CHAPTER 1: BASICS OF INFORMATION SECURITY

1. Information and Information Security

Information: Processed, organized data that holds meaning and is useful for decision-making

Locations of Information:

- Personal devices (laptops, smartphones)
- Servers & data centers
- Cloud storage (Google Drive, AWS)
- Networks (wired/wireless)
- Removable media (USB, external drives)
- Printed documents
- IoT devices
- Human mind

Information Security (InfoSec): Practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording, or destruction of information

Key Terms:

- **Information Security**: Protection of all forms of information (digital or physical)
- **Computer Security**: Protection of computing devices (PCs, laptops, servers)
- **Network Security**: Securing data during transmission across networks
- **Cybersecurity**: Broader term encompassing all above, defending against cyberattacks

---

2. Why Information Security is Important

Five Key Reasons:

1. **Protecting sensitive information**: Personal, financial, trade secrets, government data
2. **Mitigating risk**: Prevent data breaches, DoS attacks, malicious activities
3. **Compliance with regulations**: Meet legal requirements (ISO 27001, GDPR)
4. **Protecting reputation**: Prevent damage from security breaches
5. **Ensuring business continuity**: Maintain access to systems during incidents

---

3. CIA Triad (Core Principles)

Confidentiality:

- Information not disclosed to unauthorized entities

- Example: Password compromise = confidentiality breach

Integrity:

- Maintaining accuracy and completeness of data
- Only authorized persons can edit data
- Example: Employee status updates must be accurate and authorized

Availability:

- Information must be accessible when needed
- Requires collaboration across teams (network ops, incident response)
- Example: DoS attacks hamper availability

Privacy: Related to confidentiality; user's right to control personal information with consent

---

4. Information Security in Various Domains

Computer Memory:

- Storage: Primary (RAM, ROM) and Secondary (HDD, SSD)
- Threats: Malware, buffer overflows, unauthorized access
- Security: Memory encryption, secure boot, access control, anti-malware

Cloud Computing:

- Models: IaaS, PaaS, SaaS
- Challenges: Multi-tenancy, data breaches, insider threats, limited user control
- Security: Encryption (at rest, in transit, in use), IAM, auditing, zero-trust architecture, ISO 27001/GDPR compliance

IoT (Internet of Things):

- Devices: Sensors, smart appliances, wearables with limited computing power
- Threats: Eavesdropping, weak authentication, firmware vulnerabilities, physical tampering
- Security: Lightweight encryption, device authentication, secure firmware updates, network segmentation

---

5. Pyramid of PAIN

Concept: Model by David J. Bianco showing how threat indicators impact attacker's ability to succeed. Higher levels = harder for attackers to change

Six Levels (bottom to top):

1. **Hash Values**:
    - Unique file fingerprints (MD5, SHA-1, SHA-256)
    - Easy for attackers to change by recompiling files
    - **Pain Level**: Trivial
2. **IP Addresses**:
    - Block malicious IPs (e.g., C&C servers)
    - Attackers can switch IPs using fast-flux/proxies
    - **Pain Level**: Low
3. **Domain Names**:
    - Block suspicious domains (phishing, malware distribution)
    - Harder to rotate than IPs; attackers may use DGAs
    - **Pain Level**: Moderate
4. **Network/Host Artifacts**:
    - System-level indicators (registry keys, filenames, user-agent strings)
    - Harder to evade as attackers rely on standard procedures
    - **Pain Level**: Moderate to High
5. **Tools**:
    - Hacking frameworks (Mimikatz, Cobalt Strike, RATs)
    - Blocking forces attackers to find/create new tools
    - **Pain Level**: High
6. **TTPs (Tactics, Techniques, Procedures)**:
    - Behavioral patterns and strategies (spear-phishing, lateral movement)
    - Hardest to change; behavior-based detection
    - **Pain Level**: Most Painful

Strategy: Focus on higher-level detection (tools and TTPs) for proactive defense

---

6. OSI Model (7 Layers)

Purpose: Standard framework for understanding network communication

Layer	Function	Examples
7. Application	Network services to applications	HTTP, FTP, Email
6. Presentation	Data translation, encryption, compression	SSL/TLS, JPEG
5. Session	Manages sessions between applications	NetBIOS, RPC
4. Transport	End-to-end communication, reliability	TCP, UDP
3. Network	Routing across networks	IP, ICMP
2. Data Link	Node-to-node transfer, error detection	MAC, Ethernet
1. Physical	Physical transmission of bits	Cables, switches

