?
- Please review the notes and textbook reading for this module.
- Next lecture will build upon these foundational concepts.

Data Encryption Standard (DES)

- **Unit 2**: Cryptography Basics
- Lecture 11
- **Course**: DI03016035

Agenda

- Introduction to DES and History
- DES Overall Architecture
- Initial and Final Permutations
- The DES Round Structure
- The DES Key Schedule
- The Avalanche Effect in DES
- Vulnerabilities and Cryptanalysis of DES
- Double DES and Meet-in-the-Middle
- Summary
- Next Lecture Preview

Introduction to DES and History

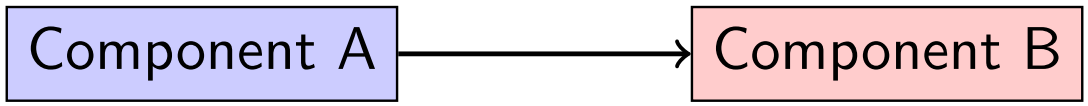
- Developed by IBM in the 1970s based on the Lucifer cipher.
- Adopted as a US federal standard (FIPS) in 1977.
- The first cryptographic algorithm openly published by the US government.
- Spurred the modern academic study of cryptography.
- Currently considered insecure due to its small key size, but historically vital.

Definition

This is a key concept in Information Security.

DES Overall Architecture

- Type: Symmetric Block Cipher.
- Structure: 16-round Feistel Network.
- Block Size: 64 bits.
- Key Size: 64 bits total, but only 56 bits are used for actual security (8 bits are for parity).
- Consists of Initial Permutation (IP), 16 rounds of complex functions, and Final Permutation (FP).



Initial and Final Permutations

- Initial Permutation (IP) rearranges the bits of the 64-bit input block.
- Final Permutation (FP, or IP-inverse) reverses the rearrangement at the end.
- These permutations do not add any cryptographic security (they are fixed and linear).
- Originally designed to make it easier to load data into 8-bit byte-oriented hardware of the 1970s.

The DES Round Structure

- Follows the classic Feistel structure: L(i) = R(i-1) and R(i) = L(i-1) XOR F(R(i-1), K(i)).
- The 'F' Function handles the core cryptographic operations.
- F Function steps: 1) Expansion Permutation (32 bit to 48 bit). 2) XOR with 48-bit Round Key. 3) Substitution Boxes (S-Boxes). 4) Permutation (P-Box).
- S-Boxes provide the crucial non-linear 'confusion'.

The DES Key Schedule

- Generates sixteen 48-bit subkeys from the original 56-bit master key.
- Process involves Permuted Choice 1 (PC-1) to strip parity bits.
- The 56 bits are split into two 28-bit halves.
- In each round, the halves are circularly shifted left by 1 or 2 bits.
- Permuted Choice 2 (PC-2) selects 48 bits from the shifted 56 bits to form the round subkey.

The Avalanche Effect in DES

- DES exhibits a strong avalanche effect.
- A small change in either the plaintext or the key causes significant changes in the ciphertext.
- Changing just 1 bit of the plaintext will alter approximately half (32 bits) of the ciphertext bits after 16 rounds.
- This ensures that statistical relationships are completely hidden.

Vulnerabilities and Cryptanalysis of DES

- **Brute-Force Attack**: With only 56 bits, there are 2^56 (approx 7.2 x 10^16) possible keys. Modern hardware can search this in hours/days.
- EFF's 'Deep Crack' machine broke DES in 56 hours in 1998.
- **Differential Cryptanalysis**: Discovered in the 1990s, but DES's S-Boxes were surprisingly resistant (IBM knew about it in the 70s).
- **Linear Cryptanalysis**: Can break DES faster than brute force (requires 2^43 known plaintexts).

Double DES and Meet-in-the-Middle

- **Idea**: Encrypt twice with two keys (K1, K2) to get a 112-bit key size: C = E(K2, E(K1, P)).
- **Meet-in-the-Middle Attack**: Encrypt P with all possible K1, store in table. Decrypt C with all K2, look for matches in table.
- Reduces security of Double DES from 2^112 to roughly 2^57.
- **Solution**: Triple DES (3DES). Uses 3 keys: C = E(K3, D(K2, E(K1, P))). Provides 112 or 168 bits of security.

Summary

- **Recap of today's core concepts**:
- DES is a historically significant, Feistel-based block cipher.
- DES uses a 64-bit block size and a 56-bit effective key.
- IP and FP are structural permutations without cryptographic value.
- The F function uses Expansion, XOR, S-Boxes, and Permutation.
- ...and more.

Q&A and Next Steps

- Any Questions?
- Please review the notes and textbook reading for this module.
- Next lecture will build upon these foundational concepts.

Advanced Encryption Standard (AES)

- **Unit 2**: Cryptography Basics
- Lecture 12
- **Course**: DI03016035

Agenda

- Introduction to AES and History
- AES Block and Key Sizes
- AES Overall Structure (State Matrix)
- **AES Transformation 1**: SubBytes
- **AES Transformation 2**: ShiftRows
- **AES Transformation 3**: MixColumns
- **AES Transformation 4**: AddRoundKey
- AES Security and Performance
- Summary
- Next Lecture Preview

Introduction to AES and History

- NIST issued a call for a DES replacement in 1997 due to its small key size.
- **Required**: Symmetric block cipher, minimum 128-bit block size, 128/192/256-bit key sizes, efficient in hardware and software.
- **Winner**: The Rijndael algorithm, designed by Belgian cryptographers Joan Daemen and Vincent Rijmen.
- Standardized as AES (FIPS 197) in 2001.
- Widely used globally today (Wi-Fi, HTTPS, VPNs).

Definition

This is a key concept in Information Security.

AES Block and Key Sizes

- **Fixed Block Size**: 128 bits (16 bytes).
- **Variable Key Sizes**: 128, 192, or 256 bits.
- **Number of Rounds depends on the key size**: 10 rounds (128-bit key), 12 rounds (192-bit), 14 rounds (256-bit).
- Unlike DES, AES is NOT a Feistel cipher. It is a Substitution-Permutation Network (SPN).

AES Overall Structure (State Matrix)

- AES processes the 128-bit data block as a 4x4 matrix of bytes, called the 'State'.
- All operations in AES are performed on this State matrix.
- Algorithm flow: Initial AddRoundKey, followed by (Nr - 1) standard rounds, followed by a final round (without MixColumns).
- Each standard round consists of 4 distinct transformation steps.

AES Transformation 1: SubBytes

- A non-linear byte substitution step.
- Each byte in the State matrix is replaced independently with another byte.
- Uses a single, fixed 16x16 lookup table called the S-Box.
- Provides 'Confusion' by introducing non-linearity.
- Designed to resist differential and linear cryptanalysis.

AES Transformation 2: ShiftRows

- A linear permutation step that operates on the rows of the State matrix.
- **Row 0**: Not shifted.
- **Row 1**: Circularly shifted left by 1 byte.
- **Row 2**: Circularly shifted left by 2 bytes.
- **Row 3**: Circularly shifted left by 3 bytes.
- Provides 'Diffusion' by spreading data across columns.

AES Transformation 3: MixColumns

- A linear substitution step operating on columns of the State.
- Each column is treated as a 4-term polynomial and multiplied by a fixed matrix over Galois Field (2^8).
- Heavily mixes the bytes within each column.
- Combined with ShiftRows, it guarantees that after a few rounds, every bit of ciphertext depends on every bit of plaintext (massive diffusion).

AES Transformation 4: AddRoundKey

- The actual encryption step where the key is introduced.
- A simple bitwise XOR operation between the State matrix and the Round Subkey.
- The round subkey is derived from the main key via the AES Key Schedule algorithm.
- Reversible simply by XORing the same round key again.

AES Security and Performance

- **Security**: Currently, there are no known practical attacks that are significantly faster than brute force.
- Brute force against a 256-bit key is computationally infeasible (2^256 operations).
- **Performance**: Designed to be highly efficient in software (uses byte-level operations) and hardware.
- Modern CPUs have hardware instructions (AES-NI) that compute AES extremely fast.
- **Side-Channel Attacks**: Implementations must be careful regarding timing and power analysis attacks.

Summary

- **Recap of today's core concepts**:
- AES (Rijndael) replaced DES as the global encryption standard.
- AES uses a 128-bit block and 128/192/256-bit keys with varying rounds.
- AES processes data in a 4x4 byte State matrix.
- SubBytes provides confusion via a highly non-linear S-Box.
- ...and more.

Q&A and Next Steps

- Any Questions?
- Please review the notes and textbook reading for this module.
- Next lecture will build upon these foundational concepts.

Asymmetric Key Cryptography Concepts

- **Unit 2**: Cryptography Basics
- Lecture 13
- **Course**: DI03016035

Agenda

- Limitations of Symmetric Cryptography
- Principles of Public-Key Cryptography
- Public and Private Keys
- **Mathematical Foundations**: Prime Numbers
- **Mathematical Foundations**: Modular Arithmetic
- Trapdoor One-Way Functions
- Applications of Asymmetric Cryptography
- Symmetric vs. Asymmetric Comparison
- Summary
- Next Lecture Preview

Limitations of Symmetric Cryptography

- **The Key Distribution Problem**: How do Alice and Bob securely share a secret key over an insecure network before communicating?
- **Scalability**: In a network of N users, N(N-1)/2 keys are needed for everyone to communicate securely.
- **Trust**: Requires absolute trust between parties; cannot provide true non-repudiation.
- These problems catalyzed the invention of Public-Key Cryptography in the 1970s.

Principles of Public-Key Cryptography

- Also known as Asymmetric Cryptography.
- Introduced by Whitfield Diffie and Martin Hellman in 1976.
- **Uses two mathematically related keys**: A Public Key and a Private Key.
- It is computationally infeasible to determine the private key from the public key.
- **Solves the key distribution problem**: Public keys can be shared openly.

Public and Private Keys

- **Public Key (PU)**: Known to everyone. Used to encrypt messages intended for the owner.
- **Private Key (PR)**: Known ONLY to the owner. Used to decrypt messages encrypted with the public key.
- **Example**: Alice wants to send a secret to Bob. Alice encrypts it using Bob's Public Key. Only Bob can decrypt it using his Private Key.
- Key pair generation is linked mathematically.

Mathematical Foundations: Prime Numbers

- Asymmetric algorithms rely heavily on Number Theory.
- **Prime Numbers**: Integers greater than 1 that have only two divisors: 1 and themselves.
- **Fundamental Theorem of Arithmetic**: Every integer can be factored into a unique product of primes.
- Factoring large numbers (finding their prime components) is computationally very hard (the basis of RSA).

Mathematical Foundations: Modular Arithmetic

- **Clock arithmetic**: Numbers 'wrap around' upon reaching a certain value (the modulus).
- **Notation**: $A \equiv B \pmod{N}$ means A and B have the same remainder when divided by N.
- Operations like addition, multiplication, and exponentiation are easily performed modulo N.
- Modular Exponentiation is easy to compute, but finding the discrete logarithm is computationally hard.

Trapdoor One-Way Functions

- **One-Way Function**: Easy to compute in one direction, but practically impossible to reverse.
- **Example**: Multiplying two large primes is easy; factoring the result is hard.
- **Trapdoor**: A secret piece of information (the private key) that suddenly makes the reverse computation easy.
- Asymmetric cryptography is entirely based on trapdoor one-way functions.

Applications of Asymmetric Cryptography

- **1. Encryption / Confidentiality**: Sender encrypts with receiver's public key.
- **2. Digital Signatures / Authentication**: Sender encrypts (signs) a hash with their own private key. Anyone can verify using sender's public key.
- **3. Key Exchange**: Safely negotiating a shared symmetric key over a public channel (e.g., Diffie-Hellman).
- Cannot be used for fast, bulk data encryption due to computational overhead.

Symmetric vs. Asymmetric Comparison

- **Keys**: Symmetric uses 1 shared key; Asymmetric uses 2 related keys.
- **Speed**: Symmetric is 100x-1000x faster; Asymmetric is computationally heavy.
- **Key Distribution**: Symmetric requires a secure channel; Asymmetric allows public distribution.
- **Usage**: Symmetric for bulk payload encryption (AES); Asymmetric for key exchange and signatures (RSA).
- **Modern systems (TLS/HTTPS) use hybrid approaches**: Asymmetric to exchange a Symmetric key.

Summary

- **Recap of today's core concepts**:
- Symmetric crypto struggles with key distribution and scalability.
- Asymmetric crypto uses a public key to encrypt and a private key to decrypt.
- Anyone can encrypt using the public key; only the owner can decrypt.
- Prime factorization is a mathematically 'hard' problem.
- ...and more.

Q&A and Next Steps

- Any Questions?
- Please review the notes and textbook reading for this module.
- Next lecture will build upon these foundational concepts.

The RSA Algorithm

- **Unit 2**: Cryptography Basics
- Lecture 14
- **Course**: DI03016035

Agenda

- Introduction to RSA
- RSA Key Generation (Step 1-3)
- RSA Key Generation (Step 4-5)
- RSA Encryption Process
- RSA Decryption Process
- **RSA Example**: Key Generation
- **RSA Example**: Encryption & Decryption
- Security of RSA
- Summary
- Next Lecture Preview

Introduction to RSA

- Developed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977 at MIT.
- The most widely used public-key cryptosystem.
- Based on the mathematical difficulty of factoring the product of two large prime numbers.
- Supports both public-key encryption and digital signatures.
- Key sizes typically range from 2048 to 4096 bits for adequate security today.

Definition

This is a key concept in Information Security.

RSA Key Generation (Step 1-3)

- **Step 1**: Choose two distinct, large prime numbers, p and q.
- **Step 2**: Calculate their product, n = p * q. The value 'n' is the modulus for both keys.
- **Step 3**: Calculate Euler's totient function, ϕ(n) = (p-1) * (q-1).
- p and q must be kept secret and are typically destroyed after key generation.

RSA Key Generation (Step 4-5)

- **Step 4**: Choose an integer 'e' such that 1 ¡ e ¡ ϕ(n), and 'e' is coprime to ϕ(n).
- 'e' becomes the Public Exponent. (Commonly 65537).
- **Step 5**: Compute 'd' as the modular multiplicative inverse of 'e' modulo ϕ(n).
- This means (d * e) $\equiv$ 1 mod ϕ(n). 'd' becomes the Private Exponent.
- **Public Key**: (e, n). Private Key: (d, n).

RSA Encryption Process

- Sender needs to encrypt a message 'M' for the Receiver.
- Sender obtains the Receiver's Public Key (e, n).
- Message 'M' must be converted to an integer such that 0 ¡= M ¡ n.
- **Ciphertext is computed using modular exponentiation**: C = M^e mod n.
- The ciphertext 'C' is sent to the Receiver.

RSA Decryption Process

- Receiver receives ciphertext 'C'.
- Receiver uses their own Private Key (d, n) to decrypt.
- **Plaintext is recovered by**: M = C^d mod n.
- Due to Euler's theorem and the way 'd' and 'e' were chosen, this mathematically reverses the encryption.
- Only the person holding 'd' can perform this efficiently.

RSA Example: Key Generation

- Let p = 61 and q = 53.
- n = 61 * 53 = 3233.
- ϕ(n) = (61-1)*(53-1) = 3120.
- Choose e = 17 (since 17 and 3120 are coprime).
- **Compute d such that (d * 17) $\equiv$ 1 mod 3120. Result**: d = 2753.
- **Public key**: (17, 3233). Private key: (2753, 3233).

RSA Example: Encryption & Decryption

- Public Key = (17, 3233), Private Key = (2753, 3233).
- Let plaintext message M = 65.
- **Encryption**: C = 65^17 mod 3233.
- C = 2790. (Ciphertext to be sent).
- **Decryption**: M = 2790^2753 mod 3233.
- M = 65. (Original plaintext recovered).

Security of RSA

- Security entirely relies on the Integer Factorization Problem.
- If an attacker can factor the public modulus 'n' back into 'p' and 'q', they can compute ϕ(n) and easily find 'd'.
- Currently, factoring a 2048-bit number is computationally infeasible with classical computers.
- **Quantum Threat**: Shor's algorithm on a sufficiently powerful quantum computer could break RSA.
- Padding Schemes (e.g., OAEP) are essential to prevent chosen-ciphertext attacks.

Summary

- **Recap of today's core concepts**:
- RSA is a pioneering asymmetric algorithm based on prime factorization.
- Key generation starts with two large primes, p and q.
- Public key is (e, n); Private key is (d, n).
- **Encryption uses the public exponent**: C = M^e mod n.
- ...and more.

