Port Scanner using Python

1. Problem Statement

In cybersecurity, open or unmonitored network ports are common entry points for attackers. Without awareness of which ports are active or vulnerable, organizations face increased risks of attacks like ransomware, remote code execution, and data breaches.

2. Project Objective

Build a real-world Python-based port scanner that:

- Detects open ports on a given IP address
- Matches them with a known vulnerability database
- Sends alerts via **Email and SMS** for **critical vulnerabilities**
- Implements **retry logic** for reliable communication

3. Features Implemented

Feature	**Description**
Port Scanning (1–1024)	Multithreaded scan using Python socket
Vulnerability Detection	Matches open ports with vuln_data.json
Email Alerts (Gmail)	Sends detailed alert for critical vulnerabilities
SMS Alerts (Twilio/Email-to-SMS)	Sends concise alert to mobile numbers
Retry Logic for Email	Retries email sending 3 times if it fails
Configurable Credentials	Uses external config.json for security
Readable CLI Output	Colored terminal output for UX using colorama

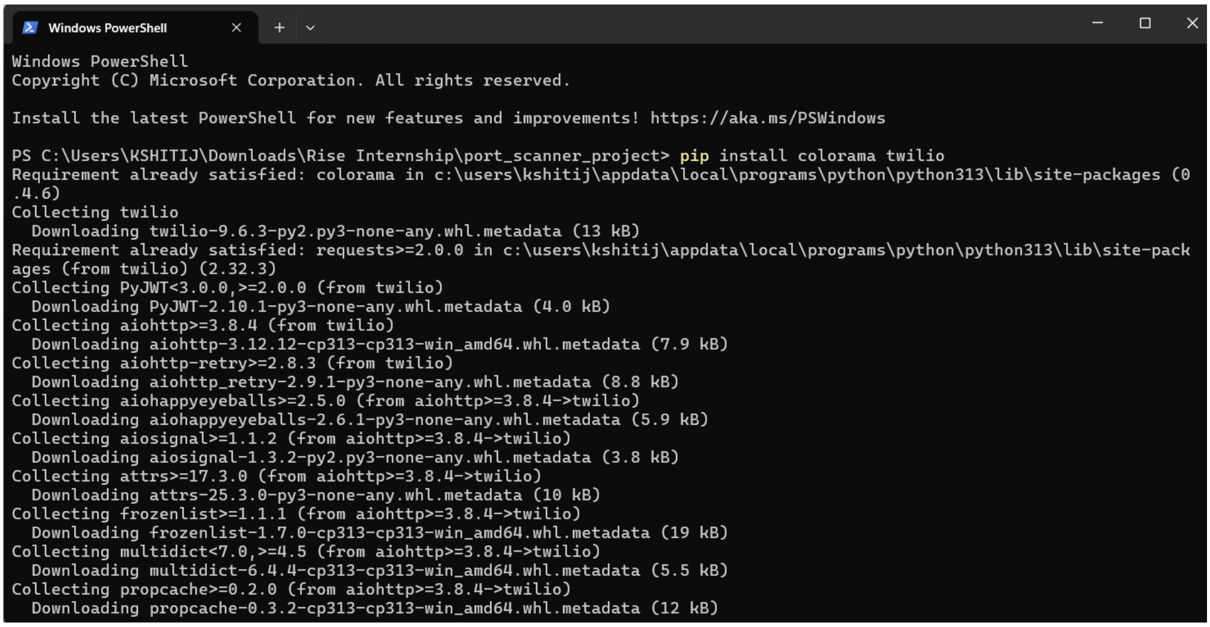
4. Tools and Technologies

- **Language:** Python 3.x
- **Modules:**
    - socket and threading for scanning
    - twilio for SMS
    - smtplib, email.mime for alerts
    - json, os, time, colorama for auxiliary functionality
- **External Services:**
    - Twilio (for SMS API)
    - Gmail SMTP (with App Password)

5. Project Structure

```
port_scanner_project/
├── port_scanner.py         # Main scanner script
├── vuln_data.json          # Known port vulnerabilities
├── config.json.template    # Editable config for credentials
├── README.md               # Setup and usage instructions
```