OSI Security Architecture (X.800):

- Standardized approach to security across OSI layers
- Defines security services, mechanisms, and threat mapping
- Enables layered protection

---

7. Aspects of Security

Three Fundamental Aspects:

1. **Security Attack**: Action compromising CIA
2. **Security Mechanism**: Tools to detect/prevent/recover (firewalls, encryption, IDS)
3. **Security Service**: Outcomes/goals supported by mechanisms (confidentiality, authentication, non-repudiation)

---

8. Security Attacks

8.1 Threat vs Attack

Aspect	Threat	Attack
Definition	Potential danger	Actual action exploiting vulnerability

Aspect	Threat	Attack
Nature	Theoretical/possible	Practical/real
Intent	May lack malicious intent	Deliberate and intentional
Example	Possibility of virus	Hacker deploying malware
Response	Risk assessment, prevention	Incident response, mitigation

8.2 Passive Attacks

Characteristics:

- Monitor/eavesdrop without altering data
- Hard to detect; prevention through encryption is key
- Do not affect system resources directly

Two Types:

1. **Release of Message Contents**:
    - Attacker reads message content (e.g., passwords, bank details)
    - Example: Darth reads Alice's unencrypted message to Bob
2. **Traffic Analysis**:
    - Attacker observes communication patterns (timing, frequency, length)
    - Even with encryption, behavioral patterns are deduced
    - Example: Darth analyzes when/how often Alice sends messages

Defense: Strong encryption, secure communication protocols

8.3 Active Attacks

Characteristics:

- Involve modification, disruption, or fabrication
- Easier to detect than passive attacks
- Directly affect integrity, authenticity, availability

Four Types:

1. **Masquerade Attack**:
    - Attacker pretends to be legitimate user

- Example: Darth sends message pretending to be Alice

2. **Replay Attack**:
    - Captured legitimate message is resent later
    - Example: Replay money transfer message
3. **Modification of Messages**:
    - Intercepted message is altered before forwarding
    - Example: "Meet at 5 PM" changed to "Meet at 9 PM"
4. **Denial of Service (DoS)**:
    - Overwhelming/blocking legitimate communications
    - Example: Flooding server to cause crash

Defense: Authentication, integrity checks, encryption, traffic control

8.4 Comparison Table

Feature	**Active Attack**	**Passive Attack**
Data Modification	Yes	No
Objective	Damage/manipulate	Gather information
Detection	Easier (system impact)	Hard (no activity)
System Impact	Disrupt operations	No direct impact
Threat Level	High	Medium
Prevention	Firewalls, IDS, encryption	Strong encryption

---

9. Security Services (X.800 Standard)

9.1 Five Core Services

1. Authentication:

- Verifying identity of communicating entity
- Methods: Passwords, biometrics, digital certificates, 2FA
- Example: Banking app login with password + OTP

2. Access Control:

- Restricting access based on identity and permissions
- Four components: Identification, Authentication, Authorization, Accounting
- Types: DAC (owner-defined), MAC (system-enforced), RBAC (role-based), ABAC (attribute-based)
- Example: Employee accesses only authorized applications

3. Data Confidentiality:

- Protecting from unauthorized disclosure
- Techniques: Encryption (SSL/TLS), access controls
- Example: Encrypted website data prevents interception

4. Data Integrity:

- Ensures data remains unaltered
- Techniques: Checksums, hash functions (SHA-256), digital signatures
- Example: Verifying software download hash

5. Non-Repudiation:

- Prevents denial of actions
- Techniques: Digital signatures, timestamping, audit logs
- Example: Digitally signed email proves sender identity

9.2 Additional Important Services

Availability:

- Reliable access to systems when needed
- Techniques: High availability (HA), redundancy, load balancing, failover, DDoS protection
- Example: Online banking backup servers

Privacy:

- User's right to control personal information
- Types: Individual, behavior, communication, data, thoughts, location, association
- Techniques: Strong passwords, MFA, encryption, software updates, firewalls

---

10. Real-World Examples

10.1 Online Banking System

Service	Implementation
Authentication	Username, password, 2FA
Access Control	User sees only own accounts
Confidentiality	HTTPS encryption
Integrity	Transaction validation, hash functions
Non-Repudiation	Activity logs, digital signatures
Availability	Backup servers, DDoS protection

