Segurança de Dados - Assignment 1

Pedro Malainho PG61005

1. Question 1

Operating system – macOS

Recent CVE's (last 12 months):

• *CVE-2025-43359 – macOS/ Kernel UDP Socket Exposure*

Description

Vulnerability in the macOS kernel that occurs when a UDP socket initially bound to a local interface can, due to a logic flaw, end up bound to all interfaces. This results in the unintended exposure of services that should be restricted to the local interface only, increasing the attack surface.

CVSS score and its interpretation

The NVD has not yet assigned an official score. According to other databases (CISA-ADP) in CVSS version 3.1, the score is 9.8 (Critical

Impact score: 5.9

Exploitation score: 3.9

Attack Vector (AV): Network

Attack Complexity (AC): Low

Privileges Required (PR): None

Users Interaction (UI): None

Scope (S): Unchanged

Confidentiality (C): High

Integrity (I): High

Availability (A): High.

Associated CWE category

CWE-670: Always-Incorrect Control Flow Implementation

Exploit available (Yes/No)

None.

Possible mitigations or fixes suggested

Yes, as it's a configuration/incorrect state vulnerability, mitigations are linked to the correct management of UDP traffic and system updates. The main possible mitigation was to download a new patch from Apple. In the meantime, firewalls, explicit socket configuration, and network auditing could be used as aditional defenses.

• *CVE-2025-43200 – Remote Code Execution via "Messages/iCloud link media"*

Description

It's a critical remote code execution vulnerability that exists in the macOS Messages application, specifically in the handling of photos and videos shared via iCloud links. A remote attacker, without authentication can trick the user into opening a specially crafted media file (video/image) via an iCloud link, thereby trigerring arbitrary code on the system.

CVSS score and its Interpretation

The NVD has not yet assigned an official score. According to other databases (CISA-ADP) in CVSS version 3.1, the score is 4.8 (Medium).

Impact score: 2.5

Exploitation score: 2.2

Attack Vector (AV): Network

Attack Complexity (AC): High

Privileges Required (PR): None

Users Interaction (UI): None

Scope (S): Unchanged

Confidentiality (C): Low

Integrity (I): Low

Availability (A): None

Associated CWE category

CWE-20 – Improper Input Validation

Exploit available (Yes/No) Yes, its still active.

Possible mitigations or fixes suggested

There are some possible mitigations such as:

– Update macOS to the fixed version.

– Not opening suspicious iCloud links with photos or videos.

–In companies, restricting automatic viewing of media from external links.

• CVE-2025-9074 – Docker Desktop

Description

A vulnerability in Docker Desktop that allows local Linux containers to access the Docker Engine API through the configured subnet (default 192.168.65.7:2375) could allow a malicious container to control the engine, create new containers, mount host files, and execute commands with the privileges of the user running Docker Desktop.

CVSS score and Its Interpretation

The NVD has not yet assigned an official score. According to other databases (CNA: Docker Inc.) CVSS version 4.0, the score is 9.3 (Critical).

Attack Vector (AV): Local

Attack Complexity (AC): Low

Attack Requirements (AT): None

Privileges Required (PR): None

Users Interaction (UI): Passive

Vulnerable System Confidentiality (VC): High

Vulnerable System Integrity (VI): High

Vulnerable System Availability (VA): High

Subsequent System Confidentiality (SC): High

Subsequent System Integrity (SI): High

Subsequent System Availability (SA): High

Associated CWE Category

CWE-668 Exposure of Resource to Wrong Sphere

Exploit available (Yes/No)

Yes. There are POC/public demonstrations and technical analyses showing how a container can leverage the exposed API. Several advisories and bulletins with details have been published.

Possible mitigations or fixes suggested

There are some possible mitigations such as:

- Update Docker Desktop immediately to the version that fixes the flaw (e.g., Docker Desktop 4.44.3 or later).
- Do not run untrusted containers locally; limit execution to signed/trusted images.
- Restrict access to the Docker subnet (firewall/ACLs) to prevent containers from accessing 192.168.65.7:2375.
- Avoid exposing the daemon (disable TCP exposure options for the daemon and use Unix sockets only, with appropriate authentication).
- Eliminate mounting the Docker socket in containers (do not mount /var/run/docker.sock unless strictly necessary) and avoid privileged containers.
- Monitoring/detection: use EDR/IDS to identify escape attempts, monitor unexpected container creation, and abnormal daemon operations.

What is the impact of the vulnerability on a production environment?