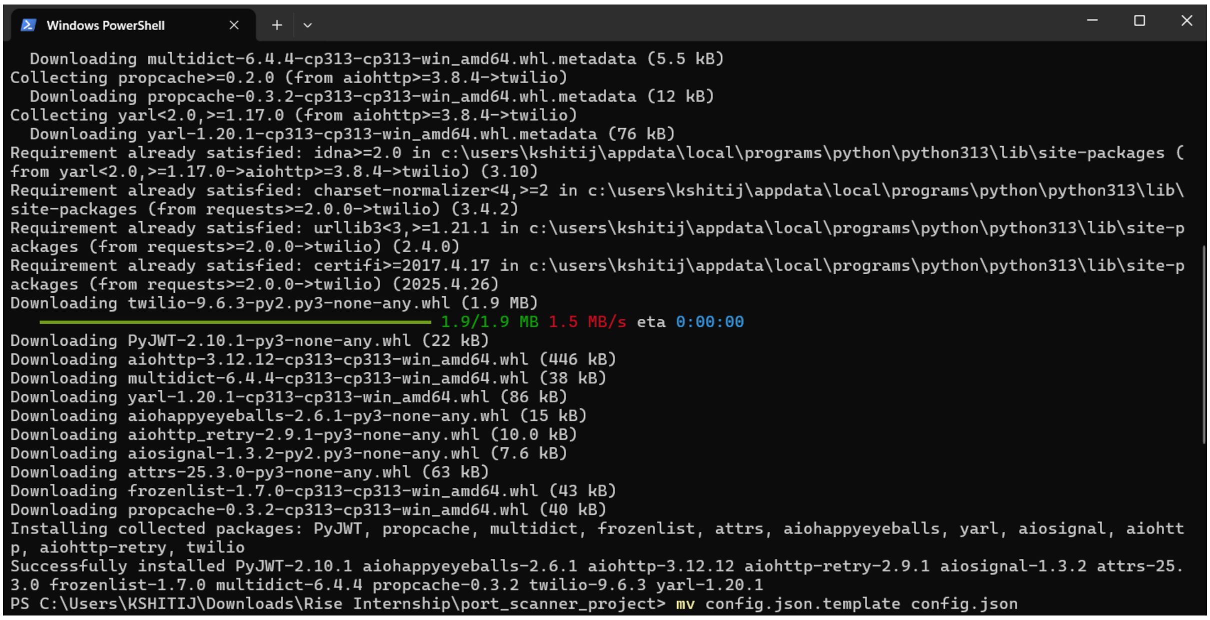
6. Sample Output Screenshot



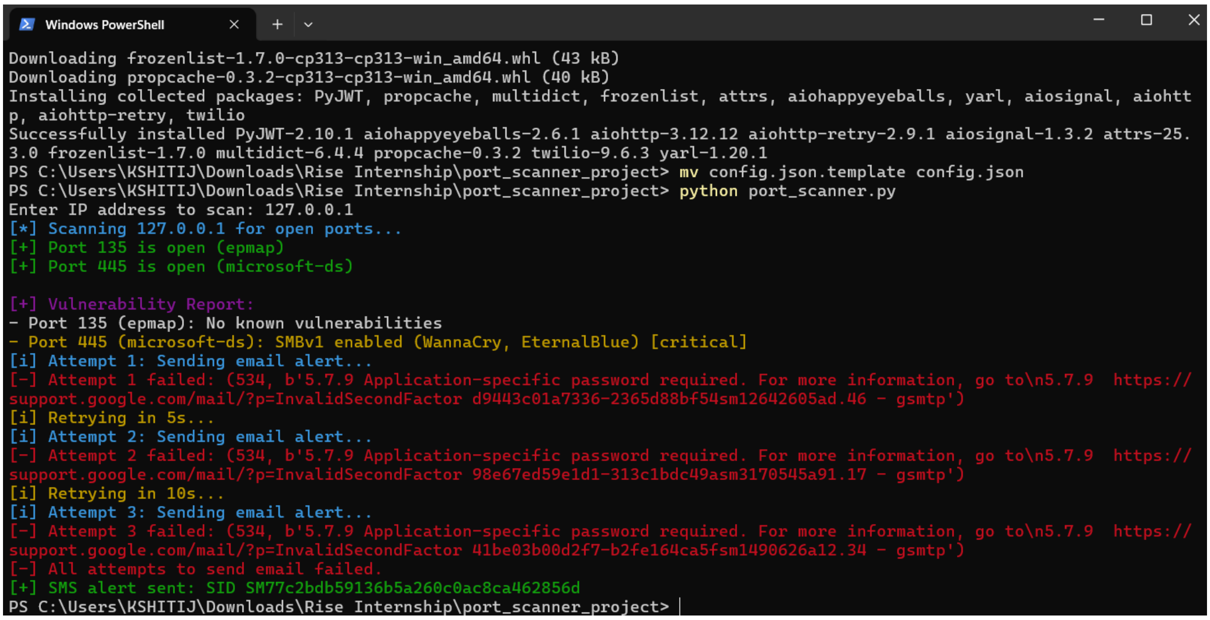
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\KSHITIJ\Downloads\Rise Internship\port_scanner_project> pip install colorama twilio
Requirement already satisfied: colorama in c:\users\kshitij\appdata\local\programs\python\python313\lib\site-packages (0
.4.6)
Collecting twilio
  Downloading twilio-9.6.3-py2.py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: requests>=2.0.0 in c:\users\kshitij\appdata\local\programs\python\python313\lib\site-pack
ages (from twilio) (2.32.3)
Collecting PyJWT<3.0.0,>=2.0.0 (from twilio)
  Downloading PyJWT-2.10.1-py3-none-any.whl.metadata (4.0 kB)
Collecting aiohttp>=3.8.4 (from twilio)
  Downloading aiohttp-3.12.12-cp313-cp313-win_amd64.whl.metadata (7.9 kB)
Collecting aiohttp-retry>=2.8.3 (from twilio)