10.2 E-Commerce Platform (Swiggy/Amazon)

Service	Implementation
Authentication	Email, password, OTP login
Authorization	Customers access own orders only
Availability	Cloud infrastructure, load balancers
Confidentiality	HTTPS, SSL for payment data
Integrity	Order details validated via checksums
Non-Repudiation	Digital receipts, transaction logs
Privacy	User consent, data control options

---

11. Security Mechanisms

Definition: Features designed to detect, prevent, or recover from security attacks

Two Types:

1. **Specific Security Mechanisms**:
    - Encipherment (encryption)
    - Access controls
    - Digital signatures
    - Data integrity checks
    - Authentication exchanges
2. **Pervasive Security Mechanisms**:
    - Trusted functionality
    - Security labels
    - Event detection
    - Audit trails
    - Security recovery methods

Foundation: Cryptographic techniques (encryption, digital signatures)

Relationship with Services: Mechanisms are tools that implement and enforce security services

- Example: Encryption (mechanism) supports Confidentiality (service)
- Example: Digital signatures (mechanism) support Integrity and Non-repudiation (services)

---

12. Attack Tree for Internet Banking Authentication

Concept: Structured diagram modeling attack paths on a system

Root Goal: Gain unauthorized access to banking account

Common Attack Vectors:

- Stealing credentials (phishing, keylogging)
- Bypassing 2FA
- Exploiting software vulnerabilities
- Session hijacking
- Man-in-the-middle (MitM) attacks
- Brute-force attacks

Defense Strategies:

- Strong encryption
- Multi-factor authentication

- Biometric verification
- User behavior analysis

---

13. Standards and Frameworks

X.800 (ITU-T):

- Comprehensive OSI security architecture
- Defines 5 core security services
- Provides layered security guidance

RFC 2828 (IETF):

- Standard glossary for internet security terms
- Provides common vocabulary
- Supports interoperability

Other Standards:

- ISO 27001: Information security management
- GDPR: Data protection regulation

CHAPTER 2: STORAGE SYSTEMS, RAID AND SECURITY MECHANISMS

1. Storage Systems Overview

Storage System: Combination of hardware and software designed to store, manage, and protect digital data

Components:

- Physical: Hard Disk Drives (HDDs), Solid-State Drives (SSDs)
- Logical: RAID configurations

Core Principles Supported: Availability, Integrity, Confidentiality

Requirements: Reliable data access, prevent data loss/tampering, restrict unauthorized access

---

2. Hard Disk Drives (HDDs)

Description: Mechanical devices using magnetic platters and read/write heads for storage

Advantages:

- High storage capacity
- Low cost

Disadvantages:

- Susceptible to mechanical failures
- Data corruption risks
- Security threats

Security Requirements: Access control, data encryption, dependability features

---

3. Disk Arrays

Definition: High-capacity storage system integrating multiple physical disk drives with a disk controller

Main Goals:

- Increase data throughput
- Ensure high availability
- Improve data integrity through backup and failover

3.1 Types of Disk Arrays

1. JBOD (Just a Bunch of Disks):

- Multiple disks connected without redundancy
- Each disk operates independently
- Low cost, but no fault tolerance

2. RAID Arrays:

- Different levels (0, 1, 5, 10, etc.)
- Balance redundancy, speed, data protection
- Prevent data loss and accelerate operations

3. NAS (Network Attached Storage):

- Connected to LAN
- Provides file-level access
- Ideal for file sharing, backups, collaborative environments
- Example: Synology DiskStation DS220

4. SAN (Storage Area Network):

- Connected via dedicated high-speed network
- Provides block-level access
- Used in data centers and enterprise environments

- Example: Dell EMC PowerStore SAN

---

4. RAID Levels

4.1 RAID 0 (Striping)

Method: Data divided into stripes and distributed across multiple disks in round-robin fashion

Performance: High (simultaneous read/write operations)

Redundancy: None

Risk: Single disk failure = complete data loss

4.2 RAID 1 (Mirroring)

Method: Data written identically to two different disks

Performance: Does not notably enhance performance

Redundancy: Excellent (exact copy on second disk)

Fault Tolerance: High (survives 1 disk failure)

4.3 RAID 10 (0+1) (Striping + Mirroring)

Method: Combines RAID 0 and RAID 1

Process: First stripes data, then mirrors those stripes

Minimum Disks: 4 (two for striping, two for mirroring)

