Homework for week 3

Attack Surface and Security Measures Analysis

1. Word Processor (WoPro)

Potential Attack Vectors:
Word processors accept various types of input, making them susceptible to multiple attack vectors:

- **Document Files:** Formats such as .docx, .odt, .rtf, and .pdf can carry malicious payloads.
- **Embedded Objects:** Macros, scripts, ActiveX controls, and OLE objects embedded within documents.
- **Clipboard Data:** Copy-paste operations that can introduce hidden or malformed data.
- **Network Resources:** Hyperlinks and embedded media referencing external servers.
- **Fonts and Styles:** Custom fonts that exploit rendering vulnerabilities.

Vulnerabilities:

- **Macro Exploits:** Malicious macros can execute unauthorised code.
- **Buffer Overflows:** Poorly handled file parsing can lead to memory corruption.
- **Script Injection:** Embedded JavaScript in documents.
- **Privilege Escalation:** Exploiting software bugs to gain higher-level access.

Security Measures:

- **Secure Coding Practices:** Input validation, disabling macros by default, and code signing for trusted macros.
- **Design Principles:** Principle of Least Privilege, sandboxing document processing, and regular software updates.
- **System-Level Protections:** Data Execution Prevention (DEP), Address Space Layout Randomisation (ASLR), and antivirus scanning for documents.

2. Media Player (MPlayer)

Potential Attack Vectors:
Media players handle various data formats, increasing their attack surface:

- **Media Files:** Audio (e.g., .mp3, .wav) and video (e.g., .mp4, .avi) files.
- **Streaming Content:** Real-time data streams (e.g., HTTP, RTSP).
- **Embedded Metadata:** ID3 tags and EXIF metadata containing crafted data.
- **Subtitles:** Subtitle files (.srt, .sub) with malicious scripts.
- **Plugins:** Third-party extensions with potential security flaws.

Vulnerabilities:

- **Buffer Overflows:** Vulnerabilities in codec parsing.
- **Codec Exploits:** Third-party codec libraries may have unpatched vulnerabilities.
- **Heap Corruption:** Through malformed media files.
- **Privilege Escalation:** Exploiting system-level permissions.

Security Measures:

- **Secure Coding Practices:** Strong bounds checking, validating metadata, and secure memory management.
- **Design Principles:** Defence in Depth, use of memory-safe languages, and minimising attack exposure.
- **System Protections:** Running media players in isolated containers and enforcing strict permission controls.

---

3. Web Browser

Potential Attack Vectors:
Web browsers have an extensive attack surface due to diverse functionalities:

- **Web Content:** HTML, CSS, JavaScript, and multimedia files.
- **Browser Extensions:** Add-ons with elevated privileges.
- **User Inputs:** Form fields, URL parameters, and cookies.
- **Network Communications:** HTTP/HTTPS traffic, WebSockets, and APIs.
- **Third-Party Scripts:** External libraries embedded in websites.

Vulnerabilities:

- **Cross-Site Scripting (XSS):** Injecting malicious scripts into web pages.
- **Cross-Site Request Forgery (CSRF):** Exploiting authenticated sessions.
- **Man-in-the-Middle Attacks:** Intercepting unsecured data.
- **Memory Corruption:** Use-after-free and buffer overflow vulnerabilities.

Security Measures:

- **Secure Coding Practices:** Input sanitisation, Content Security Policy (CSP), and Same-Origin Policy enforcement.
- **Design Principles:** Process isolation, sandboxing, and regular patch management.
- **System Support:** Enforcing HTTPS, browser updates, and employing network security tools.