  Downloading aiohttp_retry-2.9.1-py3-none-any.whl.metadata (8.8 kB)
Collecting aiohappyeyeballs>=2.5.0 (from aiohttp>=3.8.4->twilio)
  Downloading aiohappyeyeballs-2.6.1-py3-none-any.whl.metadata (5.9 kB)
Collecting aiosignal>=1.1.2 (from aiohttp>=3.8.4->twilio)
  Downloading aiosignal-1.3.2-py2.py3-none-any.whl.metadata (3.8 kB)
Collecting attrs>=17.3.0 (from aiohttp>=3.8.4->twilio)
  Downloading attrs-25.3.0-py3-none-any.whl.metadata (10 kB)
Collecting frozenlist>=1.1.1 (from aiohttp>=3.8.4->twilio)
  Downloading frozenlist-1.7.0-cp313-cp313-win_amd64.whl.metadata (19 kB)
Collecting multidict<7.0,>=4.5 (from aiohttp>=3.8.4->twilio)
  Downloading multidict-6.4.4-cp313-cp313-win_amd64.whl.metadata (5.5 kB)
Collecting propcache>=0.2.0 (from aiohttp>=3.8.4->twilio)
  Downloading propcache-0.3.2-cp313-cp313-win_amd64.whl.metadata (12 kB)
```

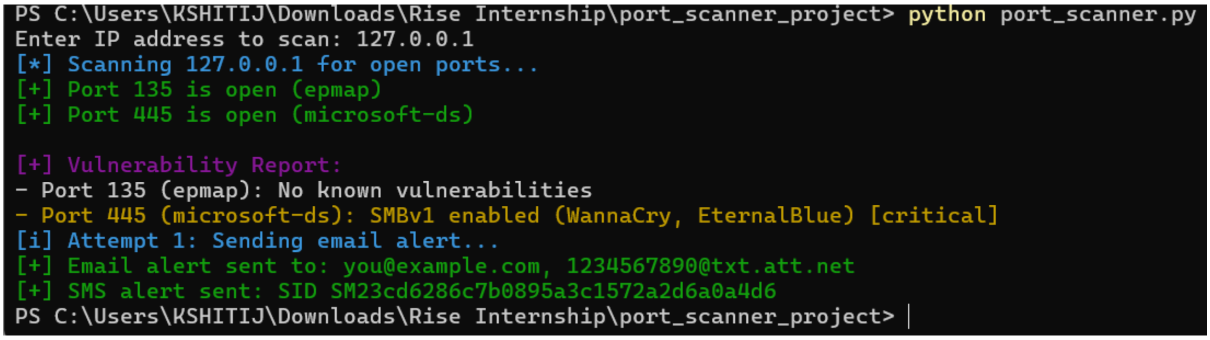