This vulnerability in Docker Desktop allows a malicious container to access the Docker Engine API on the host. In production, this means and attacker could escape container isolation, create or destroy containers, mount sensitive directories, or run arbitrary commands with the privileges of the Docker Desktop user. This can cause leakage of sensitive data, persistence of malware, compromise of host system, and lateral movement across the infraestructure.

Which systems could be compromised?

Any developer or producktion workstation running Docker Desktop on macOS or Windows, indirectly other connected systems in the network (if the attacker gains host-level control and moves laterally), also any CI/CD pipelines or local environments that rely on Docker Desktop for builds and testing are particulary at risk.

What preventive measures can reduce the risk?

Restrict container execution to trusted images only- avoid running untrusted or third-party containers, apply network restrictions to block access from containers to the special Docker subnet, avoid mouting the Docker socket into containers unless strictly necessary, and monitor logs and container activity to detect unexpected API calls.

Is there a patch available, and how can it be applied? If there is no patch, suggest alternatives to mitigate the risk.

Yes, a patch is available. Docker fixes the issue in Docker Desktop 4.44.3 and later. To apply the patch, you need to update Docker Desktop through the built in updater or download the latest version.

2. Question 2

There are three main parties involved in a credit card based sysyem:

- Client (Cardholder)

- Seller

- Credit card operator/ Bank/ Payment processor

Such a system must ensure certain security properties, and at the same time it faces well-known threat classes.

The security properties talked about are:

• Confientiality - Could be credit card numbers, CVV codes, authentication tokens and transaction details must be disclosed to unauthorized entities. Also communications must be encrypted.

• Integrity – Transaction details (amount, identity) must not be altered in transit. Also digital signatures, MACs or authenticated encryption are needed.

• Authentication – The client must prove they are the legitimate cardholder, the seller also needs to prove that they are legitimate merchant. The operator must prove they are the real payment processor.

• Authorization – Only transactions explicitly approved by the client should be processed and merchants should only be allowed to debit agreed amounts.

• Non-repudiation – The client cannot deny having authorized a transactions, and the merchant cannot deny having received the payment. This can be achieved through digital receipts, logs and signatures.

• Availability – Every action should be traceable.

As said the system also has threats shuch as:

• Eavesdropping/ Interception – Attackers sniff traffic to steal card data (This can be achieved with an attack known by man-in-the-midle and packet sniffing).

• Data Modification/ Tampering – Changing transactions details during transmissions.

• Spoofing/ Impersonating – Fake merchang websited collecting card details (Phishing), also attackers could be impersonating the bank or clients.

• Replay Attacks – Reusing valid transaction messages to trick the system into duplicating charges.

• Malware/Skimming – On the client side, keyloggers or trojans stealing information like credentials. On the merchant side, compromised POS (Point of sale) or web servers

• Denial of Service (DoS/DDoS) – Overloading the payment gateway or merchant site to prevent legitimate transactions. • Fraudulent Transactions – Using stolen credit card data to authorize purchases.

• Insider Threats – Malicious employees at merchants or operators misusing access to sensitive data.

3. Question 3

Hard-coded Cryptographic Keys (CWE-321)

Two of the listed CVEs (CVE-2025-30406 & CVE-2025-52374) fall under this category. This vulnerability arises primarily from weak key management practices, where developers embed cryptographic keys directly into source code or binaries. Once these keys are extracted by an attacker, the entire security model collapses, since the attacker can decrypt sensitive data, forge messages, or impersonate legitimate parties.

Security properties impacted: Confidentiality, Integrity, Authentication.

Improper Validation of Cryptographic Checks (CWE-354) Another CVE (CVE-2025-54887) relates to the incorrect validation of message authentication codes (MACs), digital signatures, or hash values. When these integrity checks are not properly enforced, attackers can modify or inject malicious data without being detected. This directly undermines integrity and can facilitate the introduction of harmful payloads into the system.

Security properties impacted: Integrity (core), and potentially Authentication.

Improper Certificate/Identity Validation (CWE-297)

The final CVE (CVE-2025-3501) involves systems that accept invalid or mismatched TLS/SSL certificates. This flaw enables man-in-the-middle (MITM) attacks, where an attacker intercepts and manipulates communication between client and server. The impact is severe, as it compromises both confidentiality (data interception) and authentication (false server identity).

Security properties impacted: Confidentiality, Authentication, Integrity.

The vulnerabilities arise not from flaws in cryptographic algorithms themselves, but from implementation and operational mistakes: hard-coded keys, missing validation, and improper certificate handling. These errors weaken core security properties and show that secure cryptography requires both strong algorithms and correct system integration.