🤖 Automated Scanner Setup Guide

📋 Overview

This guide helps you set up **fully automated daily scanning** that:

- ✅ Runs automatically after market close (3:45 PM)
- ✅ Analyzes stocks using your existing scanners
- ✅ Emails you BUY/WAIT/SKIP decisions for tomorrow
- ✅ Handles errors and retries automatically

🚀 Quick Start (5 Minutes)

Step 1: Install Required Package

```bash
pip install schedule
```

Step 2: Configure Email (Gmail Example)

1. **Enable 2-Factor Authentication** on your Gmail account
2. **Generate App Password**:
   - Go to: https://myaccount.google.com/apppasswords
   - Select "Mail" and "Other (Custom name)"
   - Name it "Scanner Bot"
   - Copy the 16-character password
3. **Edit Configuration** in `automated_daily_scanner.py`:

```python
class Config:
    # EMAIL SETTINGS
    EMAIL_FROM = 'your_email@gmail.com'
    EMAIL_PASSWORD = 'your_16_char_app_password'  # From step 2
    EMAIL_TO = ['your_email@gmail.com']  # Can add multiple emails
```

Step 3: Test It

```bash
python automated_daily_scanner.py
```

Choose **[1] Run scan NOW** to test immediately.

If you get BUY signals, check your email! 📧

⚙️ Configuration Options

Stock Universe

```python
STOCK_UNIVERSE = 'Nifty_50.csv'  # Options:
# - Nifty_50.csv
# - Nifty_Next_50.csv
# - Mid_Cap_Stocks.csv
# - Small_Cap_Stocks.csv
```

Signal Quality

```python
MIN_GRADE = 'A'  # Options:
# - 'A+' (only highest quality, fewer signals)
# - 'A'  (recommended balance)
# - 'B+' (more opportunities)
```

Timing

```python
SCAN_START_TIME = '15:45'  # Run 15 min after market close
```

Notifications

```python
```

```
SEND_BUY_ONLY = True  # Only email when BUY signals found
ATTACH_CSV = True     # Include detailed CSV report
```

🖥️ Running Methods

Method 1: Keep Terminal Open (Simplest)

```bash
python automated_daily_scanner.py
# Select [2] Start scheduler
# Leave terminal running
```

Pros: Easy to set up, see logs in real-time

Cons: Terminal must stay open, computer must be on

Method 2: Windows Task Scheduler (Recommended)

Set it and forget it - runs even if you restart your PC

Setup Steps:

1. **Create a batch file** `run_scanner.bat`:

```batch
@echo off
cd C:\path\to\your\scanner\folder
python automated_daily_scanner.py --auto
```

2. **Open Task Scheduler**:
   - Press `Win + R`, type `taskschd.msc`, press Enter

3. **Create Basic Task**:
   - Name: "Stock Scanner"
   - Trigger: Daily at 3:45 PM
   - Action: Start a program
   - Program: `C:\path\to\run_scanner.bat`

4. **Configure**:
   - ✅ Run whether user is logged on or not
   - ✅ Run with highest privileges
   - ✅ Wake computer to run task (if laptop)

Done! It will run automatically every weekday.

Method 3: Linux/Mac (Cron Job)

1. **Edit crontab**:

```bash
crontab -e
```

2. **Add this line** (runs at 3:45 PM, Mon-Fri):

```bash
45 15 * * 1-5 cd /path/to/scanner && /usr/bin/python3 automated_daily_scanner.py --auto
```

3. **Save and exit**

Method 4: Cloud Server (24/7 Reliability)

Deploy to a VPS (like AWS, DigitalOcean, Linode):

```bash
# Install dependencies
pip install -r requirements.txt

# Run with nohup (survives logout)
nohup python automated_daily_scanner.py --scheduler &

# Or use systemd service (more robust)
sudo systemctl enable scanner.service
sudo systemctl start scanner.service
```

Email Setup (Alternative Services)

Gmail (Recommended for Personal Use)

```python
SMTP_SERVER = 'smtp.gmail.com'
SMTP_PORT = 587
EMAIL_FROM = 'your_email@gmail.com'
EMAIL_PASSWORD = 'your_app_password'
```

Outlook/Hotmail

```python
SMTP_SERVER = 'smtp-mail.outlook.com'
SMTP_PORT = 587
EMAIL_FROM = 'your_email@outlook.com'
EMAIL_PASSWORD = 'your_password'
```