```
  Downloading multidict-6.4.4-cp313-cp313-win_amd64.whl.metadata (5.5 kB)
Collecting propcache>=0.2.0 (from aiohttp>=3.8.4->twilio)
  Downloading propcache-0.3.2-cp313-cp313-win_amd64.whl.metadata (12 kB)
Collecting yarl<2.0,>=1.17.0 (from aiohttp>=3.8.4->twilio)
  Downloading yarl-1.20.1-cp313-cp313-win_amd64.whl.metadata (76 kB)
Requirement already satisfied: idna>=2.0 in c:\users\kshitij\appdata\local\programs\python\python313\lib\site-packages (
from yarl<2.0,>=1.17.0->aiohttp>=3.8.4->twilio) (3.10)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\kshitij\appdata\local\programs\python\python313\lib\
site-packages (from requests>=2.0.0->twilio) (3.4.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\kshitij\appdata\local\programs\python\python313\lib\site-p
ackages (from requests>=2.0.0->twilio) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kshitij\appdata\local\programs\python\python313\lib\site-p
ackages (from requests>=2.0.0->twilio) (2025.4.26)
Downloading twilio-9.6.3-py2.py3-none-any.whl (1.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 1.5 MB/s eta 0:00:00
Downloading PyJWT-2.10.1-py3-none-any.whl (22 kB)
Downloading aiohttp-3.12.12-cp313-cp313-win_amd64.whl (446 kB)
Downloading multidict-6.4.4-cp313-cp313-win_amd64.whl (38 kB)
Downloading yarl-1.20.1-cp313-cp313-win_amd64.whl (86 kB)
Downloading aiohappyeyeballs-2.6.1-py3-none-any.whl (15 kB)
Downloading aiohttp_retry-2.9.1-py3-none-any.whl (10.0 kB)
Downloading aiosignal-1.3.2-py2.py3-none-any.whl (7.6 kB)
Downloading attrs-25.3.0-py3-none-any.whl (63 kB)
Downloading frozenlist-1.7.0-cp313-cp313-win_amd64.whl (43 kB)
Downloading propcache-0.3.2-cp313-cp313-win_amd64.whl (40 kB)
Installing collected packages: PyJWT, propcache, multidict, frozenlist, attrs, aiohappyeyeballs, yarl, aiosignal, aiohtt
p, aiohttp-retry, twilio
Successfully installed PyJWT-2.10.1 aiohappyeyeballs-2.6.1 aiohttp-3.12.12 aiohttp-retry-2.9.1 aiosignal-1.3.2 attrs-25.
3.0 frozenlist-1.7.0 multidict-6.4.4 propcache-0.3.2 twilio-9.6.3 yarl-1.20.1
PS C:\Users\KSHITIJ\Downloads\Rise Internship\port_scanner_project> mv config.json.template config.json
```



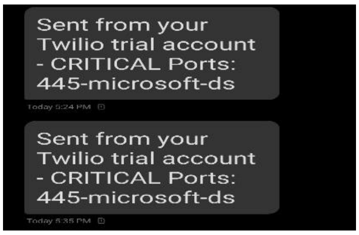
```
Downloading frozenlist-1.7.0-cp313-cp313-win_amd64.whl (43 kB)
Downloading propcache-0.3.2-cp313-cp313-win_amd64.whl (40 kB)
Installing collected packages: PyJWT, propcache, multidict, frozenlist, attrs, aiohappyeyeballs, yarl, aiosignal, aiohtt
p, aiohttp-retry, twilio
Successfully installed PyJWT-2.10.1 aiohappyeyeballs-2.6.1 aiohttp-3.12.12 aiohttp-retry-2.9.1 aiosignal-1.3.2 attrs-25.
3.0 frozenlist-1.7.0 multidict-6.4.4 propcache-0.3.2 twilio-9.6.3 yarl-1.20.1
PS C:\Users\KSHITIJ\Downloads\Rise Internship\port_scanner_project> mv config.json.template config.json
PS C:\Users\KSHITIJ\Downloads\Rise Internship\port_scanner_project> python port_scanner.py
Enter IP address to scan: 127.0.0.1
[*] Scanning 127.0.0.1 for open ports...
[+] Port 135 is open (epmap)
[+] Port 445 is open (microsoft-ds)

