**Updated .mac Email Extraction with Complete Domain Analysis**

**📧 UPDATED PROMPT 1: .mac Email Extractor with Domain Discovery**

Create a Python script called `scripts/mac\_email\_domain\_analyzer.py` for the Mingus financial wellness app that:

CONTEXT: I have a .mac email address with financial newsletter emails stored in a folder called "Mingus". Need to extract ALL URLs first, analyze domains, then filter for article library. This will help me understand what sources I'm receiving and make informed filtering decisions.

REQUIREMENTS:

1. Connect to iCloud email using IMAP (imap.mail.me.com, port 993)

2. Authenticate using .mac email address and app-specific password

3. Access the "Mingus" folder specifically (not inbox)

4. Extract ALL URLs from all emails (no filtering initially)

5. Perform comprehensive domain analysis BEFORE filtering

6. Generate domain statistics and reports

7. THEN apply configurable filtering for final article list

8. Export multiple files for analysis and processing

DOMAIN ANALYSIS FEATURES:

1. Extract and count all unique domains found

2. Categorize domains by type (news, financial, social media, etc.)

3. Calculate URL frequency per domain

4. Identify top sending email addresses by domain

5. Analyze URL patterns and subdomains

6. Generate domain popularity rankings

7. Export comprehensive domain report

EXTRACTION PROCESS:

Phase 1: Complete URL Discovery

- Extract every URL from every email (no filtering)

- Parse email metadata (subject, sender, date)

- Handle HTML and text email formats

- Clean tracking parameters but preserve original URLs

- Record which email each URL came from

Phase 2: Domain Analysis

- Parse all domains from extracted URLs

- Count URLs per domain

- Identify subdomain patterns

- Categorize domain types automatically

- Flag potentially suspicious or spam domains

- Calculate domain authority/trust scores

Phase 3: Domain Categorization

- Financial websites (bankrate, nerdwallet, etc.)

- News and media sites (cnn, bbc, etc.)

- Social media platforms (linkedin, twitter, etc.)

- Educational institutions (.edu domains)

- Government sites (.gov domains)

- Business/corporate sites

- Blog platforms (medium, wordpress, etc.)

- E-commerce sites

- Unknown/uncategorized domains

Phase 4: Configurable Filtering

- Load domain whitelist from config file

- Apply filtering rules based on analysis

- Allow manual domain approval/rejection

- Generate filtered article URL list

- Preserve audit trail of filtering decisions

OUTPUT FILES:

1. `raw\_urls\_complete.csv` - All URLs with email metadata

2. `domain\_analysis\_report.csv` - Complete domain statistics

3. `domain\_categories.json` - Categorized domain list

4. `top\_domains\_summary.txt` - Human-readable summary

5. `filtered\_article\_urls.csv` - Final filtered list for processing

6. `domain\_whitelist\_suggestions.txt` - Recommended domains to include

DOMAIN REPORT STRUCTURE:

- Domain name

- Total URL count

- Unique URL count

- First appearance date

- Last appearance date

- Email sender frequency

- Estimated domain category

- Trust/authority score (1-10)

- Sample URLs (up to 5)

- Recommended action (include/exclude/review)

INTERACTIVE FEATURES:

- Display top 20 domains during processing

- Prompt user to approve/reject unknown domains

- Show sample URLs for domain verification

- Allow real-time whitelist modification

- Progress tracking with domain discovery counts

SECURITY AND VALIDATION:

- Secure password input (no echo)

- URL format validation and sanitization

- Malware/phishing domain detection (basic)

- Rate limiting for external domain checks

- Graceful handling of connection issues

PERFORMANCE OPTIMIZATION:

- Batch domain analysis for efficiency

- Cache domain category lookups

- Parallel processing where appropriate

- Memory-efficient URL storage

- Progress saving for large email collections

Generate clean, documented Python code that gives complete visibility into email URL sources before applying any filtering, enabling informed decision-making about which domains to include in the Mingus article library.