Benefits: High speed + fault tolerance

Drawback: Resource-intensive

4.4 RAID 2 (Bit-Level Striping with Hamming Code ECC)

Method: Bit-level striping with dedicated disks for error correction codes (Hamming codes)

Features: Automatic bit error detection and correction

Requirements: Precise disk synchronization

Usage: Rarely used today (modern drives have built-in ECC)

4.5 RAID 3 (Byte-Level Striping with Dedicated Parity)

Method: Data divided at byte level across multiple disks with one parity disk

Example: "HELLO" split as H, E, L, L, O across different disks + parity byte

Fault Tolerance: Good (can reconstruct missing byte using parity)

Limitation: All disks must operate in sync; inefficient for multiple small I/O operations

4.6 RAID 4 (Block-Level Striping with Dedicated Parity)

Method: Block-level striping with dedicated parity disk

Read Operations: Fast and independent

Write Operations: Bottleneck (constant updating of single parity disk)

4.7 RAID 5 (Block-Level Striping with Distributed Parity)

Method: Block-level striping with parity distributed across all disks

Minimum Disks: 3

Fault Tolerance: Can tolerate 1 disk failure without data loss

Usage: Widely used in server environments

Benefits: Efficient disk space use + good fault tolerance

4.8 RAID 6 (Block-Level Striping with Dual Distributed Parity)

Method: Two parity blocks for each data set

Fault Tolerance: Can tolerate up to 2 disk failures

Trade-offs: Additional storage overhead, slightly reduced write performance

Use Case: Critical systems requiring higher fault tolerance

---

5. RAID Comparison Table

RAID Level	Cost	Performance	Fault Tolerance	Scalability
JBOD	Low	Low	None	Moderate
RAID 0	Low	High	None	Limited
RAID 1	Moderate	Moderate	High (1 disk)	Low
RAID 5	Moderate	Good	Moderate (1 disk)	Moderate
RAID 6	High	Moderate	High (2 disks)	Moderate
RAID 10	High	High	High (1+ disk)	Low-Moderate
NAS	Moderate	Moderate	Depends on RAID	High
SAN	Very High	Very High	Enterprise-level	Very High

6. NAS vs SAN

NAS (Network Attached Storage)

Connection: Connected to LAN
Access Type: File-level access
Use Cases: Shared file access, backups, home and enterprise
Cost: Moderate
User-friendliness: High

SAN (Storage Area Network)

Connection: Dedicated high-speed network
Access Type: Block-level access
Use Cases: Critical applications, virtualization, heavy databases
Cost: Very high
Scalability: Very high
Reliability: Enterprise-level

---

7. Dependability in Storage Systems

Definition: Reliability, availability, and maintainability of a storage system

Purpose: Ensures data remains consistently accessible and free from corruption despite component failure or attack

Critical Sectors: Banking, healthcare, government operations

---

8. Threats to Storage Systems

Three Major Threats:

1. Physical damage from hardware failure or environmental hazards
2. Malware infections
3. Unauthorized access or internal breaches

Defense Strategies: RAID, data backups, encryption, access control

---

9. Security Implications of Disk Arrays

Requirements: Access controls, encryption, data integrity mechanisms, secure erasure

Important Note: RAID protects against hardware failures but does NOT inherently safeguard against data theft or unauthorized access

Necessity: Multi-layered security mechanisms required

---

10. Security Mechanisms in Storage

Layered Approach: No single mechanism provides full protection; systems use multiple mechanisms often rooted in cryptographic techniques

10.1 Specific Security Mechanisms

- Encipherment (encryption)
- Digital signatures
- Access controls
- Data integrity checks
- Authentication exchanges
- Traffic padding
- Routing control
- Notarization

10.2 Pervasive Security Mechanisms

- Trusted functionality
- Security labels
- Event detection systems
- Security audit trails
- Recovery mechanisms

Purpose: Ensure stored data remains confidential, authentic, and available

---

11. Key Selection Factors

Choosing Disk Array Configuration:

- **RAID 0**: High speed, no fault tolerance (complete data loss if 1 disk fails)
- **RAID 1**: Excellent fault tolerance through mirroring, more expensive
- **RAID 5**: Balance between performance and reliability (single disk failure recovery)
- **SAN**: Enterprise-level performance and scalability, highest cost and complexity

Scalability: Easier in NAS and SAN; traditional RAID requires manual intervention for expansion

Decision Factors: Organization's priorities between speed, reliability, and budget