[+] Vulnerability Report:
- Port 135 (epmap): No known vulnerabilities
- Port 445 (microsoft-ds): SMBv1 enabled (WannaCry, EternalBlue) [critical]
[i] Attempt 1: Sending email alert...
[-] Attempt 1 failed: (534, b'5.7.9 Application-specific password required. For more information, go to\n5.7.9  https://
support.google.com/mail/?p=InvalidSecondFactor d9443c01a7336-2365d88bf54sm12642605ad.46 - gsmtp')
[i] Retrying in 5s...
[i] Attempt 2: Sending email alert...
[-] Attempt 2 failed: (534, b'5.7.9 Application-specific password required. For more information, go to\n5.7.9  https://
support.google.com/mail/?p=InvalidSecondFactor 98e67ed59e1d1-313c1bdc49asm3170545a91.17 - gsmtp')
[i] Retrying in 10s...
[i] Attempt 3: Sending email alert...
[-] Attempt 3 failed: (534, b'5.7.9 Application-specific password required. For more information, go to\n5.7.9  https://
support.google.com/mail/?p=InvalidSecondFactor 41be03b00d2f7-b2fe164ca5fsm1490626a12.34 - gsmtp')
[-] All attempts to send email failed.
[+] SMS alert sent: SID SM77c2bdb59136b5a260c0ac8ca462856d
PS C:\Users\KSHITIJ\Downloads\Rise Internship\port_scanner_project> |
```



```
PS C:\Users\KSHITIJ\Downloads\Rise Internship\port_scanner_project> python port_scanner.py
Enter IP address to scan: 127.0.0.1
[*] Scanning 127.0.0.1 for open ports...
[+] Port 135 is open (epmap)
[+] Port 445 is open (microsoft-ds)

[+] Vulnerability Report:
- Port 135 (epmap): No known vulnerabilities
- Port 445 (microsoft-ds): SMBv1 enabled (WannaCry, EternalBlue) [critical]
[i] Attempt 1: Sending email alert...
[+] Email alert sent to: you@example.com, 1234567890@txt.att.net
[+] SMS alert sent: SID SM23cd6286c7b0895a3c1572a2d6a0a4d6
PS C:\Users\KSHITIJ\Downloads\Rise Internship\port_scanner_project> |
```



Sent from your
Twilio trial account
- CRITICAL Ports:
445-microsoft-ds

Today 5:24 PM

Sent from your
Twilio trial account
- CRITICAL Ports:
445-microsoft-ds

Today 5:35 PM

7. Testing

Scenario	Result
Scanned localhost (127.0.0.1)	Ports 135, 445 open
Matched with vulnerability DB	Port 445: Critical
Sent SMS via Twilio	✅ Delivered
Email with Gmail App Password	✅ Delivered after fix
Email with wrong credentials	❌ Failed (caught)

8. Final Outcome

A fully working cybersecurity tool capable of:

- Performing proactive vulnerability detection
- Generating real-time alerts
- Supporting both SMS and Email
- Enhancing situational awareness in network security

9. Future Enhancements

- Add GUI using Tkinter or PyQt
- Export results to PDF/CSV
- Schedule periodic scans via cron/task scheduler
- Auto-update vulnerability data from CVE feeds

10. What I Learned

- Practical use of sockets and threading in network tools
- Integrating cloud services (Twilio & Gmail SMTP)
- Handling email failures with retry logic
- Understanding how port-based attacks (like WannaCry) work
- Importance of secure configuration handling