Complete System Components

1. **Database Schema** (`database_setup.sql`) - PostgreSQL tables for packets, alerts, and attack signatures with proper indexing
2. **Database Connection Module** (`database.py`) - Handles all database operations with connection pooling, packet storage, alert management, and statistics retrieval
3. **Packet Capture & Detection** (`packet_capture.py`) - Uses Scapy for real-time packet capture with attack detection algorithms for:
   - Port scanning
   - SYN flood attacks
   - Brute force attempts
   - ARP spoofing
4. **REST API Server** (`api_server.py`) - Flask-based API with comprehensive endpoints for:
   - Packet retrieval with filtering
   - Alert management
   - Traffic statistics
   - Advanced search capabilities
   - Node monitoring
5. **Web Dashboard** (`dashboard.html`) - Real-time visualization interface featuring:
   - Live traffic statistics
   - Security alert monitoring
   - Interactive charts (Chart.js)
   - Packet filtering and search
   - Auto-refresh capabilities
6. **System Orchestrator** (`main.py`) - Centralized control for running distributed nodes with configuration management
7. **Dependencies** (`requirements.txt`) - All required Python packages
8. **Complete Documentation** - Detailed setup, configuration, and usage instructions

Key Features Implemented

✅ **Real-time packet capture** using Scapy across multiple interfaces ✅ **Attack detection algorithms** with time-windowed counters ✅ **Centralized PostgreSQL database** with optimized schema ✅ **RESTful API** with comprehensive filtering and search ✅ **Modern web dashboard** with live updates and visualizations ✅ **Distributed architecture** supporting multiple capture nodes ✅ **Automatic maintenance** with data cleanup and retention ✅ **Comprehensive logging** and error handling

Quick Start

1. **Setup database:**

bash

```
sudo -u postgres createdb network_analyzer
psql -d network_analyzer < database_setup.sql
```

2. **Install dependencies:**

bash

```
pip install -r requirements.txt
```

3. **Run complete system:**

bash

```
sudo python3 main.py --enable-capture --enable-api --db-password your_password
```

4. **Access dashboard:**
   - Open browser to `http://localhost:5000`

The system is production-ready with proper security considerations, error handling, and scalability features. You can deploy it as a single node or distribute capture nodes across your network infrastructure.