---

WEEK 2: CHAPTERS 3 & 4 - OS SECURITY FEATURES AND ANTIVIRUS

1. Operating System (OS) Overview

Operating System: System software functioning as interface between computer hardware and user

Modern OS Examples:

- Windows 11: TPM 2.0, Secure Boot
- Linux (Ubuntu 24.04 LTS): AppArmor, Wayland support
- macOS Sonoma: Secure Enclave
- Android 14: Enhanced privacy, security updates

Core Purpose: Manages hardware/software resources, enables efficient user-application interaction

---

2. Key Functions of OS

2.1 Process Management

- Ensures processes operate safely without interference
- Enforces access control to prevent unauthorized resource access
- Uses sandboxing, process authentication, privilege levels
- Secure inter-process communication (IPC)

2.2 Memory Management

- Each process accesses only its allocated memory
- Techniques: Virtual memory, memory isolation, address space protection
- Hardware features: Memory Protection Units (MPUs), paging
- Security practices: ASLR (Address Space Layout Randomization), stack canaries
- Prevents buffer overflows and code injection

2.3 File System Management

- Protects data from unauthorized access/modification/deletion
- Enforces file permissions, ownership rules, ACLs
- Features: Encryption, auditing, secure file deletion
- Prevents data breaches, privilege escalation, malware persistence

2.4 Device Management

- Secures hardware access (printers, USB, network interfaces)
- Device drivers with strict access controls
- I/O control mechanisms, driver signing, hardware abstraction layers
- Security: Device encryption, port disabling, audit logging

2.5 Security & Access Control

Authentication Methods: User IDs, passwords, biometrics, MFA

Access Control Models:

- DAC (Discretionary Access Control): Owner-defined
- MAC (Mandatory Access Control): System-enforced
- RBAC (Role-Based Access Control): Role-based
- ABAC (Attribute-Based Access Control): Attribute-based

Additional Mechanisms: Firewalls, encryption, audit logs, security policies

2.6 User Interface

- Secure access through GUI or CLI
- Enforces authentication before granting access
- Features: Login screens, screen locking, secure input handling, activity logging

2.7 Resource Allocation & System Performance

- Manages CPU, memory, I/O efficiently while preventing vulnerabilities
- Prevents DoS attacks
- Enforces quota limits
- Monitors resource usage for abnormal behavior

---

3. OS Security

Definition: Collection of measures protecting resources, processes, and data from unauthorized access and threats

Aims: Ensure confidentiality, integrity, and availability through access controls, authentication, process isolation

Mechanisms: Password policies, encryption, firewalls, role-based permissions, kernel protection

Importance: OS is core layer controlling all hardware/software/user interactions; compromise = full system access

---

4. Threats to OS

4.1 Program Threats

Trojan Horses: Appear legitimate but execute harmful actions
Logic Bombs: Trigger malicious effects when conditions met
Trapdoors (Backdoors): Intentional security bypasses

4.2 System Threats

- **Denial of Service (DoS)**: Overwhelm system resources
- **Resource Exhaustion**: Malicious processes consume CPU/memory
- **Data Breaches**: Unauthorized access to sensitive information
- **Network Intrusion**: Compromise communication channels
- **Rootkits**: Hide malicious activity by modifying core processes

---

5. Different Attacks on OS

Password Attack: Brute-force, dictionary attacks, phishing, keylogging to bypass login

Buffer Overflow: Writing excess data to memory buffer; can inject malicious code

Masquerade: Attacker pretends to be authorized user with stolen credentials

SQL Injection: Exploit input fields to insert malicious SQL code

Shrink Wrap Code: Exploits vulnerabilities in pre-packaged software with default settings

Misconfiguration Attacks: Exploit weak permissions, open ports, default passwords, unnecessary services

Man-in-the-Middle (MitM): Intercepts communication, eavesdrops or alters data

Distributed Denial of Service (DDoS): Multiple systems flood target with traffic

Zero-Day Attack: Targets unknown vulnerability before patch exists

---

6. Core OS Security Features

6.1 Authentication

- Verifies user identity
- Methods: Passwords, biometrics, multi-factor authentication
- Prevents unauthorized access, maintains accountability

6.2 Access Control

- Regulates permissions based on identity/role/attributes
- Enforces principle of least privilege
- Prevents unauthorized access

6.3 File Protection

- Safeguards files from unauthorized access/modification/deletion
- Uses permissions, encryption, access control mechanisms

