**DEVELOPMENT PHASE: FILE SANITIZATION UTILITY**

**OBJECTIVE:**

The main objective of this is to clean the files that are of malicious content and ensuring that they are safe to open and use.

**METHODS IN FILE SANITIZATION UTILITY:**

**1.Content disarm and reconstruction:**

Break the file into small units. Scan each unit and find whether there is any malicious content or not. If malicious remove all elements from it. The cleaned components are reassembled to create a new, safe version of the original file.

**2.signature-based detection:**

It is used to identify malicious activity and also it examines network traffic, compares it to known signatures, and generate an alert if it finds a match.

A collection of unique patterns, identifiers are identified in malware. These signatures can be binary patterns, file hashes. If the file contains patterns that match a signature in the database, the scanner flags it as malicious. And this flagged file can be deleted or quarantined.

**3. Heuristic analysis:**

First you have to analyse the file for suspicious behaviour. Then check for malicious software. you can now access file using heuristic predefined rules. **Now file behaviour is being scored high for being suspicious based on Rule-Based Decision**.

**4.** **Macro stripping:**

removing macros from files, particularly from documents such as Microsoft Word or Excel files. Macros are scripts which are embedded within these documents that can execute automated tasks.

**You need to scan file if there are any embedded micros.** Macros are written in scripting languages like Visual Basic for Applications (VBA).

Detect macros and try to delete it.When you disable file macros cannot execute.

The process ensures that the main content of the document (text, images, formulas) remains unaffected.

Only the macro code, which poses potential security risks, is removed.

**5.** **Binary padding removal:**

Binary in the sense 1’s and 0’s that actually execute on computer and identify vulnerabilities. Binary padding includes adding extra bits, usually zeros at the end. Binary padding removal means taking off these extra bits that were added to data. So when you receive or read this padded data you try to strip away these extra bits to get back to original data.

**6. File Type Validation and Filtering:**

When a file is received, its type is checked, like whether it is from .jpeg or .pdf extension. So based on validation it will decide whether to allow or block.

Safe or allowed file types are let through, while dangerous or disallowed types are blocked or quarantined.

**7. Active Content Removal:**

**Scan the file** to find any active elements like macros. Remove this element in such a way that this file can no longer execute scripts. The main content will remain on file such as text and images.

**8.Embedded Object Extraction and Analysis:**

The file is scanned to detect any embedded objects such as images, documents, spreadsheets, or other files that are included within the main file. These embedded objects are separated and extracted from the main file for individual examination.

Each extracted object is thoroughly analyzed for any signs of malicious content or behaviour.

This analysis can include scanning for viruses, checking for suspicious code, and validating the file type and integrity.

If any embedded object is found to be malicious, it is either cleaned (if possible) or removed.

The cleaned or sanitized main file is then reconstructed without the harmful embedded objects.

**Required Technologies and Tools:**

**Backend**: Node.js, Python, or Java for backend services for advanced file sanitization

**Client Tool**: Python CLI, Go or C++ etc

**Libraries and ExtraTools**: Regular Expressions, PyPDF2, cryptography.fernet, Pandas, PIL , KVM & QMEU etc

**Conclusion:**

By applying these techniques, a File Sanitization Utility can effectively neutralize threats and protect sensitive information, ensuring files are safe for further processing or distribution. Implementing a combination of these methods enhances security and minimizes risks associated with file handling.