Cybersecurity Class Audio Recording MCP Agent Architecture

System Overview

The MCP agent will be a modular Python-based system that automatically records, transcribes, and generates structured notes for your cybersecurity classes.

Core Architecture Components

1. Audio Recording Manager

```
┌─────────────────────────────────────┐
│           Recording Manager         │
├─────────────────────────────────────┤
│ • Session Detection                 │
│ • Audio Source Selection            │
│ • Quality Control                   │
│ • Metadata Collection               │
└─────────────────────────────────────┘
```

Features:

- **Presencial Classes**: External microphone recording with noise reduction
- **Online Classes**: System audio capture from browser/applications
- **Smart Recording**: Voice activity detection to avoid dead air
- **Session Metadata**: Auto-detect class type, timestamp, duration

2. Audio Processing Pipeline

```
┌─────────────────────────────────────┐
│         Processing Pipeline         │
├─────────────────────────────────────┤
│ • Whisper Transcription             │
│ • Speaker Diarization               │
│ • Audio Enhancement                 │
│ • Chunk Management                  │
└─────────────────────────────────────┘
```

Capabilities:

- **Local Whisper**: Privacy-focused transcription

- **Speaker Identification**: Distinguish between instructor and students
- **Technical Term Recognition**: Custom dictionary for cybersecurity terminology
- **Real-time Processing**: Process audio chunks during recording

3. Content Analysis Engine

```
┌─────────────────────────────────────┐
│        Content Analyzer             │
├─────────────────────────────────────┤
│ • Key Concept Extraction            │
│ • Topic Segmentation                │
│ • Action Item Detection             │
│ • Reference Link Extraction         │
└─────────────────────────────────────┘
```

Intelligence Features:

- **Cybersecurity Context**: Recognize tools, techniques, vulnerabilities
- **Concept Mapping**: Link related topics across classes
- **Assignment Detection**: Identify homework, projects, deadlines
- **Resource Extraction**: Capture mentioned URLs, tools, references

4. Obsidian Note Generator

```
┌─────────────────────────────────────┐
│         Note Generator              │
├─────────────────────────────────────┤
│ • Template System                   │
│ • Markdown Formatting               │
│ • Tag Management                    │
│ • Link Generation                   │
└─────────────────────────────────────┘
```

Directory Structure

```
~/CyberSec-Audio-MCP/
├── config/
│   ├── class_schedule.yaml
│   ├── audio_settings.yaml
│   └── obsidian_templates/
├── recordings/
│   ├── raw/
│   ├── processed/
│   └── transcripts/
├── agents/
│   ├── recording_agent.py
│   ├── processing_agent.py
│   ├── analysis_agent.py
│   └── obsidian_agent.py
├── models/
│   ├── whisper_model/
│   └── cybersec_terminology.json
└── vault_integration/
    └── generated_notes/
```

Note Template Structure

Class Note Template

```markdown
# {{class_name}} - {{date}}

## ▯ Session Info
- **Type**: {{class_type}} (Presencial/Online)
- **Duration**: {{duration}}
- **Recording Quality**: {{quality_score}}
- **Instructor**: {{instructor_name}}

## ▯ Key Topics Covered
{{extracted_topics}}

## ▯ Detailed Notes
{{structured_transcript}}

## ▯ Tools & Techniques Mentioned
{{technical_references}}

## ⚠ Security Concepts
{{security_concepts}}

## ▯ Action Items
{{action_items}}

## ▯ References & Links
{{extracted_links}}

## ▯ Tags
{{generated_tags}}
```

Technical Implementation

Core Dependencies

python

```
# Audio Processing
import pyaudio
import whisper
import pydub
from scipy.io import wavfile

# NLP & Analysis
import spacy
import nltk
from transformers import pipeline

# Obsidian Integration
import os
import yaml
from pathlib import Path

# MCP Framework
from mcp import AgentFramework
```

Configuration Management

yaml

```
# class_schedule.yaml
classes:
  - name: "Network Security"
    schedule: "Monday 09:00-11:00"
    type: "presencial"
    instructor: "Prof. Martinez"

  - name: "Ethical Hacking"
    schedule: "Wednesday 14:00-16:00"
    type: "online"
    platform: "Zoom"
```

Workflow Automation

1. Pre-Class Setup

- **Schedule Detection:** Auto-start recording based on class schedule
- **Audio Source Selection:** Switch between mic (presencial) and system audio (online)

- **Quality Check**: Verify audio levels and recording setup

2. During Class

- **Live Monitoring**: Real-time audio quality assessment
- **Chunk Processing**: Process audio in segments for better performance
- **Keyword Alerting**: Highlight important cybersecurity terms in real-time

3. Post-Class Processing

- **Full Transcription**: Complete audio-to-text conversion
- **Content Analysis**: Extract key concepts and structure
- **Note Generation**: Create formatted Obsidian notes
- **Integration**: Save to vault with proper linking

Advanced Features

Smart Content Recognition

- **CVE Detection**: Automatically format CVE references
- **Tool Mentions**: Create links to tool documentation
- **Concept Definitions**: Add explanations for technical terms
- **Timeline Tracking**: Maintain chronological learning progress

Integration Capabilities

- **Calendar Sync**: Integrate with class schedules
- **Obsidian Plugins**: Work with existing note-taking workflows
- **Export Options**: Generate summaries, flashcards, study guides

Privacy & Security

- **Local Processing**: All transcription happens locally
- **Data Encryption**: Protect sensitive class recordings
- **Access Control**: Secure access to generated notes

Deployment Strategy

Phase 1: Basic Recording & Transcription

- Set up audio recording for both class types

- Implement basic Whisper transcription
- Create simple note templates

Phase 2: Content Analysis

- Add cybersecurity terminology recognition
- Implement topic extraction
- Develop structured note generation

Phase 3: Advanced Intelligence

- Add speaker diarization
- Implement concept mapping
- Create study material generation

Phase 4: Full Automation

- Schedule-based auto-recording
- Real-time processing
- Advanced Obsidian integration

Expected Outcomes

- **Time Savings**: 70% reduction in manual note-taking
- **Improved Retention**: Structured, searchable notes
- **Better Review**: Timestamped audio references
- **Enhanced Learning**: Cross-class concept linking