**📊 PROMPT 1B: Domain Analysis Report Generator**

Create a companion script called `scripts/generate\_domain\_report.py` that:

CONTEXT: After extracting all URLs and domains from .mac Mingus folder emails, need comprehensive reporting and visualization to understand email sources and make filtering decisions.

REQUIREMENTS:

1. Read raw URL data from domain analyzer output

2. Generate detailed statistical analysis of domains

3. Create visual charts and graphs for domain distribution

4. Identify patterns in email sending behavior

5. Flag potential spam or low-quality sources

6. Generate recommendations for domain filtering

7. Create interactive HTML report for easy review

ANALYSIS FEATURES:

Domain Frequency Analysis:

- Top 50 domains by URL count

- Top 20 domains by unique URL count

- Domain growth over time (if date data available)

- Sender-to-domain relationship mapping

- URL patterns within each domain

Content Quality Indicators:

- Average URL length per domain

- Subdirectory patterns that suggest article content

- Presence of tracking parameters by domain

- Domain authority estimation (using external APIs if available)

- Content freshness indicators

Spam/Quality Detection:

- Domains with excessive tracking parameters

- Suspicious URL patterns (random strings, etc.)

- Domains with very short URLs (potential link shorteners)

- High-frequency domains that might be spam

- Domains with mixed content types

Sender Analysis:

- Which email addresses send from which domains

- Sender reliability scoring

- Newsletter vs individual sender patterns

- Corporate vs personal email patterns

VISUALIZATION FEATURES:

- Bar charts of top domains by frequency

- Pie charts of domain categories

- Timeline charts of domain discovery

- Network graphs of sender-domain relationships

- Heat maps of URL frequency over time

REPORTING OUTPUT:

1. `domain\_analysis\_full\_report.html` - Interactive HTML dashboard

2. `domain\_recommendations.txt` - Actionable filtering suggestions

3. `suspicious\_domains.csv` - Domains flagged for review

4. `high\_value\_domains.csv` - Recommended domains for inclusion

5. `domain\_statistics.json` - Complete stats for further processing

6. `sender\_domain\_mapping.csv` - Email sender analysis

RECOMMENDATION ENGINE:

- Automatically suggest domains for inclusion based on:

\* Financial/career keyword presence in URLs

\* Domain authority and reputation

\* Content quality indicators

\* Sender trustworthiness

\* URL pattern analysis

- Flag domains for manual review:

\* Unknown domains with high URL counts

\* Domains with suspicious patterns

\* Mixed-quality content domains

\* Potential affiliate/marketing domains

INTERACTIVE FEATURES:

- Clickable domain links in HTML report

- Sortable tables by various metrics

- Expandable sections for detailed domain info

- Sample URL preview for each domain

- Quick approve/reject buttons for filtering decisions

HTML DASHBOARD SECTIONS:

1. Executive Summary (top stats and recommendations)

2. Domain Distribution Overview (charts and graphs)

3. Top Domains Table (detailed metrics)

4. Sender Analysis (email source patterns)

5. Quality Assessment (content indicators)

6. Filtering Recommendations (suggested actions)

7. Detailed Domain Breakdown (full analysis)

Generate a comprehensive analysis tool that transforms raw domain data into actionable insights for building a high-quality article library.

**📈 EXPECTED OUTPUT EXAMPLE**

**Raw Domain Discovery Results:**

=== DOMAIN ANALYSIS SUMMARY ===

Total URLs Found: 1,247

Unique Domains: 89

Email Sources: 23

TOP DOMAINS BY FREQUENCY:

1. nerdwallet.com (156 URLs, 23% of total)

2. blackenterprise.com (98 URLs, 8% of total)

3. linkedin.com (87 URLs, 7% of total)

4. medium.com (76 URLs, 6% of total)

5. bankrate.com (65 URLs, 5% of total)

6. investopedia.com (54 URLs, 4% of total)

7. thebalance.com (43 URLs, 3% of total)

8. essence.com (38 URLs, 3% of total)

9. forbes.com (35 URLs, 3% of total)

10. cnbc.com (32 URLs, 3% of total)

UNEXPECTED DOMAINS FOUND:

- amazon.com (45 URLs) - Product recommendations?