Yahoo Mail

```python
SMTP_SERVER = 'smtp.mail.yahoo.com'
SMTP_PORT = 587
EMAIL_FROM = 'your_email@yahoo.com'
EMAIL_PASSWORD = 'your_app_password'
```

Custom SMTP (e.g., SendGrid, Mailgun)

```python
SMTP_SERVER = 'smtp.sendgrid.net'
SMTP_PORT = 587
EMAIL_FROM = 'apikey'
EMAIL_PASSWORD = 'your_sendgrid_api_key'
```

🔍 Monitoring & Logs

View Logs

```bash
# Real-time log monitoring
tail -f automated_scanner.log

# Last 50 lines
tail -n 50 automated_scanner.log

# Search for errors
grep "ERROR" automated_scanner.log
```

Log File Location

- Same folder as `automated_daily_scanner.py`
- File: `automated_scanner.log`
- Includes timestamps, status, errors

🛠️ Troubleshooting

"❌ Could not import one_click_entry_system"

Fix: All three files must be in same folder:

```
your_project/
├── automated_daily_scanner.py
├── one_click_entry_system.py
├── institutional_scannerV2.py
└── resources/
    └── Nifty_50.csv
```

"❌ Failed to send email: Authentication failed"

Fix Gmail:

1. Enable 2-Factor Authentication
2. Generate App Password (not your regular password)

3. Use 16-character app password in config

Fix Other Email:

1. Check if "Less secure apps" access is enabled
2. Verify SMTP server and port are correct

" ⏰ Market hasn't closed yet"

Normal behavior - Scanner checks timing automatically.

To override (for testing):

```
python

# In Config class, comment out:
# MARKET_CLOSE_TIME = '15:30'
```

" ⚠️ No signals found"

This is normal! Not every day has BUY signals.

To get more signals:

- Lower `MIN_GRADE` to `'B+'`
- Expand `STOCK_UNIVERSE` to Mid Cap or Small Cap
- Scanner only triggers on strong institutional setups

📱 Advanced: WhatsApp/Telegram Notifications

Option 1: IFTTT (Easiest)

1. Create IFTTT account
2. Connect email → Telegram/WhatsApp
3. Scanner emails → IFTTT → Your phone

Option 2: Telegram Bot (Better)

Add this to `automated_daily_scanner.py`:

```python
import requests

def send_telegram(message):
    BOT_TOKEN = 'your_bot_token'
    CHAT_ID = 'your_chat_id'
    url = f'https://api.telegram.org/bot{BOT_TOKEN}/sendMessage'
    requests.post(url, json={'chat_id': CHAT_ID, 'text': message})

# In send_notifications():
if len(buy_stocks) > 0:
    send_telegram(f"🟢 {len(buy_stocks)} BUY signals found!")
```

🎯 Usage Scenarios

Scenario 1: Daily Trader (Most Common)

```python
STOCK_UNIVERSE = 'Nifty_50.csv'
MIN_GRADE = 'A'
SCAN_START_TIME = '15:45'
SEND_BUY_ONLY = True
```

Result: Get email only when high-quality BUY signals appear

Scenario 2: Swing Trader (More Opportunities)

```python
STOCK_UNIVERSE = 'Mid_Cap_Stocks.csv'
MIN_GRADE = 'B+'
SCAN_START_TIME = '16:00'
SEND_BUY_ONLY = False
```

Result: Daily email with all signals, including WAIT/SKIP

Scenario 3: Conservative (Quality Over Quantity)

```python
STOCK_UNIVERSE = 'Nifty_50.csv'
MIN_GRADE = 'A+'
SCAN_START_TIME = '15:45'
SEND_BUY_ONLY = True
```

Result: Only get absolute best Grade A+ signals

📊 Sample Email Output

Subject: 🟢 3 BUY Signal(s) for October 22

Daily Trading Signals - October 21, 2025
Post-Market Analysis Complete | 3 BUY Signal(s) Found

#1. RELIANCE
Grade: A+ | Score: 92/100 | Phase: D
Current Price: ₹2,465.00 (as of 2025-10-21)
Price Change: +0.6% from signal

📊 ENTRY PLAN FOR TOMORROW:
- Entry Range: ₹2,440 - ₹2,490
- Stop Loss: ₹2,380 (Risk: 3.4%)
- Target 1: ₹2,635 (+6.9%)
- Target 2: ₹2,720 (+10.3%)

Why BUY:
✓ Grade A+ (highest quality)