Q&A and Next Steps

- Any Questions?
- Please review the notes and textbook reading for this module.
- Next lecture will build upon these foundational concepts.

Cryptographic Hash Functions

- **Unit 2**: Cryptography Basics
- Lecture 15
- **Course**: DI03016035

Agenda

- What is a Cryptographic Hash Function?
- **Property 1**: Pre-image Resistance
- **Property 2**: Second Pre-image Resistance
- **Property 3**: Collision Resistance
- The Birthday Attack
- MD5 (Message-Digest Algorithm 5)
- The SHA (Secure Hash Algorithm) Family
- Applications of Hash Functions
- Summary
- Next Lecture Preview

What is a Cryptographic Hash Function?

- A mathematical algorithm that maps data of arbitrary size to a bit string of fixed size.
- The output is called a hash value, hash code, digest, or simply hash.
- **Deterministic**: The same message always results in the same hash.
- Fast to compute for any given message.
- Keyless operation (unlike encryption algorithms).

Property 1: Pre-image Resistance

- Also known as One-Wayness.
- Given a hash value 'h', it must be computationally infeasible to find any original message 'm' such that Hash(m) = h.
- Ensures that a hash cannot be reversed to reveal the original data.
- **Crucial for password storage**: even if the hash is stolen, the password remains unknown.

Property 2: Second Pre-image Resistance

- Also known as Weak Collision Resistance.
- Given a specific message 'm1', it is computationally infeasible to find a different message 'm2' such that Hash(m1) = Hash(m2).
- Prevents an attacker from forging a new document that has the same hash as a specific legitimate document.
- Protects the integrity of specific files.

Property 3: Collision Resistance

- Also known as Strong Collision Resistance.
- It is computationally infeasible to find ANY two different messages 'm1' and 'm2' such that Hash(m1) = Hash(m2).
- Since the input space is infinite and the output space is fixed, collisions mathematically MUST exist (Pigeonhole Principle).
- A good hash function just makes finding them computationally impossible.

The Birthday Attack

- A cryptographic attack that exploits the mathematics behind the birthday paradox.
- **The paradox**: In a room of just 23 people, there's a 50% chance two share a birthday.
- **Applied to hashes**: Finding a collision (any m1, m2) is much easier than finding a specific second pre-image.
- If a hash output is 'n' bits, a collision can typically be found in 2^(n/2) operations.
- Forces hash functions to have larger output sizes (e.g., 256 bits gives 128-bit security).

MD5 (Message-Digest Algorithm 5)

- Designed by Ron Rivest in 1991.
- Produces a 128-bit hash value.
- Widely used historically for file integrity checking.
- **Vulnerabilities**: Collision resistance was broken in 2004. Attackers can generate two files with the same MD5 hash in seconds.
- **Status**: Cryptographically broken and unsuitable for further use (except non-security checksums).

The SHA (Secure Hash Algorithm) Family

- Published by NIST.
- **SHA-1**: 160-bit output. Collision found in 2017 (SHAttered attack). Now deprecated.
- **SHA-2**: A family including SHA-224, SHA-256, SHA-384, and SHA-512. Currently the industry standard.
- **SHA-3**: Released in 2015 based on the Keccak algorithm. Uses a 'sponge construction' entirely different from SHA-2.
- SHA-2 and SHA-3 are considered secure today.

Applications of Hash Functions

- **File Integrity Verification**: Software downloads provide a hash to ensure the file wasn't corrupted or tampered with.
- **Password Storage**: Systems store the hash of a password, never the plaintext.
- **Digital Signatures**: You hash a document and encrypt the hash, not the whole document.
- **Blockchain and Cryptocurrencies**: Used for proof-of-work and chaining blocks together.

Summary

- **Recap of today's core concepts**:
- Hash functions compress arbitrary data into a fixed-size digest.
- Pre-image resistance means hashes cannot be reversed.
- Second pre-image resistance protects specific messages from substitution.
- Collision resistance means finding ANY two inputs with the same hash is infeasible.
- ...and more.

Q&A and Next Steps

- Any Questions?
- Please review the notes and textbook reading for this module.
- Next lecture will build upon these foundational concepts.

Message Authentication Codes (MAC)

- **Unit 2**: Cryptography Basics
- Lecture 16
- **Course**: DI03016035

Agenda

- Need for Message Authentication
- What is a MAC?
- MAC vs. Hash Functions
- MAC Generation and Verification Process
- Security Requirements of MACs
- HMAC (Hash-based MAC)
- HMAC Structure and Operation
- CMAC (Cipher-based MAC)
- Summary
- Next Lecture Preview

Need for Message Authentication

- Encryption provides Confidentiality (prevents eavesdropping).
- However, encryption alone does not provide Data Origin Authentication or Data Integrity.
- An attacker might flip bits in the ciphertext, causing predictable changes in the decrypted plaintext.
- We need a mechanism to prove that a message has not been altered and comes from the claimed sender.

What is a MAC?

- A short piece of information used to authenticate a message.
- Also known as a cryptographic checksum or tag.
- **Takes two inputs**: The message (arbitrary length) and a Secret Key.
- **Produces a fixed-size MAC value**: MAC = F(Key, Message).
- Only someone with the secret key can generate or verify the MAC.

MAC vs. Hash Functions

- **Hash Function**: Keyless. Anyone can compute the hash of a message. Provides integrity against random errors, but not malicious tampering.
- If an attacker changes the message, they can just compute a new hash.
- **MAC**: Keyed. Requires a secret key. Provides integrity against malicious tampering.
- An attacker cannot compute the new MAC for an altered message without the key.

MAC Generation and Verification Process

- **Sender**: 1. Calculates MAC = Algorithm(Key, Message). 2. Appends MAC to the Message. 3. Sends (Message —— MAC).
- **Receiver**: 1. Receives (Message, Received_MAC). 2. Calculates Expected_MAC = Algorithm(Key, Message).
- 3. Compares Expected_MAC with Received_MAC.
- If they match, the message is authentic and unaltered.

Security Requirements of MACs

- **Computation Resistance**: Given multiple (Message, MAC) pairs, it must be infeasible to compute a valid MAC for a new message.
- The MAC algorithm must possess strong cryptographic properties, similar to encryption or hashing.
- **Vulnerable to Replay Attacks**: An attacker can record a valid (Message, MAC) and send it again later.
- **Defense**: Use sequence numbers or timestamps within the authenticated message.

HMAC (Hash-based MAC)

- A specific construction for calculating a MAC involving a cryptographic hash function (like SHA-256) in combination with a secret key.
- Standardized in RFC 2104.
- **Motivation**: Cryptographic hash functions generally execute faster in software than symmetric block ciphers.
- Avoids vulnerabilities associated with simple implementations like Hash(Key —— Message).

HMAC Structure and Operation

- Formula: HMAC(K, M) = H((K' XOR opad) —— H((K' XOR ipad) —— M))
- K' is the key padded to block size. ipad and opad are fixed inner and outer padding constants.
- The message is hashed twice.
- First pass hashes the message with the inner pad.
- Second pass hashes the result of the first pass with the outer pad.
- Highly secure against length extension attacks.

CMAC (Cipher-based MAC)

- A MAC based on a symmetric block cipher (usually AES).
- Standardized by NIST.
- Operates similarly to Cipher Block Chaining (CBC) mode, but only keeps the final encrypted block as the MAC.
- Preferred when a hardware implementation of AES is available, or when hash functions are not desired.
- Fixes security issues found in older CBC-MAC algorithms.

Summary

- **Recap of today's core concepts**:
- Authentication ensures message integrity and origin.
- A MAC uses a shared secret key to authenticate data.
- MACs use keys; Hash functions do not.
- Verification relies on recalculating the MAC and comparing.
- ...and more.

Q&A and Next Steps

- Any Questions?
- Please review the notes and textbook reading for this module.
- Next lecture will build upon these foundational concepts.

Digital Signatures and Applications

- **Unit 2**: Cryptography Basics
- Lecture 17
- **Course**: DI03016035

Agenda

- What is a Digital Signature?
- Properties of Digital Signatures
- Digital vs. Traditional Signatures
- Creating a Digital Signature (Signing)
- Verifying a Digital Signature
- Digital Signature Standard (DSS)
- RSA Digital Signatures vs. DSA
- Public Key Infrastructure (PKI)
- Summary
- Next Lecture Preview

What is a Digital Signature?

- An electronic analog of a written signature.
- Provides assurances of data origin, identity, and status of an electronic document.
- Based on Asymmetric (Public Key) Cryptography.
- Only the signer (possessing the private key) can create the signature.
- Anyone (possessing the public key) can verify the signature.

Properties of Digital Signatures

- **Authentication**: Verifies who created the message.
- **Data Integrity**: Ensures the message has not been altered since it was signed.
- **Non-Repudiation**: The sender cannot falsely deny having sent the message, as only their private key could have created the signature.
- Unlike symmetric MACs, third parties can verify the signature and resolve disputes.

Digital vs. Traditional Signatures

- **Traditional**: Physically bound to the paper. Susceptible to forgery. Looks the same on every document.
- **Digital**: Mathematically bound to the specific data being signed. Computationally infeasible to forge.
- A digital signature changes completely if even one bit of the document changes.
- A digital signature changes for every new document signed by the same person.

Creating a Digital Signature (Signing)

- **Step 1**: The sender calculates a Hash (digest) of the message/document.
- **Step 2**: The sender encrypts the Hash using their own Private Key.
- This encrypted hash is the Digital Signature.
- **Step 3**: The sender transmits the original Message along with the Digital Signature.
- Hashing first is crucial for performance, as public-key encryption of large documents is extremely slow.

Verifying a Digital Signature

- **Step 1**: Receiver receives the Message and the Signature.
- **Step 2**: Receiver calculates a new Hash of the received Message.
- **Step 3**: Receiver decrypts the Signature using the sender's Public Key to recover the original Hash.
- **Step 4**: Receiver compares the new Hash with the decrypted original Hash.
- If they match exactly, the signature is valid.

Digital Signature Standard (DSS)

- A US Federal Standard published by NIST.
- Specifies the algorithms permissible for generating digital signatures.
- Includes RSA (Rivest-Shamir-Adleman).
- Includes DSA (Digital Signature Algorithm), which is based on discrete logarithms.
- Includes ECDSA (Elliptic Curve Digital Signature Algorithm), which provides high security with much smaller key sizes.

RSA Digital Signatures vs. DSA

- **RSA**: Can be used for both encryption and digital signatures. Verification (public key operation) is usually very fast.
- **DSA**: Designed specifically ONLY for digital signatures, not encryption. Generating a signature is faster than RSA, but verification is slower.
- DSA requires a secure random number generator for every signature; if the random number is reused or predictable, the private key can be leaked.
- Sony PlayStation 3 was hacked because of a flawed ECDSA random number implementation.

Public Key Infrastructure (PKI)

- **Problem**: How do I know the Public Key I downloaded really belongs to Alice and not an attacker?
- **Solution**: PKI and Digital Certificates.
- A Certificate Authority (CA) acts as a trusted third party.
- The CA verifies Alice's identity and issues a Certificate containing Alice's Public Key.
- The CA digitally signs this Certificate with its own private key.
- Web browsers have trusted CA public keys pre-installed to verify these certificates (HTTPS).

Summary

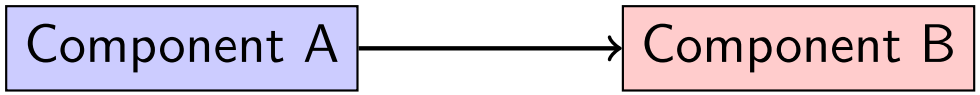
- **Recap of today's core concepts**:
- Digital signatures use private keys to sign, and public keys to verify.
- Digital signatures uniquely provide Non-Repudiation.
- Digital signatures are uniquely tied to the specific message content.
- Signing involves encrypting a message hash with a private key.
- ...and more.

Q&A and Next Steps

- Any Questions?
- Please review the notes and textbook reading for this module.
- Next lecture will build upon these foundational concepts.

Introduction to Network Security

- Network security focuses on protecting data during transmission across interconnected nodes.
- It involves policies, practices, and technologies to monitor and prevent unauthorized access.
- The primary goal is to maintain Confidentiality, Integrity, and Availability (CIA).
- Networks are inherently vulnerable due to open protocols designed for communication, not security.
- Securing a network requires Defense in Depth (layered security approach).

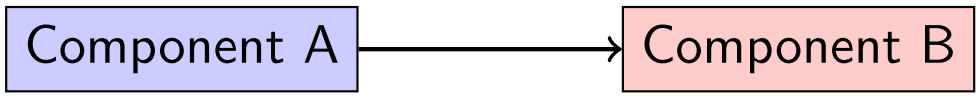


Definition

This is a key concept in Information Security.

Categorizing Network Threats

- Network threats exploit vulnerabilities in protocols, operating systems, or architectures.
- **Structured Threats**: Planned, highly skilled attacks (e.g., APTs).
- **Unstructured Threats**: Inexperienced attackers using readily available tools.
- **Passive Threats**: Eavesdropping, traffic analysis (does not alter data).
- **Active Threats**: Modification of data, masquerading, replay, and DoS.



Introduction to Denial of Service (DoS)

- A DoS attack aims to make a machine or network resource unavailable to intended users.
- It achieves this by overwhelming the target with a flood of illegitimate requests.
- Alternatively, it can exploit a software bug to crash the target system.
- DoS violates the 'Availability' pillar of the CIA triad.
- Often used as a distraction while another attack (like data exfiltration) takes place.

Definition

This is a key concept in Information Security.

Mechanics of a DoS Attack

- **Volumetric Attacks**: Consume all available bandwidth (e.g., UDP floods).
- **Protocol Attacks**: Consume actual server resources or intermediate equipment (firewalls/load balancers).
- **Application Layer Attacks**: Crash the web server by exploiting application-specific vulnerabilities (e.g., HTTP GET floods).
- **Resource Starvation**: Depleting CPU, memory, or disk space on the target.
- **Asymmetric Workload**: The attacker uses minimal resources to force the server to use maximal resources.

Common DoS Techniques: Ping of Death & Teardrop

- **Ping of Death**: Sending an oversized ICMP packet (larger than 65,535 bytes).
- When the target reassembles the fragmented packet, it causes a buffer overflow and system crash.
- **Teardrop Attack**: Sending fragmented packets with overlapping, oversized payloads.
- The target OS cannot handle the overlapping offset values during reassembly.
- Both attacks target older protocol stack implementations and are mostly mitigated today.

SYN Flood Attack: Exploiting TCP

- Exploits the TCP 3-way handshake (SYN, SYN-ACK, ACK).
- Attacker sends a rapid succession of SYN requests with spoofed source IP addresses.
- Server responds with SYN-ACK and allocates memory for the half-open connection.
- The final ACK never arrives, leaving the connection half-open until a timeout occurs.
- Server's connection queue fills up, denying service to legitimate TCP connection requests.

Introduction to Distributed DoS (DDoS)

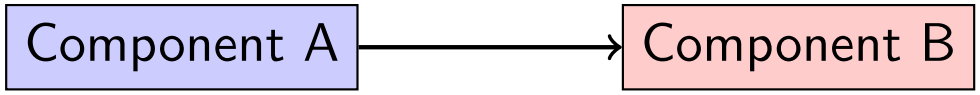
- DDoS involves multiple compromised systems attacking a single target simultaneously.
- Multiplies the attacking power exponentially compared to a single-source DoS.
- Makes mitigation difficult because the attack traffic comes from thousands of legitimate IPs.
- **Utilizes a Botnet**: A network of malware-infected devices controlled by an attacker.
- IoT devices are heavily targeted for botnets due to weak default security.

Definition

This is a key concept in Information Security.

DDoS Architecture: Botnets and C2

- Botmaster: The attacker controlling the botnet.
- Command and Control (C2) Server: Infrastructure used to issue instructions to bots.
- Zombies (Bots): Compromised endpoints (PCs, servers, cameras, routers).
- The botmaster sends a command to the C2, which distributes the attack target and payload instructions to the zombies.
- Zombies execute the coordinated attack, masking the botmaster's identity.



Amplification and Reflection Attacks

- **Reflection**: Attacker spoofs the target's IP address and sends requests to vulnerable internet servers (e.g., DNS, NTP).
- The servers reply to the target, reflecting the traffic.
- **Amplification**: The request is small, but the response is massive (e.g., requesting a large DNS zone file).
- Amplification factor can be 50x to 500x the original request bandwidth.
- Commonly uses UDP protocols like DNS, NTP, SSDP, and Memcached.