6.4 Memory Protection

- Prevents processes from accessing other processes' memory
- Uses segmentation and paging
- Ensures system stability

6.5 Process Isolation

- Each process operates independently
- Separate memory spaces, restricted resource access
- Limits impact of crashes or malicious behavior

6.6 Secure Boot

- Ensures only trusted, digitally signed software runs at startup
- Verifies bootloaders and system files against trusted database
- Prevents malware, rootkits, unauthorized code

6.7 Logging and Auditing

- Records user actions, system errors, access attempts
- Reviews logs to verify compliance, investigate incidents
- Supports forensic analysis

6.8 Network Security

- Protects data from unauthorized access across networks
- Tools: Firewalls, IPS, encryption, access controls
- Monitors traffic, blocks malicious activity

---

7. Advanced OS Security Features

7.1 Sandbox

- Isolated environment for executing untrusted code
- Restricts access to files, memory, network
- Useful for detecting zero-day threats
- Example: Email security gateways, endpoint protection

7.2 Secure Boot (Detailed)

- Built into UEFI (Unified Extensible Firmware Interface)
- Verifies digital signatures during boot process
- Halts boot if components are tampered or unsigned

- Protects against rootkits and boot-level attacks

7.3 Kernel Protection

Techniques:

- **KASLR**: Randomizes kernel memory locations
- **Control Flow Integrity (CFI)**: Ensures trusted operation sequences
- **Windows PatchGuard**: Blocks unauthorized kernel modifications
- **Linux Lockdown Mode**: Limits kernel interactions
- **Android Kernel Verification**: Verifies integrity during boot

Impact: Fortifies against advanced cyber threats, prevents system corruption

7.4 Patch Management

Process: Identify, acquire, test, deploy software updates

Cycle:

1. Scan systems for missing updates
2. Prioritize by severity
3. Test in controlled environments
4. Deploy systematically

Examples: Microsoft Patch Tuesday, Adobe/Cisco security patches

---

8. OS-Specific Security Examples

8.1 Windows Security

- **Windows Defender Antivirus**: Real-time malware protection
- **BitLocker**: Full drive encryption using TPM
- **Windows Hello**: Biometric authentication (facial recognition, fingerprint)
- **UAC (User Account Control)**: Prompts for elevated privileges
- **Windows Firewall**: Monitors network traffic
- **VBS (Virtualization-Based Security)**: Isolates sensitive processes
- **Secure Boot**: Verifies digital signatures at startup
- **Exploit Protection & ASR**: Mitigates buffer overflows, script attacks
- **Windows Sandbox**: Lightweight virtual environment

8.2 Linux Security

- **SELinux**: Mandatory access controls (MAC) by NSA

- **AppArmor**: Profile-based resource access (Ubuntu, SUSE)
- **Kernel Lockdown Mode**: Restricts root-level kernel access
- **Iptables/Nftables**: Firewall utilities for packet filtering
- **Auditd**: Tracks system calls, logs security events
- **ClamAV**: Open-source antivirus
- **Chkrootkit & Rkhunter**: Rootkit scanners
- **LUKS**: Full-disk encryption
- **Fail2ban**: Bans IPs after brute-force attempts
- **Lynis**: Vulnerability and compliance auditing tool

8.3 macOS Security

- **FileVault 2**: Full disk encryption (XTS-AES 128)
- **Gatekeeper**: Checks developer signatures before app execution
- **XProtect**: Built-in malware scanner (YARA signatures)
- **SIP (System Integrity Protection)**: Prevents root modification of system files
- **Secure Enclave**: Hardware security module in Apple Silicon
- **App Sandbox**: Isolates apps from each other
- **macOS Firewall & Stealth Mode**: Controls connections, hides from probes
- **Privacy Controls**: App permissions for location, camera, microphone
- **Find My Mac & Activation Lock**: Locates devices, prevents unauthorized reactivation
- **Runtime Protections**: ASLR, execute disable (XD), kernel hardening

---

9. Best Practices for OS Security

9.1 Core Practices

- **Regular Updates & Patch Management**: Fix known vulnerabilities
- **Principle of Least Privilege (PoLP)**: Grant minimum necessary permissions
- **Strong Authentication**: Complex passwords, MFA
- **File System Permissions**: Strict access controls, ACLs
- **Firewall Configuration**: Block unauthorized access
- **Antivirus & Anti-malware**: Keep updated
- **Logging & Auditing**: Monitor activity, review logs regularly
- **Vulnerability Scanning**: Address findings promptly