- youtube.com (38 URLs) - Video content

- substack.com (29 URLs) - Newsletter platforms

- twitter.com (27 URLs) - Social media links

- shopify.com (15 URLs) - E-commerce links

- calendly.com (12 URLs) - Scheduling links

**Domain Categories Discovered:**

FINANCIAL WEBSITES (387 URLs, 31%):

✅ nerdwallet.com, bankrate.com, investopedia.com, thebalance.com, mint.com

CAREER/BUSINESS (298 URLs, 24%):

✅ linkedin.com, blackenterprise.com, essence.com, forbes.com, harvard.edu

NEWS/MEDIA (187 URLs, 15%):

❓ cnbc.com, marketwatch.com, wsj.com, bloomberg.com

CONTENT PLATFORMS (156 URLs, 13%):

❓ medium.com, substack.com, youtube.com

E-COMMERCE (89 URLs, 7%):

❌ amazon.com, shopify.com, affiliate links

SOCIAL MEDIA (78 URLs, 6%):

❌ twitter.com, facebook.com, instagram.com

TOOLS/SERVICES (52 URLs, 4%):

❌ calendly.com, zoom.us, google.com

**Filtering Recommendations:**

RECOMMENDED FOR INCLUSION (High Value):

✅ nerdwallet.com (financial education)

✅ blackenterprise.com (cultural relevance)

✅ investopedia.com (financial education)

✅ linkedin.com/pulse (professional content)

✅ harvard.edu (educational authority)

RECOMMENDED FOR REVIEW (Medium Value):

❓ medium.com (mixed quality, manual review needed)

❓ forbes.com (high authority, some promotional content)

❓ cnbc.com (news vs educational content)

RECOMMENDED FOR EXCLUSION (Low Value):

❌ amazon.com (product sales, not educational)

❌ calendly.com (scheduling tools, not content)

❌ twitter.com (social media, not articles)

**🎯 Benefits of This Approach**

**✅ Complete Visibility**

* See **every domain** in your email collection
* Understand the **full scope** of your sources
* Identify **unexpected high-value domains** you might have missed

**✅ Informed Decision Making**

* **Data-driven filtering** instead of guesswork
* **Discover hidden gems** in your email collection
* **Avoid over-filtering** that might exclude good content

**✅ Quality Control**

* **Identify spam domains** automatically
* **Spot affiliate marketing** vs educational content
* **Ensure cultural relevance** for your target audience

**🚀 Usage Instructions**

1. **Run the domain analyzer first** to get complete domain discovery
2. **Review the generated reports** to understand your email sources
3. **Make informed filtering decisions** based on the analysis
4. **Customize the whitelist** for your specific needs
5. **Process the filtered URLs** through the article scraping pipeline

This approach gives you **complete control** over your article sources while ensuring you don't miss any valuable content hidden in unexpected domains.

**Would you like me to also create a prompt for an interactive domain approval interface where you can quickly approve/reject domains with sample URL previews?**

**Bonus: Interactive Domain Approval Web Interface**

**🌐 BONUS PROMPT: Interactive Domain Approval Interface**

Create a simple Flask web application called `scripts/domain\_approval\_interface.py` for Mingus that:

CONTEXT: After discovering all domains from .mac email extraction, need a fast, visual way to approve/reject domains for the article library. This interface should make domain filtering decisions quick and intuitive.

REQUIREMENTS:

1. Simple Flask web interface that runs locally (localhost:5000)

2. Load domain analysis results from previous scripts

3. Display domains in a card-based layout with key information

4. Show sample URLs from each domain for quick assessment

5. One-click approve/reject/review buttons for each domain

6. Real-time progress tracking of approval decisions

7. Export approved domains to configuration files

8. Integration with existing Mingus article processing pipeline

INTERFACE FEATURES:

Domain Card Layout:

- Domain name prominently displayed

- URL count and percentage of total

- Sample URLs (3-5 representative examples)

- Domain category suggestion (Financial, News, Career, etc.)