Impact of DoS/DDoS on Business

- **Financial Loss**: E-commerce sites lose revenue for every minute of downtime.
- **Reputational Damage**: Customers lose trust in the brand's reliability and security.
- **Productivity Loss**: Internal employees cannot access network resources or external cloud apps.
- **SLA Breaches**: Service providers face financial penalties for failing uptime guarantees.
- **Remediation Costs**: Emergency bandwidth purchasing, incident response consultants, and mitigation services.

Basic Mitigation Strategies for DoS/DDoS

- **Rate Limiting**: Restricting the number of requests a single IP can make.
- **SYN Cookies**: Cryptographic cookies used to prevent SYN floods without allocating memory immediately.
- **Blackholing/Sinkholing**: Routing attack traffic into a null route (drops all traffic, including legitimate).
- **Anycast Networks**: Distributing the attack load across multiple globally distributed servers.
- Dedicated DDoS Mitigation Providers (e.g., Cloudflare, Akamai) that scrub traffic before it reaches the origin.

Introduction to Spoofing and Sniffing

- Spoofing and Sniffing are fundamental network attack techniques.
- Spoofing involves masquerading as a trusted entity to deceive systems or users.
- Sniffing involves capturing and analyzing network traffic as it passes over the medium.
- Sniffing is a passive threat (gathering information), while spoofing is an active threat (falsifying identity).
- Both often act as precursors to more complex attacks, like Man-in-the-Middle (MitM).

Definition

This is a key concept in Information Security.

IP Spoofing Concepts

- IP Spoofing is the creation of IP packets with a forged source IP address.
- The goal is to conceal the identity of the sender or impersonate another computing system.
- Since routers forward packets based on the destination address, they rarely check the validity of the source address.
- Often used in DDoS amplification attacks (as discussed in the previous lecture).
- Also used to bypass IP-based authentication mechanisms.

How IP Spoofing is Used in Attacks

- **Man-in-the-Middle (MitM)**: Spoofing allows an attacker to intercept communications between two hosts.
- **Session Hijacking**: Attacker guesses the TCP sequence numbers and spoofs the IP to take over an authenticated session.
- **Blind Spoofing**: Attacker cannot see the return traffic but attempts to send commands (difficult, relies on predicting TCP sequence numbers).
- **Non-Blind Spoofing**: Attacker is on the same subnet and can see the return traffic, making session hijacking easier.

ARP Spoofing (ARP Poisoning)

- Address Resolution Protocol (ARP) maps IP addresses to physical MAC addresses on a local network.
- ARP has no authentication; hosts accept any ARP reply, even if they didn't ask for it.
- Attacker broadcasts forged ARP replies linking their MAC address to the IP address of the default gateway (router).
- Victim's traffic is sent to the attacker instead of the router.
- Enables sniffing, traffic modification, and DoS on a Local Area Network (LAN).

DNS Spoofing and Cache Poisoning

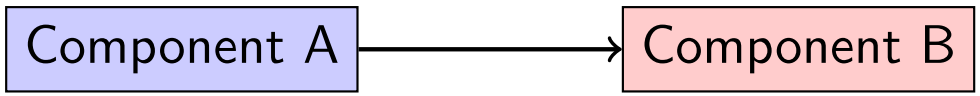
- Domain Name System (DNS) translates human-readable domain names to IP addresses.
- **DNS Spoofing**: Intercepting a DNS request and returning a forged IP address.
- **Cache Poisoning**: Sending forged DNS data to a DNS resolver, tricking it into caching the wrong IP for a domain.
- **Result**: Users typing 'www.bank.com' are secretly redirected to the attacker's phishing server.
- Dangerous because the URL in the browser looks completely legitimate.

Other Types of Spoofing

- **Email Spoofing**: Forging the 'From' address in an email. Core component of phishing. (Mitigated by SPF, DKIM, DMARC).
- **Caller ID Spoofing**: Falsifying the telephone number displayed on Caller ID to impersonate tech support or government agencies.
- **MAC Spoofing**: Changing the factory-assigned MAC address of a network interface to bypass MAC filtering or hide identity.
- **GPS Spoofing**: Broadcasting fake GPS signals to alter the location data of receivers (drones, ships).

Introduction to Network Sniffing

- Sniffing involves capturing network packets traversing the network.
- **Legitimate use**: Network administrators troubleshooting performance or analyzing protocols.
- **Malicious use**: Attackers capturing passwords, cookies, proprietary data, or mapping the network.
- Can be performed on wired (Ethernet) or wireless (Wi-Fi) networks.
- Effectiveness depends on network topology (hubs vs. switches) and encryption.



Definition

This is a key concept in Information Security.

Passive vs. Active Sniffing

- **Passive Sniffing**: Done on older Hub-based networks or open Wi-Fi. All traffic is broadcast to all ports; the attacker simply listens.
- **Active Sniffing**: Done on Switch-based networks. Switches isolate traffic to specific ports.
- To sniff on a switch, the attacker must force the switch to broadcast traffic.
- Methods include MAC Flooding (filling the switch's CAM table so it falls back to hub behavior) or ARP Spoofing.
- Active sniffing leaves a network footprint and can be detected.

Promiscuous Mode and Hardware

- By default, a Network Interface Card (NIC) ignores packets not destined for its own MAC address.
- Promiscuous Mode configures the NIC to pass all received traffic to the CPU, regardless of destination MAC.
- Required for a machine to act as a packet sniffer.
- **Network Taps**: Hardware devices inserted inline between network cables to copy traffic transparently.
- **SPAN/Mirror Ports**: Switch configurations that copy traffic from one port to another for monitoring.

Common Sniffing Tools

- **Wireshark**: The industry standard graphical protocol analyzer. Deep inspection of hundreds of protocols.
- **tcpdump**: Command-line packet analyzer for Linux/Unix. Excellent for quick troubleshooting and scripting.
- **Cain and Abel**: Historical Windows tool famous for ARP poisoning and password sniffing.
- **Ettercap / Bettercap**: Comprehensive suites for MitM attacks, ARP spoofing, and credential harvesting.
- **Kismet**: Specialized tool for passive wireless network sniffing.

The Danger of Plaintext Protocols

- Sniffing is highly effective against protocols that transmit data in plaintext.
- **Telnet**: Sends all keystrokes, including passwords, in plaintext. (Replace with SSH).
- **FTP**: Transmits credentials and files unencrypted. (Replace with SFTP).
- **HTTP**: All web traffic is visible to sniffers. (Replace with HTTPS).
- **POP3 / IMAP (without TLS)**: Email contents and passwords exposed.
- If an attacker sniffs a plaintext protocol, the data is immediately compromised.

Defending Against Spoofing & Sniffing

- **Use Encryption**: End-to-end encryption (VPNs, HTTPS, SSH) renders sniffed payloads useless.
- **Port Security**: Configure switches to restrict MAC addresses per port (prevents MAC flooding/spoofing).
- **Dynamic ARP Inspection (DAI)**: Switch feature that intercepts and validates ARP packets to prevent ARP spoofing.
- **Anti-Spoofing Filters**: Routers drop packets originating from the internet that have internal source IPs (Ingress filtering).
- **Use DNSSEC**: Adds cryptographic signatures to DNS records to prevent cache poisoning.

Introduction to Firewalls

- A firewall is a network security device or software that monitors and filters incoming and outgoing network traffic.
- Operates based on an organization's previously established security policies.
- Acts as a barrier between a trusted, secure internal network and another outside network (like the Internet).
- Can be implemented as hardware appliances, software on operating systems, or cloud-based services.
- The first line of defense in network security.

Definition

This is a key concept in Information Security.

Core Functions of a Firewall

- **Traffic Filtering**: Allowing or dropping packets based on specific criteria.
- **Network Address Translation (NAT)**: Hiding internal IP addresses behind a single public IP.
- **Logging and Auditing**: Recording traffic flows, dropped packets, and administrative access for analysis.
- **VPN Termination**: Acting as the endpoint for encrypted Virtual Private Network tunnels.
- **Intrusion Prevention (in modern firewalls)**: Actively scanning for malicious payloads.

Generations of Firewall Technology

- **First Generation**: Stateless Packet Filters.
- **Second Generation**: Stateful Inspection Firewalls.
- **Third Generation**: Application Layer / Proxy Firewalls.
- **Modern Generation**: Next-Generation Firewalls (NGFW) and Unified Threat Management (UTM).
- Each generation added deeper inspection capabilities across the OSI model.

Stateless Packet Filtering Firewalls

- Operates at the Network (Layer 3) and Transport (Layer 4) layers of the OSI model.
- Evaluates each packet individually, without context of previous packets.
- **Filtering criteria**: Source/Destination IP, Source/Destination Port, Protocol (TCP/UDP/ICMP).
- Uses Access Control Lists (ACLs) composed of allow and deny rules.
- Rules are evaluated sequentially top-to-bottom.

Pros and Cons of Stateless Firewalls

- **Pros**: Extremely fast processing, low memory overhead, good for high-speed core routing.
- **Cons**: Susceptible to IP spoofing attacks.
- **Cons**: Cannot track connection state, making protocols involving dynamic ports (like FTP) difficult to manage.
- **Cons**: Easily bypassed by fragmenting packets or manipulating TCP flags.
- Today, mostly used in core routers as simple ACLs rather than edge security.

Stateful Inspection Firewalls

- Developed by Check Point in the mid-1990s.
- Maintains a 'State Table' tracking all active connections (TCP handshakes, UDP streams).
- Only allows incoming packets if they are part of an already established and tracked outbound connection.
- **Understands context**: It knows if a packet is a legitimate response or unsolicited.
- Operates up to Layer 4 but has contextual awareness of the connection state.

How a State Table Works

- **When an internal host sends a TCP SYN packet out, the firewall adds an entry**: [Src IP, Src Port, Dst IP, Dst Port, State: SYN_SENT].
- When the SYN-ACK returns, the state updates to SYN_RECV.
- When the ACK is sent, the state becomes ESTABLISHED.
- Any external packet trying to enter that doesn't match an ESTABLISHED session in the table is dropped immediately.
- Connections are removed from the table upon TCP FIN/RST or a timeout.

Pros and Cons of Stateful Firewalls

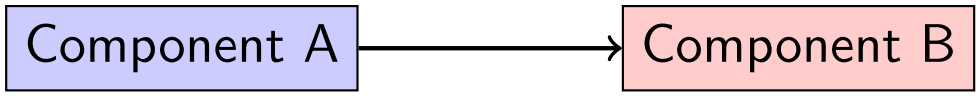
- **Pros**: Highly secure against basic spoofing and unauthorized inbound connections.
- **Pros**: Eliminates the need to open hundreds of high-numbered ports for return traffic.
- **Cons**: Higher CPU and memory usage compared to stateless filters.
- **Cons**: State tables can be overwhelmed by DDoS attacks (like SYN floods), causing the firewall to crash or drop new connections.
- **Cons**: Does not inspect the actual payload (Layer 7 data) for malware.

Firewall Rule Processing (ACLs)

- Firewall rules are processed in a top-down, sequential order.
- As soon as a packet matches a rule, the action (Allow or Deny) is taken, and processing stops.
- **Implicit Deny**: Most firewalls have an invisible 'Deny All' rule at the very bottom.
- If a packet doesn't match any explicit allow rules, it is dropped by default.
- Rule organization is critical for performance and security.

Firewall Architectures: Bastion Host

- A Bastion Host is a specialized computer deliberately exposed on a public network.
- Highly hardened: All unnecessary services, ports, and software are removed.
- Used as a gateway (e.g., a proxy server) between the internal network and the internet.
- Often has two network interface cards (Dual-homed host).
- If compromised, the attacker still must breach the internal network.



Summary of Part 1

- Firewalls enforce network boundaries.
- Stateless firewalls filter by IP/Port but lack memory.
- Stateful firewalls track connections via a state table, offering better security.
- Neither inspects the actual application data payload for malware or exploits.
- **Next time**: We will explore Proxy firewalls, Next-Gen firewalls, and DMZ architectures.

Introduction to Advanced Firewalls

- Stateful firewalls are blind to the application payload.
- Attackers started tunneling malware over allowed ports (e.g., HTTP over Port 80).
- A stateful firewall will allow the traffic because Port 80 is open.
- Advanced firewalls must inspect Layer 7 (Application Layer) to stop these attacks.
- This led to the creation of Proxy Firewalls and Next-Generation Firewalls (NGFW).

Definition

This is a key concept in Information Security.

Application/Proxy Firewalls

- Acts as an intermediary between the client and the server.
- The client never connects directly to the server; it connects to the proxy.
- The proxy inspects the request at the Application Layer (Layer 7).
- If safe, the proxy establishes a second, separate connection to the destination server on behalf of the client.
- Operates transparently to the user, though configuration may be required.

Pros and Cons of Proxy Firewalls

- **Pros**: Deepest level of inspection; can understand specific commands (e.g., block FTP 'PUT' but allow 'GET').
- **Pros**: Excellent caching and content filtering capabilities (blocking specific URLs).
- **Cons**: High latency and performance overhead due to breaking and rebuilding connections.
- **Cons**: Requires a specific proxy for each application protocol (HTTP proxy, SMTP proxy).
- **Cons**: Can break some client-server applications that do not support proxies.

Next-Generation Firewalls (NGFW)

- Combines stateful inspection with deep packet inspection (DPI).
- Unlike proxies, NGFWs don't always break the connection; they inspect payloads inline at wire speed.
- Integrates Intrusion Prevention Systems (IPS) directly into the firewall.
- **Threat Intelligence Feeds**: Automatically blocks known malicious IPs and domains dynamically.
- Replaces the need for multiple standalone security appliances.

Application Awareness and Identity in NGFWs

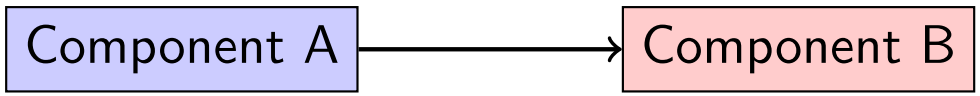
- Legacy firewalls filter by Port (e.g., Port 80 = Web).
- NGFWs filter by Application (e.g., allows Facebook, but blocks Facebook Games).
- Can identify applications even if they use non-standard ports or encrypt traffic.
- **Identity Awareness**: Integrates with Active Directory to create rules based on Usernames, not just IP addresses.
- **Example rule**: 'Allow HR Department to use Dropbox, block everyone else'.

Web Application Firewalls (WAF)

- A specialized proxy firewall designed specifically for web applications (HTTP/HTTPS).
- Deployed in front of web servers to protect against OWASP Top 10 attacks.
- Inspects traffic for SQL Injection, Cross-Site Scripting (XSS), and session hijacking.
- Unlike network firewalls (which protect the whole network), WAFs protect specific web apps.
- Can be deployed on-premise or via cloud (e.g., AWS WAF, Cloudflare).

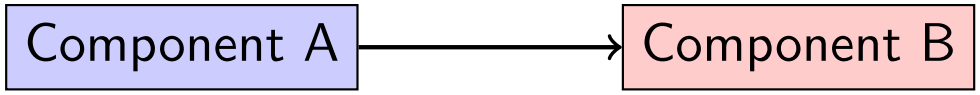
Firewall Architectures: Screened Host

- Combines a packet-filtering router with a bastion host.
- The router (screening router) sits on the edge and routes all incoming traffic to the bastion host.
- The bastion host acts as an application proxy.
- Internal hosts can only communicate with the internet via the proxy on the bastion host.
- Provides two layers of defense, but the internal network is fully exposed if the bastion host is compromised.



Firewall Architectures: Demilitarized Zone (DMZ)

- A DMZ is an isolated sub-network that contains public-facing services (Web, Email, DNS).
- Creates an intermediate zone between the untrusted internet and the trusted internal network.
- External users can access the DMZ but are blocked from the internal network.
- Internal users can access both the DMZ and the internet.
- If a web server in the DMZ is hacked, the internal network remains protected.



DMZ Configuration: Screened Subnet

- The most secure and common DMZ architecture.
- **Uses Two Firewalls**: An external firewall and an internal firewall.
- **External Firewall**: Allows internet traffic into the DMZ.
- **Internal Firewall**: Strictly controls traffic from the DMZ into the internal network.
- Often uses firewalls from two different vendors to prevent a single zero-day exploit from breaching both.