- **Software Whitelisting**: Allow only approved applications
- **Encryption**: Protect data at rest and in transit

9.2 OS-Specific Hardening

OS	Hardening Highlights
Windows	Group Policy, Secure Boot, Defender, SmartScreen
macOS	FileVault, Gatekeeper, SIP, Secure Enclave
Linux	Disable unused services, SELinux/AppArmor, iptables/nftables

9.3 Advanced Measures

- System call filtering (seccomp on Linux)
- Control Flow Integrity (CFI)
- Kernel integrity checks
- Application sandboxing

---

10. Viruses

Definition: Malicious software attaching to legitimate files to spread and cause harm

Key Characteristics:

- Self-replicating
- Requires activation trigger
- Carries payload (corrupt data, delete files, slow performance)

10.1 Types of Viruses

1. **File Infector**: Attaches to executables (.exe, .com)
2. **Macro Virus**: Targets macro-enabled documents (Word, Excel)
3. **Boot Sector Virus**: Infects master boot record
4. **Polymorphic Virus**: Alters code to evade detection
5. **Resident Virus**: Loads into memory, infects files continuously
6. **Direct Action Virus**: Activates only when infected file runs
7. **Multipartite Virus**: Attacks multiple areas simultaneously
8. **Overwrite Virus**: Replaces file content (unrecoverable)

9. **Browser Hijacker**: Alters browser settings
10. **Web Scripting Virus**: Injects code into websites
11. **Encrypted Virus**: Hides payload using encryption
12. **Stealth Virus**: Shows clean versions of infected files
13. **Armored Virus**: Resists analysis
14. **Companion Virus**: Creates malicious file with same name, different extension
15. **FAT Virus**: Targets File Allocation Table
16. **Spacefiller Virus**: Inserts code into unused file sections

10.2 Impact on OS Security

- Compromises integrity (alters/deletes system files)
- Threatens confidentiality (steals data)
- Disrupts availability (crashes, slows performance)
- Facilitates further attacks (opens doors for worms, trojans, ransomware)

10.3 Defense Strategies

- Real-time antivirus software (keep updated)
- OS-level protections (Windows Defender, macOS XProtect)
- Regular security patches
- Avoid untrusted downloads
- Safe browsing and email hygiene

---

11. Worms

Definition: Self-replicating malware spreading across networks without human intervention

Main Objective: Self-replicate and spread rapidly across networks by exploiting vulnerabilities

11.1 Key Characteristics

- Self-replicating without host file
- Autonomous propagation through networks
- No host dependency
- Network-centric behavior (consumes bandwidth)
- Payload delivery (deletes files, installs backdoors, steals data)
- Stealth and speed
- Resource exhaustion

- Remote execution capability
- Mass infection potential

11.2 Types of Worms

1. **Email Worms**: Spread via infected attachments (e.g., ILOVEYOU worm)
2. **Internet Worms**: Exploit OS/network vulnerabilities (e.g., Code Red)
3. **File-Sharing Worms**: Hide in P2P shared files (e.g., Phatbot)
4. **IM Worms**: Spread through chat apps (e.g., Kelvir - MSN Messenger)
5. **IRC Worms**: Use IRC channels to distribute files
6. **Cryptoworms**: Encrypt files and demand ransom
7. **P2P Worms**: Use peer-to-peer networks, disguised as media files

11.3 Defense Strategies

Proactive:

- Keep software updated (patch vulnerabilities)
- Install firewalls
- Use antivirus/anti-malware
- Harden system configurations
- Educate users

Reactive:

- Deploy IDS/IPS (intrusion detection/prevention)
- Blacklist infected IPs
- Rate-limit outgoing connections
- Apply emergency patches

Advanced:

- Use honeypots and darknets
- Randomize address space (ASLR)
- Isolate infected machines

---

12. Virus vs Worm

Aspect	Virus	Worm
Dependency	Needs host file/program	Self-contained, no host needed
Mode of Spread	Spreads when infected file executed	Spreads automatically through networks
Speed	Slower (depends on user action)	Rapid (no user interaction)
Payload	Corrupts/deletes data	Consumes bandwidth, system resources
Detection	Easier (modifies files)	Harder (stealthy, network-based)

