**Standard Operating Procedure (SOP) for AbuseIPDB & VirusTotal Automation**

**1. Objective**

This SOP outlines the procedure for automating the retrieval of **blacklisted IP addresses** from **AbuseIPDB**, performing **IP reputation checks** using **VirusTotal**, storing the results in **MongoDB**, and sending an automated **email report** with the results.

**2. Pre-requisites**

**2.1 Accounts & API Keys**

* **AbuseIPDB API Key** (For fetching blacklisted IPs)
* **VirusTotal API Key** (For IP reputation checks)
* **AWS Secrets Manager** (For securely storing credentials)
* **MongoDB Server** (Local or Cloud, for storing VirusTotal results)
* **SMTP Email Credentials** (For sending reports)

**2.2 Required Software & Python Libraries**

**AWS CLI Setup**

Download and install AWS CLI:  
<https://awscli.amazonaws.com/AWSCLIV2.msi>

Open a terminal or Command Prompt and run:

**aws configure**

You will be prompted to enter:

* AWS Access Key ID
* AWS Secret Access Key
* Default Region Name (e.g., us-east-1)
* Default Output Format (leave blank or enter json)

Install the necessary Python libraries:

**pip install -r requirements.txt**

**3. Workflow Steps**

**Step 1: Fetch Credentials from AWS Secrets Manager**

* The script retrieves API keys, MongoDB credentials, and SMTP details from **AWS Secrets Manager**.
* If credentials retrieval fails, the script logs an error and exits.

**Step 2: Check for Locally Cached AbuseIPDB Data**

* The script checks if **AbuseIPDB blacklist data** was retrieved within the last **24 hours**.
* If found, it uses the local copy. otherwise, it fetches new data from **AbuseIPDB API**.

**Step 3: Retrieve Blacklist Data from AbuseIPDB**

* Calls **AbuseIPDB API** and retrieves IPs with an **abuse confidence score > 97**.
* Saves data to a **JSON file** (abuseipdb\_blacklist.json).
* Also saves the data to an **Excel file** (AbuseIPDB\_data.xlsx).

**Step 4: Fetch IP Reputation from VirusTotal**

* Selects the first **50 IP addresses** from the **AbuseIPDB blacklist**.
* For each IP:
  + Calls **VirusTotal API** to check its reputation.
  + Implements **exponential backoff** to handle rate limits (max **4 requests per min**).
* Extracts key details:
  + **Country**
  + **Detected URLs**
  + **Detected Downloaded Samples**
  + **Undetected Downloaded Samples**
  + **Undetected URLs**
* Logs errors if an IP fails after **5 retry attempts**.

**Step 5: Store VirusTotal Data in MongoDB**

* Connects to the **MongoDB server** using credentials from **AWS Secrets Manager**.
* Saves the VirusTotal data to the **'AbuseVT' collection** inside the **'optiv' database**.
* If MongoDB connection fails, logs an error and exits.

**Step 6: Send an Automated Email Report**

* Composes an **HTML email** with:
  + A **formatted table** containing **VirusTotal reputation data**.
  + The **AbuseIPDB blacklist Excel file** attached.
* Sends the email using **SMTP** to[**lokesh.kumawat@optiv.com**](mailto:lokesh.kumawat@optiv.com).
* If email sending fails, logs an error.

**4. Error Handling & Logging**

* Uses **Python logging** to maintain a log file (abuseIPDB\_VirusTotal.log).
* Handles errors such as:
  + **Network failures** when calling APIs.
  + **Rate limit exceeded (429 Too Many Requests)** with exponential backoff.
  + **MongoDB connection errors**.
  + **Email sending failures**.
* If a critical error occurs (e.g., failed credential retrieval, MongoDB failure), the script exits safely.