Firewall Best Practices

- **Default Deny**: Always end rule sets with an implicit deny all.
- **Principle of Least Privilege**: Only open ports and protocols explicitly required for business.
- **Rule Auditing**: Regularly review rules to remove obsolete or redundant ones.
- **Change Management**: Document who requested a rule change and why.
- **Logging**: Forward all firewall logs to a centralized SIEM for monitoring and alerting.

Introduction to IDS

- Intrusion Detection System (IDS) is a monitoring system that detects suspicious activities and generates alerts.
- It acts like a burglar alarm for your network or computer.
- Designed to catch attacks that bypass the firewall.
- **Passive Device**: It monitors and alerts, but does not take action to drop the traffic.
- Detects malware, policy violations, and unauthorized access attempts.

Definition

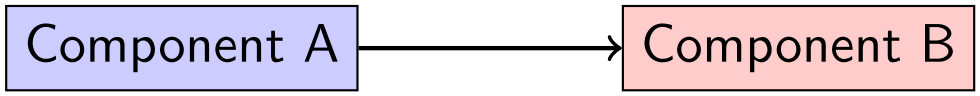
This is a key concept in Information Security.

Firewall vs. IDS

- **Firewall**: Active access control device. Determines what is allowed in based on rules (IP/Port).
- **IDS**: Passive monitoring device. Evaluates allowed traffic to see if its contents are malicious.
- Firewalls sit at the perimeter; IDS sensors are placed internally to catch internal threats and firewall bypasses.
- Firewalls prevent intrusions; IDS detects intrusions that slip through.
- Both are necessary for a defense-in-depth strategy.

Network-Based IDS (NIDS)

- Monitors traffic for an entire network segment.
- Deployed via a network tap or switch SPAN/mirror port (promiscuous mode).
- Analyzes packets as they traverse the network.
- Cannot see inside encrypted traffic (unless deployed with SSL decryption).
- Cannot determine if an attack against a specific host was actually successful.



Host-Based IDS (HIDS)

- Software agent installed on a specific host (server or endpoint).
- **Monitors internal host activity**: OS logs, file system modifications, memory, and system calls.
- **File Integrity Monitoring (FIM)**: Alerts if critical system files are altered.
- Can determine if an attack was successful by observing the host's reaction.
- Resource intensive for the host and harder to manage at scale.

Signature-Based Detection

- Compares network traffic or system activity against a database of known threat signatures.
- A 'signature' is a specific pattern, such as a known byte sequence in malware or a specific malicious URL.
- Works similarly to traditional antivirus software.
- Very fast and highly accurate for known attacks.
- Requires constant updates to the signature database.

Pros and Cons of Signature Detection

- **Pros**: Extremely low false positive rate (it only alerts on definite matches).
- **Pros**: Fast processing; easy to understand the generated alerts.
- **Cons**: Zero-Day Vulnerability: Completely blind to new, unknown attacks.
- **Cons**: Easily evaded by slight modifications to the attack payload (e.g., changing the code slightly).
- **Cons**: Managing and storing massive signature databases.

Anomaly-Based Detection (Heuristics)

- Creates a baseline of 'normal' network or host behavior over a learning period.
- Monitors for deviations from this baseline.
- E.g., If a server usually sends 10MB of data at 2 AM, and suddenly sends 10GB, an alert is triggered.
- Capable of detecting Zero-Day attacks and insider threats.
- Increasingly driven by Machine Learning and AI models.

Pros and Cons of Anomaly Detection

- **Pros**: Can detect entirely new, unknown attacks (Zero-Days).
- **Pros**: Can detect insider threats (e.g., an employee accessing unusual files).
- **Cons**: High False Positive rate (legitimate but unusual behavior triggers alerts).
- **Cons**: Requires a long learning period to establish an accurate baseline.
- **Cons**: Attackers can slowly alter network behavior over time to create a new, malicious baseline.

False Positives vs. False Negatives

- **True Positive**: Legitimate attack detected (Good).
- **True Negative**: Normal traffic ignored (Good).
- **False Positive**: Normal traffic flagged as an attack (Bad - causes alert fatigue).
- **False Negative**: Legitimate attack goes undetected (Worst - results in a breach).
- IDS tuning is the constant struggle to minimize both False Positives and False Negatives.

Honeypots and Honeynets

- A Honeypot is a decoy system designed to lure attackers away from real systems.
- It contains fake data and vulnerabilities.
- Any traffic sent to a honeypot is inherently suspicious, providing high-fidelity alerts.
- **Honeynet**: A network of multiple honeypots.
- Used by security researchers to study attacker tactics, techniques, and procedures (TTPs).

Summary of IDS

- IDS provides visibility into attacks that bypass perimeter defenses.
- NIDS monitors the network; HIDS monitors the endpoint.
- Signatures catch known threats; Anomaly detection catches unknown threats.
- Because IDS is passive, manual intervention is required to stop an ongoing attack.
- **Next lecture**: How we automate the response with Intrusion Prevention Systems (IPS).

Introduction to IPS

- Intrusion Prevention System (IPS) extends IDS by actively blocking detected threats.
- While IDS is a passive alarm, IPS is an active guard.
- Deployed 'in-line' with network traffic, meaning traffic must pass through the IPS to reach its destination.
- Can drop malicious packets, block IP addresses, or reset TCP connections dynamically.
- Requires extremely high confidence in detection to avoid blocking legitimate traffic.

Definition

This is a key concept in Information Security.

IDS vs. IPS: Key Differences

- **Deployment**: IDS is out-of-band (promiscuous mode); IPS is in-line.
- **Action**: IDS alerts; IPS drops packets and alters configurations.
- **Impact of Failure**: If IDS fails, it stops alerting. If IPS fails (hardware crash), it can bring down the entire network (unless deployed with a 'fail-open' configuration).
- **Latency**: IDS adds zero latency to traffic. IPS adds minor latency because it inspects traffic before forwarding.
- **Risk**: IPS carries the risk of False Positives causing business disruption.

How IPS Stops Attacks

- **Dropping Packets**: Silently discarding the malicious packet.
- **TCP Reset (RST)**: Injecting a TCP RST packet to forcefully terminate the malicious connection.
- **Dynamic ACL Updates**: Sending a command to the edge firewall or router to block the attacking IP address permanently.
- **Traffic Scrubbing**: Removing the malicious payload but allowing the rest of the legitimate session to continue.
- **Quarantine**: Moving an infected host (HIPS) to an isolated VLAN.

Types of IPS Deployment

- **Network-Based IPS (NIPS)**: Inspects traffic across the entire network segment. Placed behind the firewall.
- **Host-Based IPS (HIPS)**: Installed on endpoints. Can prevent malicious processes from executing or accessing system memory.
- **Wireless IPS (WIPS)**: Monitors Wi-Fi radio spectrum for rogue access points, de-authentication attacks, and MAC spoofing.
- **Network Behavior Analysis (NBA)**: Specialized IPS focusing on DoS attacks, policy violations, and anomaly detection over time.

The Risk of False Positives in IPS

- In an IDS, a false positive wastes an analyst's time.
- In an IPS, a false positive blocks legitimate business traffic (e.g., blocking a critical database update or a customer payment).
- Therefore, IPS relies heavily on highly tuned Signature-Based Detection rather than Anomaly-Based Detection.
- Many organizations deploy an IPS in 'IDS-mode' (alert only) for weeks to tune rules before enabling 'blocking' mode.

Unified Threat Management (UTM)

- Historically, organizations bought separate hardware for Firewalls, VPNs, IDS, IPS, and Antivirus.
- UTM appliances consolidate all these features into a single device.
- **Pros**: Cheaper, easier to manage, single pane of glass for administration.
- **Cons**: Single point of failure, can suffer from performance bottlenecks if all features are enabled.
- Modern Next-Generation Firewalls (NGFW) essentially encompass UTM capabilities.

Open Source IDS/IPS Solutions

- **Snort**: The oldest and most famous open-source NIDS/NIPS, developed by Sourcefire (now Cisco). Uses a widely adopted rule syntax.
- **Suricata**: A modern alternative to Snort. Highly multi-threaded, offering better performance on multi-core CPUs.
- **Zeek (formerly Bro)**: Focuses heavily on network analysis and metadata extraction rather than just signatures.
- **OSSEC**: A popular open-source HIDS platform for log analysis and file integrity monitoring.

Evading IDS/IPS

- Attackers use techniques to bypass IPS inspection.
- **Fragmentation**: Breaking the attack payload into tiny TCP fragments. The IPS might fail to reassemble them properly before inspection.
- **Encryption**: Hiding the payload inside SSL/TLS tunnels. If the IPS doesn't have the decryption keys, it cannot inspect the payload.
- **Obfuscation**: Encoding the payload (e.g., Base64 or URL encoding) to avoid signature matching.
- **Timing Attacks**: Sending packets incredibly slowly to evade state table tracking.

Overcoming Evasion: SSL Inspection

- Over 80% of web traffic is encrypted (HTTPS), making traditional IPS blind.
- SSL/TLS Inspection (SSL Decryption) solves this.
- The IPS acts as a Man-in-the-Middle (MitM) authorized by the organization.
- It decrypts the traffic, inspects the payload, re-encrypts it, and sends it to the destination.
- Requires deploying enterprise root certificates to all internal endpoints to avoid browser warnings.

Integration with SIEM

- Security Information and Event Management (SIEM) systems aggregate logs from Firewalls, IDS, IPS, servers, and endpoints.
- **Examples**: Splunk, IBM QRadar, ELK Stack.
- **Provides correlation**: 'The firewall saw a connection, the IPS triggered an alert, and the HIDS saw a new user created on the server'.
- Correlating these logs confirms a successful breach rather than just an isolated alert.

Introduction to VPNs

- A Virtual Private Network (VPN) creates a secure, encrypted connection over a less secure network, such as the Internet.
- It extends a private corporate network across a public network.
- Enables users to send and receive data as if their computing devices were directly connected to the private network.
- Ensures Confidentiality, Integrity, and Authentication of data in transit.
- Fundamental tool for remote work and branch office connectivity.

Definition

This is a key concept in Information Security.

Core Mechanisms of a VPN

- **Tunneling**: Encapsulating a network protocol within another network protocol (e.g., placing an IP packet inside another IP packet).
- **Encryption**: Securing the encapsulated payload so that intermediaries cannot read the data (Confidentiality).
- **Hashing/HMAC**: Ensuring the data was not tampered with in transit (Integrity).
- **Authentication**: Verifying the identity of the user or device connecting to the VPN (Authentication).
- Without encryption, it's just a tunnel (like GRE); with encryption, it becomes a VPN.

Types of VPNs

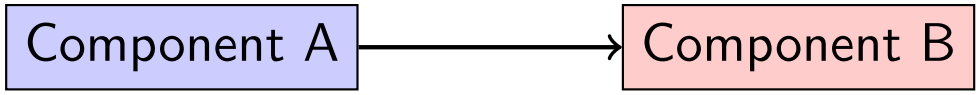
- **Remote Access VPN**: Connects individual users (e.g., telecommuters) to a corporate network. Requires client software on the endpoint.
- **Site-to-Site VPN**: Connects entire networks (e.g., a branch office to headquarters). Handled by routers/firewalls; transparent to end users.
- Client-to-Site is typically used for human access.
- Site-to-Site is typically used for infrastructure connectivity.
- Both rely on standard protocols like IPsec or SSL/TLS.

IPsec (Internet Protocol Security) Overview

- IPsec is a suite of protocols for securing IP communications at the Network Layer (Layer 3).
- Standardized by the IETF, highly interoperable between different vendors (Cisco to Juniper, etc.).
- Requires dedicated client software for Remote Access, making deployment more complex.
- The industry standard for Site-to-Site VPNs.
- Provides authentication, integrity, and confidentiality.

IPsec Architecture: AH and ESP

- Authentication Header (AH): Provides data integrity and origin authentication, but NO encryption. (Rarely used alone today).
- Encapsulating Security Payload (ESP): Provides encryption (confidentiality), plus integrity and authentication.
- ESP encrypts the payload, ensuring that sniffers only see cipher text.
- Modern IPsec VPNs almost exclusively use ESP.
- ESP uses protocols like AES for encryption and SHA for hashing.



IPsec Modes: Transport vs. Tunnel Mode

- **Transport Mode**: Only the payload (data) of the IP packet is encrypted/authenticated. The original IP header remains intact. Used for host-to-host communications.
- **Tunnel Mode**: The entire original IP packet is encrypted and encapsulated inside a new IP packet with a new header.
- Tunnel Mode is used for network-to-network (Site-to-Site) VPNs, hiding internal IP addresses from the public internet.
- Most corporate VPNs operate in Tunnel Mode.

Internet Key Exchange (IKE)

- IKE is the protocol used to set up the IPsec security associations (SAs).
- **Phase 1**: Establishes a secure, authenticated channel between the two peers. (Main Mode or Aggressive Mode).
- **Phase 2**: Negotiates the specific IPsec parameters (encryption/hash algorithms) for the actual data tunnel.
- Uses Diffie-Hellman for secure key exchange over the untrusted network.
- IKEv2 is the modern standard, offering faster connection times and better resilience to network changes.

SSL/TLS VPNs

- Operates at the Application/Transport Layer (Layer 4/7).
- Uses the same TLS encryption that secures HTTPS websites.
- **Clientless VPN**: Can be accessed via a standard web browser without installing custom VPN software.
- Provides granular access control (e.g., granting access to a specific internal web app, not the whole network).
- Widely used for Remote Access VPNs due to ease of deployment.

Differences: IPsec vs. SSL VPN

- **OSI Layer**: IPsec (Layer 3) vs SSL (Layer 4/7).
- **Use Case**: IPsec (Site-to-Site) vs SSL (Remote Access).
- **Access**: IPsec provides access to the entire remote network; SSL often restricts access to specific applications.
- **Client**: IPsec requires heavy client software; SSL can be clientless (browser-based).
- **Firewall Traversal**: IPsec can be blocked by NAT/Firewalls (requires NAT-T); SSL (HTTPS/Port 443) passes easily through most firewalls.

Modern VPN Protocols: OpenVPN and WireGuard

- **OpenVPN**: Highly configurable open-source VPN protocol. Can run over TCP or UDP. Uses OpenSSL for encryption. Slower but highly compatible.
- **WireGuard**: The newest VPN protocol. Built directly into the Linux kernel.
- WireGuard uses state-of-the-art cryptography (Curve25519, ChaCha20).
- Has a tiny codebase (˜4,000 lines of code) compared to OpenVPN/IPsec (˜100,000+), making it faster and easier to audit.
- Rapidly becoming the industry standard for performance and security.

VPN Topologies Overview

- When designing a VPN infrastructure, the topology determines how different sites and users interact.
- Impacts network performance, scalability, and security.
- **Primary topologies for Site-to-Site VPNs**: Hub-and-Spoke and Full Mesh.
- **Primary routing decisions for Remote Access**: Full Tunnel vs. Split Tunnel.
- Topology choice depends on the size of the organization and traffic patterns.

Hub-and-Spoke VPN Topology

- One central location (the Hub) connects to multiple remote branches (the Spokes).
- Branches do NOT communicate directly with each other.
- If Branch A wants to send data to Branch B, traffic must travel to the Hub first, and then back out to Branch B.
- **Pros**: Easy to manage, centralized security inspection at the Hub.
- **Cons**: Creates a bottleneck at the Hub; higher latency for spoke-to-spoke traffic.

Full Mesh VPN Topology

- Every site maintains a direct VPN tunnel to every other site.
- If Branch A talks to Branch B, traffic goes directly between them.
- **Pros**: Optimal routing, lowest latency, high redundancy (no single point of failure).
- **Cons**: Very complex to manage. A network with N sites requires N(N-1)/2 tunnels.
- A 10-site network requires 45 tunnels; a 50-site network requires 1,225 tunnels.

Split Tunneling vs. Full Tunneling

- Applies to Remote Access VPNs.
- **Full Tunneling**: ALL internet traffic from the remote user goes through the VPN to corporate HQ, even general web browsing.
- **Split Tunneling**: Only traffic destined for corporate resources goes through the VPN. General internet traffic goes directly out the user's local ISP.
- The decision is based on routing tables configured by the VPN client.

Security Risks of Split Tunneling

- Why do security teams hate Split Tunneling?
- If an employee is at a coffee shop on an insecure Wi-Fi network, their general web traffic is exposed.
- More dangerously, the laptop acts as a bridge. A hacker on the coffee shop Wi-Fi could infect the laptop, and use the active VPN tunnel to pivot into the corporate network.
- Full Tunneling ensures the employee benefits from the corporate firewall, IPS, and web filtering, regardless of their physical location.