---

13. Antivirus Software

Definition: Security program detecting, preventing, and removing malware

Threats Protected: Viruses, worms, trojans, spyware, adware, ransomware

13.1 What Antivirus Does

- Scans files/programs for known threats using signature database
- Monitors system behavior (heuristic analysis)
- Blocks malicious websites and downloads
- Quarantines or deletes threats
- Provides real-time protection

13.2 Detection Methods

- **Signature-Based**: Matches known virus code patterns (fast, reliable for known threats)
- **Heuristic Detection**: Flags unusual behavior (catches unknown threats, may have false positives)
- **Sandboxing**: Runs suspicious files in safe environment
- **Cloud-Based Analysis**: Uses online databases for emerging threats

13.3 Why Needed

Without protection, devices are vulnerable to:

- Data theft (passwords, banking info)
- System damage (corrupted files, crashes)
- Spyware and surveillance

- Ransomware attacks

---

14. Types of Antivirus

14.1 Detection-Based Types

1. **Signature-Based**: Compares files to malware database
2. **Heuristic-Based**: Analyzes file behavior/structure
3. **Behavior-Based**: Monitors real-time system behavior
4. **Machine Learning**: Uses AI to predict malicious behavior
5. **Cloud-Based**: Offloads scanning to cloud servers

14.2 Feature-Based Types

1. **Standalone Antivirus**: Focuses solely on virus detection
2. **Internet Security Suites**: Includes firewall, email protection, anti-phishing
3. **Endpoint Protection Platforms (EPP)**: For businesses, centralized management
4. **Mobile Antivirus**: For smartphones/tablets

Note: Modern antivirus combines multiple detection methods for multi-layered protection

---

15. Top 5 Antivirus Software (2025)

15.1 Bitdefender Antivirus Plus

- Excellent malware detection, multi-layered ransomware protection
- Features: VPN, anti-tracker, vulnerability scanner
- Best for: Minimal system impact

15.2 Norton AntiVirus Plus / Norton 360 Deluxe

- High lab test scores, strong phishing protection
- Features: Firewall, password manager, cloud backup, VPN
- Best for: Families, all-in-one security

15.3 McAfee Total Protection

- Covers unlimited devices, identity theft protection
- Features: File shredder, home network analyzer, scam detector
- Best for: Multiple devices, privacy concerns

15.4 Malwarebytes Premium

- Exceptional malware removal, real-time protection

- Features: Web protection, exploit mitigation
- Best for: Lightweight, effective malware cleaner

15.5 AVG Internet Security

- Strong antivirus engine, budget-friendly
- Features: Webcam protection, DNS hijack prevention, email shield
- Best for: Solid protection on budget

---

16. Keyloggers

Definition: Tool (software/hardware) recording every keystroke, often without user knowledge

Purpose: Monitor, track, or steal information typed on keyboard

Types:

1. **Software Keylogger**: Runs silently in background, sends data to remote server
2. **Hardware Keylogger**: Physical device between keyboard and computer

16.1 Infection Methods

- Phishing emails (malicious attachments/links)
- Malicious websites (automatic downloads)
- Bundled software (free/pirated apps)
- Drive-by downloads
- Social engineering (fake tech support, updates)
- Exploit kits (target outdated software)
- Physical access (plugged into device)

Example: DarkHotel attack targeted hotel Wi-Fi with fake software updates

16.2 Detection Methods

Behavioral Clues:

- Delayed typing
- Sluggish mouse movement
- Unexplained system slowdowns

Technical Checks:

- Check Task Manager for unfamiliar processes
- Review startup programs and installed apps
- Inspect browser extensions

- Physically examine for external devices

16.3 Removal Methods

1. Run full system scan with trusted antivirus (Bitdefender, Avast, Malwarebytes)
2. Clear temporary files manually or using Disk Cleanup
3. Reset browser settings
4. Factory reset PC (last resort after backup)

16.4 Protection Measures

- Keep OS and apps updated
- Avoid unreliable download sources
- Use virtual keyboards for sensitive information
- Enable two-factor authentication

---

17. Other Protection Measures

17.1 Strengthen Defenses

- Enable firewall
- Keep software updated
- Use strong passwords (password manager recommended)

17.2 Stay Vigilant Online

- Avoid suspicious links (hover before clicking)
- Limit downloads to trusted sources
- Use secure networks (VPN on public Wi-Fi)

17.3 Backup & Restore

- Regular backups to external drives or cloud
- Set up system restore points

17.4 Smart Habits

- Monitor system behavior
- Educate yourself on cybersecurity basics