Experiment 1

Objective

Provide a compact theoretical overview of common vulnerabilities, contemporary attacks, and how attackers exploit weaknesses.

What is a Vulnerability?

A vulnerability is a weakness in software, hardware, or configuration that can be abused to compromise confidentiality, integrity, or availability.

Common examples: weak/default passwords, unpatched software, misconfigured firewalls, open ports, insecure APIs, SQL injection (SQLi), cross-site scripting (XSS), buffer overflows, poor encryption, and human factors like phishing.

---

10 Contemporary Attacks

1. **Ransomware:** Malware encrypts files and demands payment for the decryption key. Impacts availability and leads to data loss.
2. **Phishing:** Deceptive emails/webpages trick users into revealing credentials or executing actions.
3. **Spear Phishing:** Targeted phishing at specific individuals using tailored information.
4. **Business Email Compromise (BEC):** Impersonation of executives to authorize fraudulent transfers.
5. **Zero-Day Exploits:** Attacks that use undisclosed or unpatched software flaws.
6. **Cryptojacking:** Unauthorized use of a system's resources to mine cryptocurrency.
7. **Man-in-the-Middle (MITM):** Intercepting or altering communications between parties.
8. **IoT Exploits:** Attacks on poorly secured smart devices, often due to weak default credentials or lack of updates.
9. **Advanced Persistent Threats (APTs):** Long-term, stealthy campaigns usually aimed at espionage or data exfiltration.
10. **Supply Chain Attacks:** Compromising vendors, updates, or third-party components to reach downstream victims.

How Attackers Exploit Vulnerabilities

- **Reconnaissance:** Identify targets, open ports, service versions, and public-facing apps.
- **Initial access:** Use phishing, weak passwords, or known vulnerabilities to gain entry.
- **Exploitation:** Inject code (SQLi/XSS), run malware, exploit buffer overflows, or misuse APIs.
- **Privilege escalation & lateral movement:** Seek higher access and pivot to other systems.
- **Persistence & exfiltration:** Install backdoors and steal or encrypt data.
- **Covering tracks:** Modify logs or use stealthy channels to avoid detection.

DDoS (Distributed Denial of Service)

A DDoS attack floods a service with traffic (volumetric, protocol, or application-layer) to make it unavailable. Common causes include botnets, amplification attacks (e.g., DNS/NTP amplification), and application-layer floods targeting specific endpoints.

Impact: downtime, lost revenue, reputational damage.
Basic mitigations: rate-limiting, traffic filtering (firewalls, WAF), CDNs and scrubbing services, redundancy and capacity planning.

Basic Remediations & Best Practices

- **Keep software patched** and remove unsupported systems.
- **Use strong, unique passwords + MFA.**
- **Validate and sanitize inputs; use parameterized queries** to prevent SQLi.
- **Implement Content Security Policy (CSP) + input/output encoding** for XSS defenses.
- **Encrypt data in transit & at rest** (TLS, strong ciphers).
- **Harden services and close unnecessary ports.**
- **Backups & recovery plans** to mitigate ransomware.
- **Vendor risk management & supply-chain review** to reduce third-party risk.
- **Logging, monitoring, and incident response** to detect and react quickly.