Implementing VPNs: Hardware vs. Software

- **Hardware VPNs (Firewalls/Routers)**: Dedicated appliances with specialized ASICs for cryptographic acceleration (e.g., Palo Alto, Cisco ASA). Used for high-throughput Site-to-Site links.
- **Software VPNs**: Runs on standard servers or VMs (e.g., OpenVPN on Linux, pfSense). Flexible and cheaper, but rely on CPU for encryption.
- **Cloud VPNs (VPN as a Service - VPNaaS)**: AWS Client VPN or Azure VPN Gateway. Easily scalable for remote workforces without buying hardware.

VPN Authentication Mechanisms

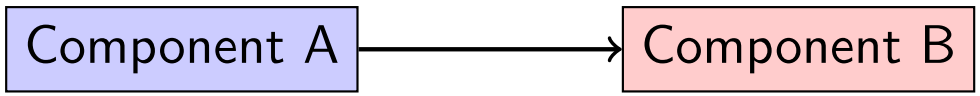
- **Pre-Shared Keys (PSK)**: A simple password shared between two VPN endpoints. Easy to setup, but if compromised, the whole tunnel is breached. Hard to scale.
- **Digital Certificates (PKI)**: Uses public/private key pairs and a Certificate Authority (CA) to authenticate endpoints. Highly secure and scalable.
- **RADIUS/TACACS+**: For remote access, the VPN gateway forwards user credentials to a central server to verify identity.
- **LDAP / Active Directory Integration**: Verifying the user against the corporate directory.

Multi-Factor Authentication (MFA) in VPNs

- Compromised VPN credentials are a leading cause of ransomware breaches.
- Passwords alone are insufficient for remote access.
- MFA requires something you know (password) and something you have (a push notification to a phone, an RSA token, or a YubiKey).
- Virtually all modern remote access VPNs enforce MFA via SAML integration (e.g., Okta, Duo).

Zero Trust Network Access (ZTNA)

- The modern alternative/evolution of the Remote Access VPN.
- Traditional VPN gives a user access to the *entire network* once authenticated (Perimeter model).
- ZTNA assumes no network is trusted. Users are granted access only to *specific applications*, not the underlying network.
- Continuously verifies user identity, device health, and location before granting access.
- Reduces the attack surface significantly compared to full VPNs.



Summary and VPN Best Practices

- Use strong encryption (AES-256) and modern protocols (IKEv2, WireGuard).
- Enforce MFA for all human-interactive VPN sessions.
- Avoid Split Tunneling for high-security environments.
- Monitor VPN logs in a SIEM for impossible travel anomalies (e.g., user logs in from NY and China within 10 minutes).
- Begin migrating to ZTNA for remote workforce security.

Introduction to Wireless Security

- Wireless networks (Wi-Fi) transmit data over radio waves using the IEEE 802.11 standard.
- Unlike wired networks where an attacker needs physical access to a cable, wireless signals travel through the air and walls.
- Anyone with an antenna within range can intercept the signals.
- Therefore, strong encryption is the ONLY way to secure a wireless network.
- The history of Wi-Fi is a history of broken encryption standards.

Definition

This is a key concept in Information Security.

Wireless Terminology

- **Access Point (AP)**: The hardware device bridging the wireless and wired networks.
- **SSID (Service Set Identifier)**: The name of the wireless network (e.g., 'Cafe_WiFi').
- **BSSID (Basic Service Set Identifier)**: The MAC address of the Access Point.
- **Client (Station/STA)**: The endpoint connecting to the AP (laptop, phone).
- **Band**: The radio frequency used (2.4 GHz or 5 GHz).

Common Wireless Attacks

- **War Driving**: Driving around a city with a laptop and antenna mapping unsecured Wi-Fi networks.
- **Rogue Access Point**: An unauthorized AP plugged into the corporate network by an employee (bypassing the firewall).
- **Evil Twin**: An attacker sets up a malicious AP with the exact same SSID as the corporate network, tricking users into connecting to it.
- **De-authentication Attack**: Sending spoofed 'disconnect' frames to knock users off the legitimate network (often forcing them to connect to an Evil Twin).

Ineffective Defense: SSID Hiding

- Administrators often configure APs to stop broadcasting the SSID, thinking it makes the network invisible.
- However, the SSID is still transmitted in plaintext during the probe request and association phases when a legitimate client connects.
- An attacker simply uses a sniffer (like Airodump-ng), sends a De-auth frame to a connected client, and reads the hidden SSID when the client reconnects.
- **Security by obscurity**: It provides no real security.

Ineffective Defense: MAC Filtering

- Configuring the AP to only allow specific client MAC addresses to connect.
- **Flaw**: MAC addresses are sent in plaintext over the air, even on encrypted networks.
- An attacker sniffs the air, captures a legitimate, allowed MAC address, and then spoofs their own network card to match that MAC address.
- Provides a false sense of security and adds massive administrative overhead.
- Easily defeated by MAC spoofing tools (like macchanger).

Introduction to WEP (Wired Equivalent Privacy)

- Introduced in 1997 with the original 802.11 standard.
- **Goal**: Provide security equivalent to a physical wired network.
- Used a Pre-Shared Key (PSK) of either 64-bit or 128-bit length.
- Used the RC4 stream cipher for encryption.
- Considered completely broken and obsolete today. Can be cracked in minutes.

Definition

This is a key concept in Information Security.

How WEP Encryption Worked

- WEP concatenates a 24-bit Initialization Vector (IV) with the pre-shared key.
- The resulting string is used as the seed for the RC4 cipher to generate a keystream.
- The plaintext data is XORed with the keystream to produce the ciphertext.
- The 24-bit IV is transmitted in plaintext along with the packet so the receiver knows which IV was used.
- CRC-32 is used for integrity checking (ICV).

The Fatal Flaw of WEP: The IV

- The Initialization Vector (IV) is only 24 bits long.
- 2^24 = 16.7 million possible IVs.
- On a busy network transmitting thousands of packets a second, the IV space is exhausted in a few hours.
- When the IV space is exhausted, IVs must be reused (IV Collision).
- Because the Pre-Shared Key is static, reusing an IV means reusing the exact same keystream.

Cracking WEP (FMS Attack)

- Fluhrer, Mantin, and Shamir (FMS) published an attack showing that certain 'weak IVs' leaked information about the secret key.
- An attacker passively captures millions of WEP packets.
- Using a tool like Aircrack-ng, the attacker isolates the weak IVs.
- Statistical analysis of the weak IVs recovers the static WEP key entirely.
- To speed up the process, attackers use Packet Injection (ARP Replay) to force the AP to generate millions of IVs rapidly.

Summary of WEP's Failure

- WEP failed on all three pillars of security.
- **Confidentiality**: Broken due to short IVs and RC4 weaknesses.
- **Integrity**: CRC-32 is linear and unkeyed; attackers can easily manipulate the ciphertext and forge the checksum.
- **Authentication**: No user authentication, just device authentication via a shared password.
- Never use WEP. It is a massive compliance violation today.

The WPA Stopgap

- Wi-Fi Protected Access (WPA) was released in 2003 as an interim solution to replace WEP.
- **Hardware limitation**: Millions of devices had Wi-Fi cards with hardware built only for the RC4 cipher.
- WPA had to fix WEP's security flaws but still run on older WEP hardware via firmware updates.
- Introduced TKIP (Temporal Key Integrity Protocol) to address the IV reuse and integrity issues of WEP.
- WPA is better than WEP, but still considered deprecated.

How TKIP Improved WEP

- **Increased IV size**: Upgraded from 24-bit to 48-bit, effectively eliminating IV reuse collisions.
- **Per-Packet Keying**: Dynamically generates a new unique key for every single packet, rather than using the same static key for everything.
- **Message Integrity Check (MIC)**: Replaced the weak CRC-32 with an algorithm called 'Michael' to prevent packet tampering.
- Defeated the FMS attack used against WEP.
- However, cryptographic flaws were eventually found in TKIP as well.

Introduction to WPA2 (802.11i)

- Released in 2004, WPA2 is the fully realized 802.11i security standard.
- Abandoned the flawed RC4 cipher entirely.
- Introduced AES (Advanced Encryption Standard) for encryption.
- Introduced CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) for both encryption and integrity.
- Required new hardware, as older Wi-Fi cards lacked the CPU power to process AES encryption.

Definition

This is a key concept in Information Security.

WPA2 Personal vs. Enterprise

- **WPA2-Personal (PSK)**: Uses a single Pre-Shared Key (password) for all users. Easy for homes, bad for business (if an employee leaves, the password must be changed for everyone).
- **WPA2-Enterprise (802.1X/EAP)**: Integrates with a RADIUS server (like Active Directory).
- In Enterprise mode, every user logs in with their own unique username and password.
- The server dynamically generates a unique encryption key for that specific user session.
- If an employee leaves, IT simply disables their AD account; no need to change Wi-Fi passwords.

The WPA2 4-Way Handshake

- When a client connects to WPA2, they do not send the password over the air.
- Instead, the AP and Client perform a cryptographic 4-Way Handshake.
- They prove to each other that they both know the PSK without ever transmitting it.
- They derive a Pairwise Transient Key (PTK) which is used to encrypt the actual data session.
- This handshake is the only vulnerability point in WPA2-Personal.

Cracking WPA2 (Offline Dictionary Attack)

- WPA2 AES encryption cannot be cracked. The attack focuses on the handshake.
- Attacker forces a client off the network (De-auth attack).
- When the client reconnects, the attacker sniffs the 4-Way Handshake in the air.
- The attacker takes the captured handshake offline and runs a dictionary attack (using Hashcat or Aircrack-ng).
- They hash millions of passwords and compare the output to the captured handshake.
- If the password is weak (e.g., 'password123'), it will be cracked. If it is long and complex, WPA2 is secure.

The KRACK Attack (2017)

- Key Reinstallation AttaCK (KRACK) discovered a flaw in the WPA2 protocol itself.
- An attacker manipulates the 3rd step of the 4-way handshake, forcing the client to reinstall an already-in-use cryptographic key.
- This forces the client to reset the packet numbers and IVs, causing keystream reuse (similar to the WEP flaw).
- Allowed attackers to decrypt traffic and inject malware into HTTP streams.
- Fixed via firmware patches on client devices, but highlighted that WPA2 was aging.

Introduction to WPA3

- Released in 2018 to address the shortcomings of WPA2.
- Mandates Protected Management Frames (PMF) to prevent De-authentication attacks.
- Replaces the PSK 4-way handshake with SAE (Simultaneous Authentication of Equals).
- **SAE provides Forward Secrecy**: Even if an attacker learns the Wi-Fi password later, they cannot decrypt previously captured traffic.
- Provides robust protection against offline dictionary attacks.

Definition

This is a key concept in Information Security.

WPA3 OWE (Opportunistic Wireless Encryption)

- Addresses the security of Open public Wi-Fi networks (e.g., Coffee shops, Airports).
- On an open WPA2 network, traffic is unencrypted and easily sniffed.
- WPA3 OWE provides individualized encryption for open networks.
- Users don't need a password to connect, but the AP and the client silently negotiate unique encryption keys.
- Protects against passive sniffing, though it does not protect against Active MitM (Evil Twin) attacks since there is no authentication.

Summary of Wireless Security

- **WEP**: Broken, uses RC4, short IVs. Never use.
- **WPA**: Stopgap, uses RC4 + TKIP. Obsolete.
- **WPA2**: Uses AES + CCMP. Secure with strong passwords, but vulnerable to dictionary attacks.
- **WPA3**: Uses AES, replaces PSK with SAE. Defeats dictionary attacks and provides forward secrecy.
- Enterprise environments should always use WPA2/WPA3-Enterprise (802.1X).

4.1 Introduction to Web Security (Part 1)

- **Unit 4**: Web and Application Security
- Lecture 28
- **Course**: DI03016035

Definition

This is a key concept in Information Security.

Agenda

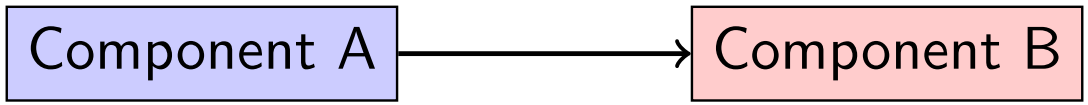
- What is Web Security?
- The Web Architecture
- Why are Web Applications Vulnerable?
- HTTP/HTTPS Fundamentals
- Client-Side vs Server-Side Security

What is Web Security?

- Protection of websites, web applications, and web services from attacks.
- Ensures Confidentiality, Integrity, and Availability (CIA) of web assets.
- Involves safeguarding user data from unauthorized access or modification.
- A shared responsibility between developers, admins, and end-users.

The Typical Web Architecture

- Client (Browser) -¿ Network -¿ Web Server -¿ Application Server -¿ Database.
- Each layer introduces its own set of vulnerabilities.
- Attackers often target the weakest link (e.g., outdated server software or poor input validation).
- Data must be protected in transit and at rest.



Why are Web Applications Vulnerable?

- **Complex Codebases**: Millions of lines of code with frequent updates.
- **Third-Party Dependencies**: Heavy reliance on external libraries and APIs.
- **User Input**: Web apps must accept input from untrusted users.
- **State Management**: HTTP is stateless; maintaining secure sessions is challenging.

HTTP Protocol Fundamentals

- HTTP (Hypertext Transfer Protocol) is plaintext by default.
- **Request Methods**: GET, POST, PUT, DELETE.
- **Headers**: Carry metadata (Cookies, User-Agent, Authorization).
- **Status Codes**: 200 OK, 403 Forbidden, 404 Not Found, 500 Server Error.

The Stateless Nature of HTTP

- Each HTTP request is independent.
- The server does not inherently remember previous requests.
- **Workaround**: Sessions and Cookies are used to track user state.
- If sessions are hijacked, attackers can impersonate users.

Client-Side Security

- Executes in the user's browser (HTML, CSS, JavaScript).
- **Threats**: Malicious scripts, UI redressing (Clickjacking).
- **Defenses**: Content Security Policy (CSP), Output Encoding.
- Never trust client-side validation for security purposes.

Server-Side Security

- Executes on the web/application server (PHP, Node.js, Python, Java).
- **Threats**: SQL Injection, Command Injection, Directory Traversal.
- **Defenses**: Input Validation, Parameterized Queries, Principle of Least Privilege.
- Server-side is the final line of defense.

Common Attack Vectors Overview

- Injection Attacks (SQLi, Command Injection).
- Cross-Site Scripting (XSS).
- Broken Authentication.
- Security Misconfigurations.

Summary

- Web security is essential to protect user data and maintain service availability.
- Web architectures have multiple attack surfaces.
- HTTP's stateless nature introduces session management risks.
- Both client-side and server-side defenses are required, but server-side is authoritative.

Introduction to Web Security (Part 2)

- Authentication, Authorization, and Sessions
- **Unit 4**: Web and Application Security
- Lecture 29

Definition

This is a key concept in Information Security.

Agenda

- Authentication vs. Authorization
- Authentication Mechanisms
- Session Management
- Cookies and Security Flags
- Tokens (JWT)

Authentication (AuthN)

- **Definition**: Verifying the identity of a user or system.
- 'Who are you?'
- **Factors**: Something you know (Password), Something you have (Token/Phone), Something you are (Biometrics).
- Multi-Factor Authentication (MFA) combines these for stronger security.

Authorization (AuthZ)

- **Definition**: Determining what an authenticated user is allowed to do.
- 'What are you allowed to do?'
- Requires successful authentication first.
- **Models**: Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC).

Session Management Basics

- HTTP is stateless; sessions map a sequence of requests to a single user.
- **Session ID**: A unique string assigned upon login.
- The Session ID is sent with every subsequent request.
- Server uses the ID to look up user state in memory or a database.

Definition

This is a key concept in Information Security.

Cookies for Session Storage

- Cookies are small pieces of data stored in the user's browser.
- Automatically sent to the server with every HTTP request to the same domain.
- Most common way to store Session IDs.
- Vulnerable to interception if sent over plaintext HTTP.

Securing Cookies: Important Flags

- **Secure Flag**: Cookie is only sent over encrypted HTTPS connections.
- **HttpOnly Flag**: Prevents JavaScript from accessing the cookie (mitigates XSS).
- **SameSite Flag**: Controls cross-origin cookie sending (mitigates CSRF).
- **Attributes**: Strict, Lax, None.

Token-Based Authentication

- Stateless alternative to traditional server-side sessions.
- JSON Web Tokens (JWT) are the most popular format.
- Token is sent in the Authorization header (Bearer token).
- Server validates the token's cryptographic signature without looking up a database.