- Trust/authority score visualization

- Quick action buttons (Approve, Reject, Review Later)

Quick Decision Tools:

- Bulk approve by category (e.g., "Approve all Financial domains")

- Search/filter domains by name or category

- Sort by URL count, category, or alphabetical

- Keyboard shortcuts (A=approve, R=reject, Space=next)

- Undo last decision capability

Progress Tracking:

- Progress bar showing decisions made vs total domains

- Counter of approved/rejected/pending domains

- Estimated articles that will be processed

- Time remaining estimate based on current pace

Sample URL Display:

- Show actual article titles when available

- Highlight keywords relevant to target audience

- Display URL structure to assess content quality

- Show publication dates if extracted

- Flag potential affiliate/marketing URLs

DECISION PERSISTENCE:

- Save decisions to JSON configuration file

- Auto-save every decision to prevent data loss

- Export approved domains to whitelist file

- Generate rejected domains report with reasons

- Create review queue for uncertain domains

VISUAL DESIGN:

- Clean, minimal interface focused on decision-making

- Color coding: Green (approved), Red (rejected), Yellow (review)

- Responsive design for different screen sizes

- Easy-to-read typography and clear action buttons

- Visual feedback for all user actions

INTEGRATION FEATURES:

Configuration Export:

- Generate `approved\_domains.txt` for article scraper

- Create `domain\_whitelist.json` with metadata

- Export `filtering\_decisions.csv` for audit trail

- Update main extraction script configuration

Smart Recommendations:

- Pre-select likely valuable domains (financial, educational)

- Flag suspicious domains (high tracking, affiliate patterns)

- Suggest review for mixed-content domains

- Highlight culturally relevant sources for target audience

FLASK APPLICATION STRUCTURE:

# Basic structure for the Flask app

from flask import Flask, render\_template, request, jsonify

import json

import pandas as pd

app = Flask(\_\_name\_\_)

@app.route('/')

def domain\_approval\_dashboard():

"""Main dashboard for domain approval"""

# Load domain analysis data

# Render approval interface

pass

@app.route('/api/approve\_domain', methods=['POST'])

def approve\_domain():

"""Approve a domain for inclusion"""

# Save approval decision

# Return updated stats

pass

@app.route('/api/reject\_domain', methods=['POST'])

def reject\_domain():

"""Reject a domain from inclusion"""

# Save rejection decision

# Return updated stats

pass

@app.route('/api/bulk\_approve', methods=['POST'])

def bulk\_approve\_category():

"""Approve all domains in a category"""

# Bulk approval logic

pass

@app.route('/api/export\_decisions', methods=['GET'])

def export\_decisions():

"""Export all approval decisions"""

# Generate configuration files

pass

HTML TEMPLATE FEATURES:

Domain Approval Dashboard:

- Header with progress statistics

- Search and filter controls

- Grid of domain cards

- Bulk action buttons

- Export and save controls

Domain Card Template:

- Domain name and logo (if available)

- Key statistics (URL count, category)

- Sample URL previews with titles

- Action buttons (large, clear)

- Visual status indicators

Keyboard Shortcuts:

- A: Approve current domain

- R: Reject current domain

- S: Skip to review later

- ↑/↓: Navigate between domains

- Enter: Confirm action

- U: Undo last action

ADVANCED FEATURES: Machine Learning Assistance:

* Suggest approval likelihood based on domain patterns
* Learn from user decisions to improve recommendations
* Identify similar domains for batch processing
* Flag outliers that need special attention

Cultural Relevance Assessment:

* Highlight domains with African American focus
* Flag content relevant to target age group (25-35)
* Identify career/professional development sources
* Mark financial education vs commercial content

Quality Indicators:

* Domain authority scores (if API available)
* Content freshness indicators
* Editorial vs affiliate content detection
* Mobile-friendly content assessment

WORKFLOW INTEGRATION:

* Start interface after domain analysis completes
* Auto-load previous decisions if resuming session
* Export approved list directly to article scraper config
* Generate summary report of filtering decisions

ERROR HANDLING:

* Graceful han