**Work Request: Simple Linux File Audit Program**

**Objective**

Implement a cross-platform workflow in which the Linux system monitors file changes, queries **VirusTotal** by SHA-256, and forwards only **unknown** files to a Windows environment. The Windows side accumulates files and runs periodic batch scans using **Windows Defender (MpCmdRun.exe) or third-party malware scan software**. The solution must provide a way to **submit files** and **retrieve processing results**; the specific mechanism is not predetermined.

**Scope of Work**

**Linux Component**

* **File Monitoring**
  + Detect file creation/modification (e.g., inotify or equivalent).
  + Collect basic metadata (path, size, timestamps).
* **Hash & Reputation**
  + Compute SHA-256 for each detected file.
  + Query **VirusTotal** by hash.
  + If found: cache and return the reputation.
  + If not found: mark for Windows submission.
* **Submission & Result Retrieval**
  + Submit unknown files to the Windows system (planned protocol: **FTP**; may include staging/queuing).
  + Provide a capability to retrieve the **processing result** for each submitted file after scanning completes (e.g., polling, callback, file drop, MQ—implementation to be decided).
* **Caching**
  + Store results in local **SQLite** with TTL for consistent responses on repeated requests.

**Windows Component (Background)**

* **Collection**
  + Continuously watch the incoming FTP directory and queue files for scanning.
* **Batch Scanning**
  + On a defined schedule, invoke **MpCmdRun.exe or third-party malware scan software** to scan all queued files in bulk.
  + Normalize outputs into structured results (e.g., JSON with hash, verdict, timestamps, tool/version).
* **Result Delivery**
  + Make results available to Linux through the chosen retrieval method, preserving reliable mapping to the original file/hash.

**Expected Workflow (High Level)**

1. Linux detects file change → computes SHA-256.
2. Linux queries VirusTotal.
3. **If found** → return and cache reputation.
4. **If not found** → enqueue file; transfer to Windows via **FTP** on schedule.
5. Windows batches and scans via **MpCmdRun.exe or third-party malware scan software**.
6. Results are produced and made available to Linux.
7. Linux ingests results, updates cache and serves consistent responses within TTL.

**Key Requirements**

* **VirusTotal First**: Only files not present in VirusTotal are forwarded.
* **Batch Operation**: Transfers and Defender scans run on **scheduled intervals** to reduce load.
* **Background Execution**: Windows side runs continuously in the background (formal Windows Service is optional).
* **Submission & Results**: Provide a method to submit files and obtain their scan results after processing; exact mechanisms will be selected during implementation.
* **Caching & Consistency**: SQLite cache with TTL; identical files (same SHA-256) must yield consistent responses within TTL.
* **Resilience & Security**:
  + Retries/backoff for FTP and result retrieval.
  + Secure handling of VirusTotal API keys and FTP credentials.
  + Basic integrity (e.g., HMAC or checksum) on results channel is preferred.

**Deliverables**

* Linux monitoring program with VirusTotal integration and SQLite caching.
* Windows background scanning program (batch runner around **MpCmdRun.exe or third-party malware scan software**) plus configuration for scheduled execution.
* Configuration/operations guide describing file submission and result retrieval options and how to enable the chosen method.
* Result data schema (e.g., JSON) and mapping rules (filename ↔ hash ↔ result).
* Test report covering VT queries, FTP transfers, batch scans, and end-to-end result retrieval.

**Acceptance Criteria**

* Reliable detection of create/modify events and correct SHA-256 computation on Linux.
* Successful VirusTotal lookups; “found” cases returned from cache within TTL.
* Unknown files transferred to Windows in scheduled batches via FTP.
* **MpCmdRun.exe or third-party malware scan software** executes on schedule and produces standardized results for each file.
* Linux successfully retrieves results through the selected method and updates cache.
* Repeated queries for the same file return consistent responses while within TTL.
* System tolerates transient network/API failures with automatic recovery.

[VirusTotal API]

Docs: [VirusTotal API v3 Overview](https://docs.virustotal.com/reference/overview)

If you have any questions, please contact: [gnh1201@catswords.re.kr](mailto:gnh1201@catswords.re.kr)