JWT Structure

- Header: Algorithm and token type.
- Payload: Claims (user data, expiry time).
- Signature: Ensures the token hasn't been altered.
- Format: header.payload.signature (Base64Url encoded).

Common Authentication Flaws

- Weak passwords and lack of rate limiting (Brute force attacks).
- Insecure password recovery mechanisms.
- **Session Fixation**: Forcing a user to use a known session ID.
- Lack of session timeout.

Summary

- Authentication verifies identity; Authorization verifies permissions.
- Sessions maintain state over stateless HTTP using unpredictable IDs.
- Cookies must be protected with Secure, HttpOnly, and SameSite flags.
- JWTs offer a stateless alternative but require careful implementation.

SSL and TLS (Part 1)

- Concepts, History, and Architecture
- **Unit 4**: Web and Application Security
- Lecture 30

Agenda

- Need for Secure Communication
- **History**: SSL to TLS
- What SSL/TLS Provides
- Where SSL/TLS Operates in the OSI Model
- Cryptography Review for TLS

The Need for Secure Communication

- Internet traffic is routed through multiple untrusted nodes (ISPs, public Wi-Fi).
- Plaintext protocols (HTTP, FTP, Telnet) can be intercepted (Packet Sniffing).
- Attackers can read sensitive data (passwords, credit cards) or modify content (Man-in-the-Middle).
- Encryption in transit is essential for e-commerce and privacy.

Evolution: From SSL to TLS

- **SSL 1.0, 2.0, 3.0**: Developed by Netscape in the 1990s. (All deprecated due to severe flaws).
- **TLS 1.0 (1999)**: The standardized upgrade to SSL 3.0.
- **TLS 1.1, TLS 1.2 (2008)**: Major cryptographic improvements.
- **TLS 1.3 (2018)**: Modern standard; faster handshakes, removed outdated crypto.

What SSL/TLS Provides (The CIA Triad)

- **Confidentiality**: Encrypts data so only the intended recipient can read it.
- **Integrity**: Ensures data has not been altered in transit using Message Authentication Codes (MACs).
- **Authentication**: Verifies the identity of the server (and optionally the client) using digital certificates.

SSL/TLS in the Protocol Stack

- Operates between the Transport Layer (TCP) and the Application Layer (HTTP).
- **Application Layer**: HTTP, SMTP, FTP.
- **Security Layer**: TLS.
- **Transport Layer**: TCP.
- HTTP over TLS = HTTPS.

Cryptography Review for TLS

- TLS uses a hybrid cryptographic approach.
- **Asymmetric Cryptography (RSA, ECC)**: Used for the initial handshake and key exchange (slow but solves key distribution).
- **Symmetric Cryptography (AES, ChaCha20)**: Used for bulk data encryption after the handshake (very fast).
- **Hash Functions (SHA-256)**: Used for data integrity and digital signatures.

The TLS Record Protocol

- The protocol responsible for securing application data.
- Takes application data, fragments it into manageable blocks.
- Optionally compresses data.
- Applies a MAC for integrity and encrypts the block.
- Adds a TLS header and transmits over TCP.

Cipher Suites

- A cipher suite is a negotiated set of algorithms used for a TLS connection.
- **Format example**: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- **Components**: Key Exchange (ECDHE), Authentication (RSA), Bulk Encryption (AES_256_GCM), MAC/Hash (SHA384).

Summary

- TLS has replaced SSL as the standard protocol for secure web communication.
- HTTPS is simply HTTP transmitted over a TLS-secured TCP connection.
- TLS provides Confidentiality (Encryption), Integrity (MACs), and Authentication (Certificates).
- It utilizes a hybrid cryptographic model for efficiency.

SSL and TLS (Part 2)

- The TLS Handshake and Digital Certificates
- **Unit 4**: Web and Application Security
- Lecture 31

Agenda

- The TLS Handshake Protocol
- Step-by-Step Handshake Breakdown
- What is a Digital Certificate?
- Public Key Infrastructure (PKI)
- Certificate Authorities (CAs)
- TLS 1.3 Improvements

The TLS Handshake Protocol

- Occurs before any application data (HTTP) is transmitted.
- **Goals of the Handshake**:
- 1. Agree on the TLS version and Cipher Suite.
- 2. Authenticate the server (using its certificate).
- 3. Exchange and generate symmetric session keys.

TLS Handshake Steps (TLS 1.2) - Phase 1

- **1. Client Hello**: Client sends supported TLS versions, cipher suites, and a random byte string.
- **2. Server Hello**: Server responds with chosen TLS version, chosen cipher suite, and its own random byte string.
- At this point, algorithms are agreed upon.

TLS Handshake Steps (TLS 1.2) - Phase 2

- **3. Server Certificate**: Server sends its digital certificate containing its public key.
- **4. Server Hello Done**: Server indicates it has finished its part of the negotiation.
- **5. Client Key Exchange**: Client verifies the certificate, generates a 'Pre-Master Secret', encrypts it with the server's public key, and sends it.

TLS Handshake Steps (TLS 1.2) - Phase 3

- 6. Both sides independently calculate the 'Master Secret' (symmetric key) using the random strings and Pre-Master Secret.
- 7. Client sends 'Finished' message encrypted with the new symmetric key.
- 8. Server sends 'Finished' message encrypted with the new symmetric key.
- 9. Secure bulk communication begins!

Digital Certificates (X.509)

- An electronic document used to prove ownership of a public key.
- Binds a public key to an identity (Domain Name, Organization).
- **Contains**: Subject (e.g., google.com), Public Key, Expiration Date, Issuer (CA).
- It is digitally signed by a trusted third party (Certificate Authority).

Public Key Infrastructure (PKI)

- The ecosystem of roles, policies, and software that manage digital certificates.
- **Root Certificate Authorities (CAs)**: Highly trusted entities pre-installed in your browser/OS (e.g., DigiCert, Let's Encrypt).
- **Intermediate CAs**: Subordinate authorities that issue certificates on behalf of the Root CA.
- **Chain of Trust**: Browsers verify the server cert's signature up the chain to a trusted Root CA.

TLS 1.3 Improvements

- **Faster**: Reduces the handshake from 2 round trips to 1 round trip (0-RTT for resumed connections).
- **More Secure**: Removed outdated/weak algorithms (MD5, SHA-1, DES).
- **Mandates Perfect Forward Secrecy (PFS)**: Uses Ephemeral Diffie-Hellman keys so past traffic can't be decrypted if the server's long-term private key is compromised later.

Summary

- The TLS Handshake negotiates algorithms, authenticates the server, and establishes a symmetric session key.
- Digital certificates, signed by trusted CAs, prevent Man-in-the-Middle attacks.
- Browsers rely on a PKI Chain of Trust to validate certificates.
- TLS 1.3 improves performance and security significantly over TLS 1.2.

OWASP Top 10 (Part 1)

- Introduction and Key Vulnerabilities
- **Unit 4**: Web and Application Security
- Lecture 32

Agenda

- What is OWASP?
- The Top 10 List (Overview)
- **A01**: Broken Access Control
- **A02**: Cryptographic Failures
- **A03**: Injection

What is OWASP?

- Open Worldwide Application Security Project.
- A nonprofit foundation that works to improve the security of software.
- Provides unbiased, practical, cost-effective information.
- Best known for the 'OWASP Top 10' awareness document.

The OWASP Top 10 (2021 Overview)

- Updated every few years based on data from thousands of applications.
- Represents a broad consensus about the most critical security risks.
- Shifted towards data-driven categories rather than specific vulnerabilities.
- Companies often mandate that their apps must be 'OWASP Top 10 Compliant'.

A01: Broken Access Control

- The #1 vulnerability in the 2021 list.
- Occurs when users can act outside of their intended permissions.
- **Examples**: Bypassing access control checks by modifying the URL (e.g., /user/123 to /user/124).
- Elevation of privilege (a standard user acting as an admin).

Mitigating Broken Access Control

- Implement Access Control Mechanisms once and re-use them throughout the application.
- Deny by default.
- Enforce record ownership (user A cannot read user B's data).
- Log access control failures and alert admins.

A02: Cryptographic Failures

- Previously known as Sensitive Data Exposure.
- Focuses on failures related to cryptography that lead to data exposure.
- **Examples**: Storing passwords in plaintext or using weak hashes (MD5, SHA1).
- Transmitting sensitive data (PII, credit cards) over unencrypted HTTP.

Mitigating Cryptographic Failures

- Classify data and identify what is sensitive.
- Encrypt all sensitive data at rest using strong algorithms (AES-256).
- Use strong, salted, adaptive hashing functions for passwords (Argon2, bcrypt).
- Ensure TLS is enforced for all network traffic.

A03: Injection

- Occurs when untrusted data is sent to an interpreter as part of a command or query.
- The attacker's hostile data tricks the interpreter into executing unintended commands.
- **Most common form**: SQL Injection (SQLi).
- Also includes OS Command Injection, LDAP injection, and NoSQL injection.

Mitigating Injection

- **The primary defense**: Use Parameterized Queries (Prepared Statements).
- Parameterized queries treat user input strictly as data, never as executable code.
- Use Object Relational Mapping Tools (ORMs) which often use parameterized queries by default.
- Positive or 'whitelist' server-side input validation.

Summary

- The OWASP Top 10 is the leading awareness document for web app security.
- A01 Broken Access Control emphasizes strict, server-side permission enforcement.
- A02 Cryptographic Failures highlights the need for strong encryption and hashing.
- A03 Injection demonstrates the danger of mixing untrusted data with code.

OWASP Top 10 (Part 2)

- Design, Misconfiguration, and Logging
- **Unit 4**: Web and Application Security
- Lecture 33

Agenda

- **A04**: Insecure Design
- **A05**: Security Misconfiguration
- **A06**: Vulnerable and Outdated Components
- **A07**: Identification and Authentication Failures
- A08, A09, A10 Overview

A04: Insecure Design

- A new category focusing on risks related to design and architectural flaws.
- Differentiates between flawed design and flawed implementation.
- A secure implementation of a flawed design is still insecure.
- **Example**: An e-commerce app without bot protection allows scalpers to buy all inventory.

A05: Security Misconfiguration

- The most common vulnerability type in modern cloud environments.
- **Examples**: Default passwords left unchanged, open cloud storage buckets (AWS S3).
- Unnecessary features enabled (unnecessary ports, services, pages).
- Detailed error messages exposing stack traces to users.

A06: Vulnerable and Outdated Components

- Modern apps are built on open-source libraries and frameworks (npm, PyPI).
- Using components with known vulnerabilities exposes the app.
- **Examples**: Equifax breach was caused by an outdated Apache Struts component.
- You must track your inventory of dependencies (SBOM).

A07: Identification and Authentication Failures

- Weaknesses in how user identity is verified.
- **Examples**: Allowing automated brute force or credential stuffing attacks.
- Permitting weak or easily guessable passwords.
- Ineffective multi-factor authentication (MFA) implementations.

A08: Software and Data Integrity Failures

- Relates to code and infrastructure that does not protect against integrity violations.
- **Example**: CI/CD pipelines pulling malicious code from untrusted repositories.
- **Example**: Deserialization of untrusted data leading to Remote Code Execution (RCE).
- **Example**: Unsigned firmware updates.

A09: Security Logging and Monitoring Failures

- Without logging and monitoring, breaches cannot be detected.
- Most successful attacks take an average of 200 days to detect.
- **Failures include**: Not logging failed logins, logs not stored securely.
- Without alerts, attackers can probe the system indefinitely.

A10: Server-Side Request Forgery (SSRF)

- Occurs when a web application fetches a remote resource without validating the user-supplied URL.
- Allows an attacker to force the server to make requests to internal, firewalled systems.
- **Example**: Fetching cloud metadata from an AWS instance via SSRF.
- Becoming more common due to modern architectures (webhooks, URL fetching).

Summary

- Insecure Design emphasizes threat modeling early in the SDLC.
- Misconfigurations and outdated components are low-hanging fruit for attackers.
- Robust authentication, integrity checks, and comprehensive logging are mandatory.
- SSRF highlights the danger of apps making outward requests on behalf of users.

SQL Injection (SQLi)

- Understanding the Mechanics and Defenses
- **Unit 4**: Web and Application Security
- Lecture 34

Agenda

- What is SQL Injection?
- How SQLi Works
- Types of SQL Injection
- Impact of a Successful SQLi
- Defending against SQLi

What is SQL Injection?

- A code injection technique that attacks data-driven applications.
- Malicious SQL statements are inserted into entry fields for execution.
- Allows attackers to bypass authentication, access, modify, or delete database data.
- Caused by dynamically concatenating user input directly into SQL queries.

How SQLi Works: The Flawed Code

- **Example Query**: SELECT * FROM users WHERE username = "" + userInput + "';
- **If userInput is**: admin
- **Result**: SELECT * FROM users WHERE username = 'admin';
- **If userInput is**: admin' OR '1'='1
- **Result**: SELECT * FROM users WHERE username = 'admin' OR '1'='1';

Types of SQL Injection

- **In-band SQLi (Classic)**: Data is extracted using the same channel (e.g., results displayed on the webpage).
- **Error-based SQLi**: Attacker uses database error messages to gather information.
- **Union-based SQLi**: Uses the UNION operator to combine results from multiple tables.
- **Inferential (Blind) SQLi**: No data is returned; attacker infers data by asking true/false questions (Boolean-based or Time-based).

The Impact of SQL Injection

- **Confidentiality**: Reading sensitive data (passwords, credit cards, PII).
- **Integrity**: Modifying or deleting data (dropping tables, altering balances).
- **Authentication Bypass**: Logging in as administrator without a password.
- **Remote Code Execution**: In some cases, interacting with the underlying OS file system or executing shell commands.

Defending against SQLi: Parameterized Queries

- Also known as Prepared Statements.
- The absolute best defense against SQLi.
- The SQL structure is sent to the database first, and user inputs are sent later as parameters.
- The database treats the parameters strictly as literal values, never as executable code.
- **Example**: PreparedStatement pstmt = con.prepareStatement("SELECT * FROM users WHERE username = ?");

Defending against SQLi: Other Methods

- Use Object Relational Mapping (ORM) tools (e.g., Entity Framework, Hibernate), which use parameterized queries automatically.
- Stored Procedures (if implemented securely without dynamic SQL).
- Input Validation / Escaping (Least preferred, prone to bypass).
- **Principle of Least Privilege**: The database user should only have access to necessary tables, preventing DROP TABLE attacks.

SQLMap: The Attacker's Tool

- SQLMap is a popular open-source penetration testing tool.
- It automates the process of detecting and exploiting SQL injection flaws.
- Can automatically dump database tables, extract hashes, and even gain OS shells.
- Used by both attackers and ethical hackers.

Summary

- SQL Injection results from combining code and untrusted data via string concatenation.
- It can lead to full database compromise and data breaches.
- Attackers use various techniques, including Error, Union, and Blind SQLi.
- **Prevention is straightforward**: Always use Parameterized Queries / Prepared Statements.

XSS and CSRF

- Cross-Site Scripting and Request Forgery
- **Unit 4**: Web and Application Security
- Lecture 35

Agenda

- What is Cross-Site Scripting (XSS)?
- Types of XSS (Stored, Reflected, DOM)
- Impact of XSS
- Defending against XSS
- What is Cross-Site Request Forgery (CSRF)?
- Defending against CSRF

What is Cross-Site Scripting (XSS)?

- A vulnerability where an attacker injects malicious scripts into trusted websites.
- The victim's browser executes the script, believing it came from the trusted server.
- Payloads are usually written in JavaScript.
- Targets the user, not the application server directly.

Types of XSS: Stored XSS

- Also known as Persistent XSS.
- The injected script is permanently stored on the target server (e.g., in a database via a forum post).
- Every time a user visits the affected page, the malicious script is served and executed.
- Highest impact due to broad reach.

Types of XSS: Reflected & DOM-based

- **Reflected XSS**: The payload is part of the URL/request and is 'reflected' back by the server immediately (e.g., search results). Requires social engineering to trick the user into clicking the link.
- **DOM-based XSS**: The vulnerability exists purely in the client-side code. The server never sees the payload.
- The payload alters the Document Object Model in the browser.

Impact of XSS

- **Session Hijacking**: Stealing session cookies using document.cookie.
- **Keylogging**: Recording keystrokes on the infected page.
- **Phishing**: Modifying the DOM to present a fake login form.
- **Actions on Behalf of User**: Forcing the browser to make requests while logged in.

