Email Security Analyzer: API Integrations

Integration Architecture

I integrated five external security intelligence APIs to create email analysis coverage. Each service provides different threat intelligence data and combining them creates more accurate phishing detection. The modular design means each API integration operates independently, so failures in one service don't break the entire analysis.

VirusTotal API

VirusTotal aggregates results from multiple antivirus engines and URL scanners, providing security vendor consensus on domain reputation. I chose this service because it offers the broadest coverage of malware detection engines.

API Endpoint	https://www.virustotal.com/api/v3/domains/{domain}
Authentication	API key in HTTP headers (x-apikey)
Data Returned	Security vendor analysis, malware detection, reputation scores, malicious indicators

Implementation makes GET requests to domain endpoints and processes JSON responses. The security vendor analysis shows how many engines flagged the domain as malicious, providing confidence scores for threat assessment.

AbuseIPDB API

AbuseIPDB tracks malicious IP addresses with confidence scores based on community reports. This catches phishing infrastructure even when domains are newly registered and haven't been analyzed by antivirus engines yet.

API Endpoint	https://api.abuseipdb.com/api/v2/check
Authentication	API key in HTTP headers (Key)
Data Returned	Abuse confidence score (0-100), total reports, abuse categories, ISP information

Implementation queries IP-specific endpoints with the ipAddress parameter and parses JSON for abuse confidence. High confidence scores indicate widespread malicious activity from that IP address.

WhoisXML API

WhoisXML provides domain registration data for legitimacy assessment. Newly registered domains or domains with privacy-protected WHOIS records often indicate phishing infrastructure.

API Endpoint	https://www.whoisxmlapi.com/whoisserver/WhoisService
Authentication	API key in query parameters (apiKey)
Data Returned	Registration dates, registrant details, name servers, registrar information, contact details

Implementation passes the API key with domainName and outputFormat=JSON parameters, using regex for data extraction from the response. Registration age is a key indicator since phishing domains are typically registered recently.

Google Custom Search API

Google Custom Search enables programmatic web searches for domain presence verification. Legitimate organizations typically have established web and social media presence, while phishing domains often lack any legitimate online footprint.

API Endpoint	https://www.googleapis.com/customsearch/v1
Authentication	API key and search engine ID (key, cx parameters)
Data Returned	Page titles, URLs, snippets indicating social media and web presence

Implementation constructs targeted searches with the site: operator and returns top 5 results per query. This verifies whether domains have legitimate social media profiles and established web presence.

DNS Resolution (dnspython)

Direct DNS queries validate email authentication records without requiring external APIs. This provides immediate SPF and DMARC verification since DNS is publicly accessible infrastructure.

Implementation	Direct queries to authoritative name servers using public DNS
Authentication	No authentication required (public DNS infrastructure)
Data Returned	SPF record configuration, DMARC policy, pass/fail status for each mechanism

Implementation uses dns.resolver.resolve() for TXT record queries, checking for SPF and DMARC records. Legitimate organizations configure these authentication mechanisms to prevent email spoofing.

Data Flow & Processing

Each API integration follows a consistent request pattern that handles authentication, error management, and data extraction:

- Construct API URL with required parameters
- Add authentication credentials in headers or query parameters
- Make HTTP GET request using requests library
- Verify HTTP 200 status code before processing
- Parse JSON response data extracting relevant fields
- Handle exceptions with fallback messaging

Results from multiple APIs get consolidated into a single analysis report. The tool cross-references data to assess overall legitimacy and combines security findings for comprehensive blocklist status. This aggregation provides higher confidence assessment than individual sources.

Technical Requirements

Running the analyzer requires these dependencies and API accounts:

- Python 3.8 or higher
- requests library for HTTP communications
- dnspython for DNS resolution
- Standard Python modules: email, re, json, argparse, sys
- VirusTotal API key (free or premium tier)
- AbuseIPDB API key (free tier available)
- WhoisXML API subscription (paid service)
- Google Custom Search API key and search engine ID

The API accounts represent the main setup complexity. Most services offer free tiers suitable for development and testing, though production use may require paid subscriptions depending on volume.

Why Multi-API Integration Matters

Integrating multiple security intelligence sources provides several advantages:

- Broader threat coverage catching attacks missed by individual services
- Higher confidence assessment through consensus across sources
- Resilience against single service failures or unavailability
- Different temporal coverage since services update at different rates
- Complementary data types addressing different attack indicators

The modular architecture makes the tool maintainable and extensible. Each API integration operates independently through standardized request patterns, so adding new services or replacing existing ones doesn't require architectural changes. This demonstrates practical API integration skills applicable to security automation and threat intelligence platforms.