Defending against XSS

- **Context-Aware Output Encoding**: Convert special characters (¡, ¿) into safe HTML entities before rendering them in the browser.
- **Input Validation**: Reject data that doesn't conform to expected formats.
- **Content Security Policy (CSP)**: An HTTP header that restricts where scripts can be loaded and executed from.
- Use the HttpOnly flag on cookies to prevent theft.

What is Cross-Site Request Forgery (CSRF)?

- An attack that forces an authenticated user to execute unwanted actions.
- Exploits the fact that browsers automatically send cookies (like session IDs) with cross-origin requests.
- **Example**: An attacker site sends a hidden POST request to bank.com/transfer.
- Since the user is logged into bank.com, the browser includes their session cookie, and the bank executes the transfer.

Defending against CSRF

- **Anti-CSRF Tokens**: Unpredictable, unique tokens generated by the server, embedded in forms, and verified on submission.
- **SameSite Cookie Attribute**: Tells the browser not to send the session cookie in cross-origin requests (SameSite=Lax or Strict).
- Require re-authentication for sensitive actions (e.g., asking for a password before changing an email address).

Summary

- XSS allows attackers to execute malicious JavaScript in a victim's browser, leading to session theft.
- Defend XSS via Output Encoding and Content Security Policy (CSP).
- CSRF tricks the browser into issuing unauthorized commands using ambient credentials.
- Defend CSRF using Anti-CSRF tokens and SameSite cookie attributes.

Secure Software Development Life Cycle (SSDLC)

- Introduction and Threat Modeling
- **Unit 4**: Web and Application Security
- Lecture 36

Agenda

- What is SSDLC?
- The 'Shift Left' Concept
- Phases of the SSDLC
- **Phase 1**: Requirements & Risk Assessment
- **Phase 2**: Secure Design & Threat Modeling
- STRIDE Methodology

The Traditional vs Secure SDLC

- **Traditional SDLC**: Security testing happens right before deployment (Penetration Testing phase).
- **Result**: Fixing bugs late in the cycle is extremely expensive and delays launches.
- **Secure SDLC (SSDLC)**: Integrates security activities into every single phase (Requirements, Design, Code, Test).
- **Result**: Vulnerabilities are found and fixed early, saving time and money.

The 'Shift Left' Concept

- A movement in software development to perform testing and security checks earlier in the lifecycle.
- Moving tasks to the 'left' on the project timeline.
- Empowers developers with tools in their IDEs and CI/CD pipelines.
- Reduces the bottleneck of relying solely on a small security team at the end.

Phases of the SSDLC

- **1. Requirements**: Security requirements, risk assessment.
- **2. Design**: Threat modeling, secure architecture.
- **3. Development (Coding)**: Secure coding guidelines, Static Analysis (SAST).
- **4. Testing**: Dynamic Analysis (DAST), Penetration testing.
- **5. Deployment & Maintenance**: Incident response, monitoring, patching.

Phase 1: Security Requirements

- Define what security means for this specific application.
- Identify regulatory and compliance needs (e.g., GDPR, HIPAA, PCI-DSS).
- Create 'Abuse Cases' alongside 'Use Cases' (e.g., 'As an attacker, I want to bypass login').
- Define data classification (what data is sensitive).

Phase 2: Secure Design & Threat Modeling

- Threat Modeling is the process of identifying, understanding, and communicating potential threats to an application.
- **Ask 4 Questions**:
- 1. What are we building?
- 2. What can go wrong?
- 3. What are we going to do about it?
- 4. Did we do a good job?

The STRIDE Methodology

- Developed by Microsoft, used during Threat Modeling to categorize threats.
- S - Spoofing (Identity violation)
- T - Tampering (Integrity violation)
- R - Repudiation (Non-repudiation violation)
- I - Information Disclosure (Confidentiality violation)
- D - Denial of Service (Availability violation)
- E - Elevation of Privilege (Authorization violation)

Applying STRIDE

- Draw a Data Flow Diagram (DFD) showing external entities, processes, and data stores.
- Trace data as it moves through the system.
- Apply the STRIDE categories to every boundary and data flow.
- Document mitigations for every identified threat (e.g., use TLS to mitigate Tampering in transit).

Summary

- SSDLC integrates security into every phase, shifting security activities 'left'.
- Fixing bugs early in the design phase is vastly cheaper than fixing them in production.
- Threat Modeling is a critical design-phase activity.
- The STRIDE methodology helps systematically identify threats to the architecture.

SSDLC (Part 2)

- Secure Coding, Testing, and Deployment
- **Unit 4**: Web and Application Security
- Lecture 37

Agenda

- **Phase 3**: Secure Coding
- **Phase 4**: Security Testing (SAST vs DAST)
- Software Composition Analysis (SCA)
- **Phase 5**: Secure Deployment (DevSecOps)
- Incident Response
- Unit 4 Review

Phase 3: Secure Coding Practices

- Developers must adhere to secure coding standards (e.g., OWASP Secure Coding Practices).
- Conduct regular Peer Code Reviews.
- Avoid banned APIs and vulnerable functions.
- Implement the mitigations identified during Threat Modeling.

Phase 4: Security Testing Tools

- Security testing cannot rely on manual review alone; automation is required.
- **Static Application Security Testing (SAST)**: 'White-box' testing.
- **Dynamic Application Security Testing (DAST)**: 'Black-box' testing.
- **Software Composition Analysis (SCA)**: Dependency checking.

SAST vs DAST

- SAST scans the source code at rest. Finds issues like hardcoded passwords, dangerous function calls. Very fast, but high false positive rate.
- DAST scans the running application from the outside. Finds issues like misconfigurations, authentication bypasses. Slower, but lower false positive rate.
- Interactive Application Security Testing (IAST) combines aspects of both.

Software Composition Analysis (SCA)

- Analyzes the application's third-party dependencies.
- Checks libraries against databases of known vulnerabilities (e.g., CVE databases).
- Generates a Software Bill of Materials (SBOM).
- Addresses OWASP A06 (Vulnerable and Outdated Components).

Phase 5: Secure Deployment (DevSecOps)

- DevSecOps integrates security directly into CI/CD pipelines.
- Security checks (SAST, SCA, linting) are run automatically on every commit.
- If a critical vulnerability is found, the build fails and deployment is halted.
- Infrastructure as Code (IaC) ensures servers are deployed in a hardened state.

Maintenance and Incident Response

- No system is 100% secure. You must plan for a breach.
- Implement continuous monitoring, logging, and alerting (SIEM).
- **Establish an Incident Response (IR) plan**: Preparation, Identification, Containment, Eradication, Recovery, Lessons Learned.
- Run Bug Bounty programs to incentivize ethical hackers to find flaws.

Unit 4 Review

- Web Architecture & HTTP Fundamentals.
- Authentication, Authorization, and Sessions.
- SSL/TLS Cryptography and Handshakes.
- OWASP Top 10 Web Vulnerabilities (SQLi, XSS, CSRF).
- Secure Software Development Life Cycle (SSDLC).

Summary

- Secure coding requires established standards and peer review.
- SAST, DAST, and SCA provide automated security testing in DevSecOps pipelines.
- Deployment requires hardened infrastructure and continuous monitoring.
- Incident response planning ensures readiness for inevitable breaches.

Introduction to Unit 5: Cyber Laws and IT Standards

- Overview of Cyber Laws and IT Standards.
- Significance of legal frameworks in the digital age.
- **Topics covered**: Need for laws, IT Act 2000, ISO 27001, Cybercrimes, Ethical Hacking.

Definition

This is a key concept in Information Security.

What are Cyber Laws?

- Cyber law encompasses the legal and regulatory aspects of the Internet and computing.
- Also known as 'Information and Communications Technology Law' (ICT Law).
- It covers electronic commerce, digital signatures, data privacy, and cybercrime.
- Intersects with multiple legal domains including contract, intellectual property, and criminal law.

The Evolution of Cyber Laws

- Traditional laws were designed for the physical world (tangible assets, borders).
- Cyberspace is borderless, making jurisdiction complex.
- Rapid technological advancement (e-commerce, IoT) outpaced existing legal structures.
- UNUNCITRAL Model Law on Electronic Commerce (1996) paved the way for global e-commerce legislation.

Why Technical Security is Not Enough

- No system is 100% secure; breaches are inevitable.
- Technical controls (firewalls, encryption) prevent attacks but cannot prosecute attackers.
- Cyber laws provide the necessary deterrent through penal consequences.
- Laws enable organizations to claim damages and seek legal recourse post-incident.

Need for Cyber Laws: Electronic Commerce

- Legal recognition of electronic records and digital signatures.
- Ensuring non-repudiation in electronic transactions.
- Establishing validity of electronic contracts (e-contracts).
- Consumer protection in the digital marketplace.

Need for Cyber Laws: Data Protection and Privacy

- Explosion of data generation and collection by tech giants.
- Need to protect Personally Identifiable Information (PII).
- Defining data controller and data processor responsibilities.
- Mandating data breach notifications to users and authorities.

Need for Cyber Laws: Combating Cybercrime

- Defining and categorizing new types of crimes (e.g., ransomware, cryptojacking).
- Establishing procedures for digital forensics and evidence collection.
- Empowering law enforcement agencies (LEAs) with jurisdiction and authority.
- Facilitating international cooperation for cross-border cybercrimes.

Need for Cyber Laws: Intellectual Property Rights (IPR)

- Protecting digital content (software, media, literature) from piracy.
- Enforcing copyright, patents, and trademarks in cyberspace.
- Addressing issues of domain name disputes (cybersquatting).
- Digital Rights Management (DRM) circumvention penalties.

Challenges in Enforcing Cyber Laws

- **Jurisdictional issues**: Attackers and victims are often in different countries.
- Anonymity of the internet (VPNs, Tor networks, cryptocurrency).
- Volatility of digital evidence (can be easily deleted or altered).
- Lack of trained personnel in law enforcement and the judiciary.

Global Cyber Law Frameworks

- GDPR (General Data Protection Regulation) - European Union.
- CCPA (California Consumer Privacy Act) - United States.
- HIPAA (Health Insurance Portability and Accountability Act) - Healthcare US.
- Budapest Convention on Cybercrime - International treaty.

The Role of Compliance

- Organizations must comply with relevant cyber laws to operate legally.
- Compliance involves implementing specific security controls and policies.
- Non-compliance results in hefty fines, legal action, and reputational damage.
- Security is an ongoing process of aligning technical controls with legal mandates.

Introduction to the IT Act, 2000

- Enacted on 9th June 2000, it is the primary cyber law in India.
- Based on the UNCITRAL Model Law on Electronic Commerce.
- **Objective**: Provide legal recognition for transactions carried out by means of electronic data interchange (EDI).
- Facilitate electronic filing of documents with Government agencies.

Definition

This is a key concept in Information Security.

Scope and Applicability

- Extends to the whole of India.
- Applies also to any offence or contravention committed outside India by any person (irrespective of nationality).
- **Condition**: The act must involve a computer, computer system, or computer network located in India.
- **Exceptions**: Negotiable instruments, Power of Attorney, Trust, Will, contracts for sale of immovable property.

Legal Recognition of Electronic Records

- **Section 4**: Where any law requires information to be in writing, electronic records fulfill that requirement.
- Ensures electronic records are treated on par with paper documents.
- **Condition**: Information must be accessible for subsequent reference.
- Paved the way for e-governance and paperless offices.

Legal Recognition of Digital Signatures

- **Section 5**: Where any law requires a signature, a digital signature satisfies the requirement.
- Uses asymmetric cryptosystem and hash function.
- Ensures authentication, integrity, and non-repudiation.
- Regulated by the Controller of Certifying Authorities (CCA).

Electronic Governance (E-Governance)

- Sections 6-10 deal with the use of electronic records and digital signatures in Government and its agencies.
- Filing of any form, application, or document with government offices.
- Issue or grant of any license, permit, or sanction.
- Receipt or payment of money in a particular manner.

Controller of Certifying Authorities (CCA)

- The CCA is appointed by the Central Government.
- Licenses and regulates the working of Certifying Authorities (CAs).
- CAs issue Digital Signature Certificates (DSC) to users.
- Establishes a hierarchy and trust model for public key infrastructure (PKI) in India.

Certifying Authorities (CA)

- Entities licensed to issue Digital Signature Certificates.
- Responsible for verifying the identity of the applicant before issuing a DSC.
- Must maintain strict physical and logical security over their operations.
- **Examples in India**: e-Mudhra, NSDL, Safescrypt.

Offences and Penalties (Overview)

- Chapter IX outlines penalties for contraventions (civil wrongs).
- Chapter XI defines cybercrimes (criminal offences).
- Differentiates between unauthorized access resulting in damage (civil) vs. criminal intent.
- Penalties can range from fines to imprisonment.

Section 43: Penalty for Damage to Computer System

- Unauthorized access, downloading, or copying data.
- Introducing computer contaminants or viruses.
- Damaging or causing disruption to a computer network.
- Denial of Service (DoS) attacks.
- **Penalty**: Compensation by way of damages to the person affected.

Section 66: Computer Related Offences

- If any person, dishonestly or fraudulently, does any act referred to in Section 43.
- Transforms a civil contravention into a criminal offence due to mens rea (criminal intent).
- **Punishment**: Imprisonment up to 3 years or fine up to Rs. 5 Lakh, or both.

The Need for the IT Amendment Act, 2008

- Technology evolved rapidly between 2000 and 2008 (proliferation of mobile phones, social media).
- New forms of cybercrimes emerged (phishing, identity theft, cyber terrorism).
- Need to address data privacy and security practices of corporations.
- Amendment was passed in Dec 2008 and came into effect in Oct 2009.

Key Additions in the 2008 Amendment

- Introduction of 'Electronic Signature' as a broader term encompassing digital signatures.
- **Addition of new cybercrimes**: Identity theft, cheating by personation, cyber terrorism.
- **Section 43A**: Compensation for failure to protect data.
- **Section 69**: Expanded powers for interception and decryption.
- **Section 79**: Safe harbor provisions for intermediaries.

Section 43A: Corporate Liability for Data Protection

- Applies to 'Body Corporates' handling 'Sensitive Personal Data or Information' (SPDI).
- Mandates implementing 'reasonable security practices and procedures'.
- If negligence causes wrongful loss or gain, the company must pay compensation.
- Led to the IT (Reasonable Security Practices and Procedures and Sensitive Personal Data or Information) Rules, 2011.

Section 66C & 66D: Identity Theft and Personation

- **Section 66C**: Identity Theft - Fraudulent use of electronic signature, password, or unique identification feature.
- **Section 66D**: Cheating by personation by using a computer resource.
- Covers phishing attacks, fake social media profiles, and credential stuffing.
- **Punishment**: Imprisonment up to 3 years and fine up to Rs. 1 Lakh.

Section 66E: Violation of Privacy

- Capturing, publishing, or transmitting the image of a private area of any person without consent.
- Addresses issues of hidden cameras, non-consensual sharing of intimate images (revenge porn).
- **Punishment**: Imprisonment up to 3 years or fine up to Rs. 2 Lakh, or both.

Section 66F: Cyber Terrorism

- Acts intent on threatening the unity, integrity, security, or sovereignty of India.
- Denying access to authorized personnel to computer resources.
- Accessing protected systems to obtain classified information.
- **Punishment**: Life imprisonment.

Section 67: Obscenity in Electronic Form

- Publishing or transmitting material which is lascivious or appeals to the prurient interest.
- **Section 67A**: Material containing sexually explicit act.
- **Section 67B**: Child pornography (creation, transmission, or browsing).
- Strict punishments, increasing with subsequent convictions.

Section 69: Interception and Decryption

- Empowers Central/State Government to intercept, monitor, or decrypt any information.
- **Conditions**: Sovereignty of India, security of the State, public order, preventing incitement to offence.
- Subscribers and intermediaries must assist; failure leads to up to 7 years imprisonment.

Section 69A: Blocking of Websites

- Power to issue directions for blocking for public access of any information through any computer resource.
- Used for blocking apps, websites, and specific social media accounts.
- Reasons similar to Sec 69 (National security, public order).
- The process must follow specific rules and a review committee.

Section 79: Intermediary Liability

- Intermediaries (ISPs, Social Media platforms, Search Engines) are not liable for third-party data.
- **Condition**: They observe 'due diligence' and act merely as a conduit.
- Must take down unlawful content upon receiving 'actual knowledge' (court order or government notice).
- Creates a 'safe harbor' for tech platforms to operate.

The Controversial Section 66A (Struck Down)

- Punished sending 'offensive' messages through communication services.
- Highly misused to arrest individuals for political comments on social media.
- Supreme Court struck it down in Shreya Singhal v. Union of India (2015).
- Ruled unconstitutional for violating freedom of speech (Article 19(1)(a)).

Introduction to ISO 27001

- International standard for managing information security.
- Provides a framework for establishing, implementing, maintaining, and continually improving an ISMS.
- Helps organizations manage the security of assets such as financial info, IP, and employee details.
- Based on a risk management approach.

Definition

This is a key concept in Information Security.

What is an ISMS?

- Information Security Management System (ISMS).
- A systematic approach to managing sensitive company information so that it remains secure.
- Encompasses people, processes, and IT systems.
- Holistic view of security, integrating it into corporate culture and business processes.

The PDCA Cycle (Deming Wheel)

- **Plan**: Establish ISMS policy, objectives, processes, and procedures relevant to managing risk.
- **Do**: Implement and operate the ISMS policy, controls, processes, and procedures.
- **Check**: Assess and measure process performance against policies, objectives, and report results.
- **Act**: Take corrective and preventive actions, based on the results of the internal ISMS audit.

Key Clauses of ISO 27001 (Clauses 4-10)

- **Clause 4**: Context of the organization (Internal/External factors).
- **Clause 5**: Leadership (Management commitment, policy).
- **Clause 6**: Planning (Risk assessment and treatment).
- **Clause 7**: Support (Resources, awareness, communication).
- **Clause 8**: Operation (Execution of risk treatment).
- **Clause 9**: Performance evaluation (Monitoring, internal audit).
- **Clause 10**: Improvement (Corrective action).

Risk Assessment Methodology

- Identify assets, threats, and vulnerabilities.
- Determine the likelihood and impact of risks materializing.
- Calculate risk levels and compare against risk acceptance criteria.
- Risk = Asset Value x Threat x Vulnerability (Conceptual).

Risk Treatment Options

- **Modify (Mitigate)**: Apply security controls to reduce risk.
- **Retain (Accept)**: Accept the risk if it falls within the acceptance criteria.
- **Avoid**: Stop the activity that causes the risk.
- **Share (Transfer)**: Transfer the risk to a third party (e.g., insurance, outsourcing).

Statement of Applicability (SoA)

- A crucial document in ISO 27001 certification.
- Lists all the controls from Annex A.
- States whether each control is applicable or not.
- Provides justification for inclusion or exclusion.
- Links risk treatment with the chosen controls.

Annex A Controls (Overview)

- ISO 27001 Annex A provides a catalog of 93 security controls (in the 2022 revision).
- **Grouped into 4 themes**: Organizational, People, Physical, Technological.
- **Examples**: Access control, cryptography, physical security, incident management.
- Organizations select relevant controls based on their risk assessment.

The Certification Process

- **Stage 1 Audit**: Documentation review by an external auditor to ensure policies/procedures are in place.
- **Stage 2 Audit**: Field audit to verify that the ISMS is implemented and functioning as documented.
- Certification issued for 3 years.
- Requires annual surveillance audits to ensure ongoing compliance.

Benefits of ISO 27001 Certification

- Enhanced security posture and reduced risk of breaches.
- Competitive advantage and increased customer trust.
- Demonstrates compliance with legal and regulatory requirements (like GDPR).
- Operational efficiency through standardized processes.

Defining Cybercrime

- Any criminal activity that involves a computer, networked device, or a network.
- The computer may be the target of the crime or used as a tool to commit the crime.
- Often motivated by financial gain, espionage, or hacktivism.
- Evolves rapidly with technological advancements.

Classification of Cybercrimes

- Cybercrimes against Individuals.
- Cybercrimes against Property/Organizations.
- Cybercrimes against the Government/Nation.
- Cybercrimes against Society at large.

Cybercrimes Against Individuals

- **Phishing and Spear Phishing**: Deceptive emails to steal credentials.
- **Cyber Stalking & Harassment**: Repeated online harassment.
- **Identity Theft**: Stealing personal data to impersonate someone.
- Ransomware targeting personal computers.
- **Social Engineering**: Manipulating individuals into divulging confidential info.

Cybercrimes Against Organizations

- **Data Breaches**: Stealing customer data, intellectual property.
- **Denial of Service (DoS/DDoS)**: Overwhelming servers to disrupt services.
- **Ransomware (Enterprise level)**: Encrypting corporate networks for extortion.
- **Corporate Espionage**: Stealing trade secrets for competitive advantage.
- **Business Email Compromise (BEC)**: Tricking employees into transferring funds.

Cybercrimes Against the Government

- **Cyber Terrorism**: Attacks designed to cause widespread fear and disrupt critical infrastructure.
- **Cyber Warfare**: State-sponsored attacks against another nation's networks.
- **Website Defacement**: Hacking government sites to spread political messages.
- **Espionage**: Stealing classified national security information.

Focus: Ransomware

- Malware that encrypts the victim's files.
- The attacker demands a ransom (usually in cryptocurrency) for the decryption key.
- **Evolution**: 'Double Extortion' - attackers also threaten to leak the stolen data if the ransom isn't paid.
- Major impact on healthcare, municipal governments, and large enterprises.

Focus: Phishing and Social Engineering

- Phishing is the most common delivery vector for malware.
- Relies on creating a sense of urgency, fear, or curiosity.
- Vishing (Voice Phishing) and Smishing (SMS Phishing).
- Defense requires user awareness training and robust email filtering.

Focus: DDoS Attacks

- Distributed Denial of Service.
- Uses a botnet (network of compromised computers/IoT devices) to flood a target with traffic.
- **Objective**: Exhaust server resources, making the service unavailable to legitimate users.
- Used for extortion, hacktivism, or masking other attacks.

The Role of the Dark Web

- A hidden part of the internet accessible only via specialized software (like Tor).
- Provides anonymity for users and website operators.
- **Functions as a marketplace for cybercrime**: selling stolen credit cards, exploits, drugs, and offering 'Ransomware-as-a-Service'.
- Makes tracking and prosecuting cybercriminals extremely difficult.

Impact of Cybercrime

- **Financial Loss**: Trillions of dollars lost globally to theft, ransom, and remediation.
- **Reputational Damage**: Loss of customer trust following a data breach.
- **Operational Disruption**: Downtime causing loss of productivity and revenue.
- **Legal and Regulatory Penalties**: Fines for non-compliance (e.g., under GDPR or IT Act).

Learning from Case Studies

- Real-world incidents provide critical insights into attack vectors.
- Highlight the failures in security posture and human processes.
- Demonstrate the practical application (or lack) of cyber laws.
- Essential for understanding the evolving threat landscape.

Case Study 1: WannaCry Ransomware (2017)

- **Attack Type**: Global Ransomware worm.
- **Vector**: Exploited the 'EternalBlue' vulnerability in Windows SMB protocol.
- **Impact**: Infected ¿300,000 computers across 150 countries. Crippled the UK's NHS (National Health Service).
- **Root Cause**: Failure to apply a critical Microsoft security patch released months earlier.

WannaCry: Lessons Learned

- The necessity of robust Patch Management policies.
- The danger of using unsupported, end-of-life operating systems (e.g., Windows XP).
- The importance of network segmentation (to prevent lateral movement).
- The critical nature of offline backups for recovery.

Case Study 2: Stuxnet (2010)

- **Attack Type**: Highly sophisticated cyber warfare malware.
- **Target**: Iranian nuclear facilities (uranium enrichment centrifuges).
- **Vector**: Introduced via infected USB drives (jumping the air gap).
- **Mechanism**: Exploited multiple zero-day vulnerabilities to specifically target Siemens SCADA systems.

Stuxnet: Lessons Learned

- Air-gapped networks (not connected to the internet) are not immune to attacks.
- The devastating potential of zero-day exploits (vulnerabilities unknown to the vendor).
- The convergence of IT and OT (Operational Technology) security.
- Insider threats and supply chain vulnerabilities are potent attack vectors.

Case Study 3: Equifax Data Breach (2017)

- **Attack Type**: Massive data breach (PII of ~147 million people).
- **Vector**: Unpatched vulnerability in Apache Struts web application framework.
- **Impact**: Massive financial loss, reputational ruin, executive resignations, congressional hearings.
- **Root Cause**: Poor vulnerability management and failure to renew an internal security certificate.

Equifax: Lessons Learned

- **Asset management is crucial**: You can't patch what you don't know you have.
- **The need for defense-in-depth**: A single web app vulnerability should not lead to the compromise of the entire database.
- **Importance of monitoring and alerting**: The breach went undetected for 76 days.
- Incident Response planning is essential for managing the fallout.

Case Study 4: SolarWinds Supply Chain Attack (2020)

- **Attack Type**: Sophisticated Supply Chain Attack.
- **Vector**: Attackers compromised SolarWinds' build system and inserted a backdoor (SUNBURST) into official Orion software updates.
- **Impact**: ~18,000 organizations, including top US government agencies and Fortune 500 companies, downloaded the malicious update.
- **Significance**: Extremely stealthy; bypassed traditional perimeter defenses because the malware came from a trusted vendor.

SolarWinds: Lessons Learned

- Third-party risk management is a critical component of security.
- **The necessity of Zero Trust Architecture**: Assume the network is already breached and verify every action.
- Enhanced monitoring for anomalous behavior, even from trusted applications.
- The need for better software bill of materials (SBOM) and code signing verification.

Indian Context: Cosmos Bank Cyber Attack (2018)

- **Attack Type**: ATM Cash-out and SWIFT fraud.
- **Vector**: Attackers compromised the bank's ATM switch server.
- **Impact**: Rs.94 Crores siphoned off through thousands of fraudulent ATM withdrawals globally and SWIFT transfers.
- **Significance**: Highlighted vulnerabilities in banking infrastructure and integration with payment gateways.

Summary of Case Studies

- Most major breaches exploit basic vulnerabilities (unpatched software, weak passwords, phishing).
- Advanced attacks target supply chains or use zero-days, requiring advanced defense strategies.
- Legal compliance (ISO 27001, IT Act) requires organizations to implement controls that could prevent these incidents.
- Continuous learning and adaptation are non-negotiable in information security.

What is Ethical Hacking?

- The practice of intentionally probing computer systems, networks, or applications to find vulnerabilities.
- Conducted with the explicit permission of the system owner.
- **Objective**: To identify and patch security flaws before malicious hackers can exploit them.
- Also known as 'White Hat' hacking.

Hacker Classifications (The Hats)

- **White Hat**: Ethical hackers, authorized, defensive intent.
- **Black Hat**: Malicious hackers, unauthorized, criminal intent (financial gain, destruction).
- **Grey Hat**: Hacks without permission but usually reports vulnerabilities to the owner (often seeking a bounty); walks a legal gray line.
- **Script Kiddies**: Unskilled individuals using pre-written tools to launch attacks.

The Importance of Permission (Legality)

- Permission is the ONLY difference between ethical hacking and a cybercrime (under laws like the IT Act).
- Requires a signed 'Rules of Engagement' (RoE) document.
- **Scope definition is critical**: What is allowed to be tested, and what is off-limits?
- Protects the ethical hacker from legal prosecution.

Phases of Ethical Hacking: 1. Reconnaissance

- Also known as Footprinting.
- Gathering information about the target system or network.
- **Passive Recon**: Gathering info without direct interaction (e.g., WHOIS, social media, public records).
- **Active Recon**: Directly interacting with the target (e.g., DNS zone transfers, basic ping sweeps).

Phases of Ethical Hacking: 2. Scanning

- Using tools to identify live hosts, open ports, and services running on the target network.
- **Network Scanning**: Mapping the network topology (e.g., using Nmap).
- **Vulnerability Scanning**: Identifying known weaknesses in discovered services (e.g., using Nessus).
- Identifying the specific operating systems and application versions.

Phases of Ethical Hacking: 3. Gaining Access

- The actual exploitation phase.
- Using vulnerabilities identified during scanning to breach the system.
- **Methods**: Password cracking, buffer overflows, SQL injection, social engineering.
- **Goal**: Attain a level of access (user, administrator, or root).

Phases of Ethical Hacking: 4. Maintaining Access

- Ensuring persistent access to the compromised system.
- Installing backdoors, rootkits, or creating new administrative accounts.
- Allows the attacker to return later without re-exploiting the initial vulnerability.
- Often involves escalating privileges (moving from a standard user to root/admin).

Phases of Ethical Hacking: 5. Clearing Tracks

- Removing evidence of the hack to avoid detection.
- Deleting or altering log files (system logs, application logs).
- Hiding malware and backdoors.
- Ethical hackers simulate this phase to test the organization's incident detection and forensic capabilities.

Phases of Ethical Hacking: 6. Reporting

- The most critical phase for an ethical hacker.
- Documenting all findings, vulnerabilities exploited, and data accessed.
- Providing actionable remediation strategies (how to fix the flaws).
- The report is the deliverable the client pays for.

Key Skills of an Ethical Hacker

- Deep understanding of networking protocols (TCP/IP, DNS, HTTP).
- Proficiency in operating systems (especially Linux and Windows).
- Knowledge of programming and scripting (Python, Bash, PowerShell).
- Familiarity with security tools (Metasploit, Wireshark, Burp Suite).
- Strong ethics and trustworthiness.

What is Penetration Testing (Pen Testing)?

- A formal, authorized simulated cyberattack on a computer system.
- Performed to evaluate the security of the system and identify exploitable vulnerabilities.
- A subset of ethical hacking focused specifically on discovering and exploiting flaws.
- Required for compliance with standards like ISO 27001, PCI-DSS.

Vulnerability Assessment vs. Pen Testing (VAPT)

- **Vulnerability Assessment (VA)**: Automated scanning to identify and list known vulnerabilities. It stops at discovery.
- **Penetration Testing (PT)**: Involves manual exploitation to verify if the vulnerability can be used to breach the system.
- **VA asks**: 'What are our weaknesses?'
- **PT asks**: 'Can someone break in using these weaknesses?'

Types of Penetration Testing: Based on Knowledge

- **Black Box Testing**: Tester has zero knowledge of the internal network (simulates an external hacker).
- **White Box Testing**: Tester has full knowledge (source code, architecture diagrams). Simulates an insider threat or focuses on exhaustive testing.
- **Grey Box Testing**: Partial knowledge (e.g., standard user credentials). Simulates a compromised employee account.

Types of Penetration Testing: Based on Target

- **Network Pen Testing**: Targeting firewalls, routers, switches, and servers.
- **Web Application Pen Testing**: Testing web apps for SQLi, XSS, CSRF (OWASP Top 10).
- **Mobile App Pen Testing**: Assessing iOS/Android app security.
- **Social Engineering Testing**: Phishing employees to test human awareness.
- **Physical Pen Testing**: Attempting to bypass physical security (locks, badges).

Standard Methodologies

- Pen testing follows structured frameworks to ensure thoroughness and consistency.
- OSSTMM (Open Source Security Testing Methodology Manual).
- PTES (Penetration Testing Execution Standard).
- OWASP Testing Guide (specifically for web applications).
- NIST SP 800-115 (Technical Guide to Information Security Testing).

Essential Pen Testing Tools: Nmap

- Network Mapper (Nmap).
- The industry standard for network discovery and security auditing.
- Used for host discovery, port scanning, service enumeration, and OS fingerprinting.
- Command-line based, highly flexible and powerful.

Essential Pen Testing Tools: Metasploit Framework

- The world's most used penetration testing framework.
- Provides a massive database of exploits, payloads, and post-exploitation modules.
- Allows testers to easily launch attacks against known vulnerabilities.
- Automates the exploitation phase.

Essential Pen Testing Tools: Burp Suite

- The premier tool for Web Application Security Testing.
- Acts as an intercepting proxy between the browser and the web server.
- Allows testers to inspect and modify traffic (HTTP requests/responses) on the fly.
- Essential for finding vulnerabilities like SQL injection and XSS.

Essential Pen Testing Tools: Wireshark

- The world's foremost network protocol analyzer (Packet Sniffer).
- Captures and displays network traffic in microscopic detail.
- Used for troubleshooting, analysis, software and communications protocol development, and education.
- Can capture cleartext passwords traversing the network.

The Value of the Pen Test Report

- **Executive Summary**: High-level overview of risk for management.
- **Technical Details**: Step-by-step documentation of vulnerabilities and exploitation.
- **Risk Rating**: CVSS (Common Vulnerability Scoring System) scores for each finding.
- **Remediation Steps**: Concrete advice on how to patch the vulnerabilities.
- The report bridges the gap between technical flaws and business risk.