

# Atlas Processor Authentication Guide

---

## Overview

---

AtlasIED Atmosphere audio processors typically require HTTP Basic Authentication to access their web interface and API. This guide explains how authentication has been implemented in the Sports Bar TV Controller system.

## What Changed

---

### Database Schema

Added two new fields to the `AudioProcessor` model:

- `username` (optional): HTTP Basic Auth username
- `password` (optional): HTTP Basic Auth password (stored encrypted)

### Authentication Features

1. **Credential Storage:** Usernames and passwords are securely stored in the database with basic encryption
2. **Auto-Detection:** System can automatically try common default credentials during connection testing
3. **Manual Configuration:** Users can manually enter credentials when adding or editing processors
4. **Connection Testing:** Enhanced connection testing that handles authentication requirements

## Default Credentials

---

Most Atlas processors ship with default credentials:

- **Username:** `admin`
- **Password:** `admin`

Alternative common passwords:

- Empty/blank password
- `password`
- `Admin`
- `1234`

**Security Note:** Always change default credentials in production environments!

## How to Use

---

### Adding a New Processor with Authentication

1. Navigate to the Audio Processor Manager
2. Click "Add Audio Processor"
3. Fill in the basic information (name, model, IP address)
4. In the **Authentication** section:
  - Enter username (default: `admin` )
  - Enter password (default: `admin` )

5. Click "Add Processor"




## Auto-Detection Feature

If you don't know the credentials:

1. Leave username and password fields with defaults or empty
2. Add the processor
3. Click "Test Connection"
4. The system will automatically try common credential combinations
5. If successful, credentials are saved automatically

## Testing Connection

When you test a connection:

-  **Success + Authenticated:** Processor connected with valid credentials
-  **Requires Authentication:** Processor needs credentials - add them in settings
-  **Connection Failed:** Check network connectivity and IP address

## API Changes

### POST /api/audio-processor

Now accepts optional `username` and `password` fields:

```
{
  "name": "Main Bar Audio",
  "model": "AZM4",
  "ipAddress": "192.168.1.100",
  "port": 80,
  "username": "admin",
  "password": "admin"
}
```

### POST /api/audio-processor/test-connection

Enhanced with authentication support:

```
{
  "processorId": "abc123",
  "ipAddress": "192.168.1.100",
  "port": 80,
  "username": "admin",
  "password": "admin",
  "autoDetectCredentials": true
}
```

Response includes authentication status:

```
{
  "connected": true,
  "authenticated": true,
  "message": "Successfully connected with credentials: admin"
}
```

## Security Considerations

### Current Implementation

- Passwords are base64 encoded (NOT encrypted)
- This provides basic obfuscation but is NOT secure

### Production Recommendations

1. **Use proper encryption:** Implement AES-256 encryption for passwords
2. **Environment variables:** Store encryption keys in environment variables
3. **HTTPS only:** Always use HTTPS for API communication
4. **Change defaults:** Never use default credentials in production
5. **Access control:** Implement role-based access control
6. **Audit logging:** Log all authentication attempts

### Upgrading Security

To implement proper encryption, replace the simple base64 encoding in `src/lib/atlas-auth.ts`:

```
import crypto from 'crypto'

const ENCRYPTION_KEY = process.env.ATLAS_ENCRYPTION_KEY! // 32 bytes
const IV_LENGTH = 16

export function encryptPassword(password: string): string {
  const iv = crypto.randomBytes(IV_LENGTH)
  const cipher = crypto.createCipheriv('aes-256-cbc', Buffer.from(ENCRYPTION_KEY), iv)
  let encrypted = cipher.update(password)
  encrypted = Buffer.concat([encrypted, cipher.final()])
  return iv.toString('hex') + ':' + encrypted.toString('hex')
}

export function decryptPassword(encrypted: string): string {
  const parts = encrypted.split(':')
  const iv = Buffer.from(parts.shift()!, 'hex')
  const encryptedText = Buffer.from(parts.join(':'), 'hex')
  const decipher = crypto.createDecipheriv('aes-256-cbc',
    Buffer.from(ENCRYPTION_KEY), iv)
  let decrypted = decipher.update(encryptedText)
  decrypted = Buffer.concat([decrypted, decipher.final()])
  return decrypted.toString()
}
```

## Troubleshooting

### “Authentication Required” Error

**Problem:** Processor requires credentials but none are configured

**Solution:**

1. Edit the processor settings
2. Add username and password (try `admin / admin` first)
3. Test connection again

## “Connection Failed” with Credentials

**Problem:** Credentials are incorrect

**Solution:**

1. Try default credentials: `admin / admin`
2. Try blank password with `admin` username
3. Check processor’s web interface directly in browser
4. Consult processor documentation for default credentials
5. Reset processor to factory defaults if necessary

## Auto-Detection Not Working

**Problem:** System can’t find working credentials

**Solution:**

1. Manually enter credentials you know work
2. Access processor web interface to verify credentials
3. Check if processor has custom credentials configured

## Migration Guide

---

### Existing Processors

Processors added before this update will not have credentials stored. To add them:

1. Edit each processor
2. Add username and password
3. Test connection to verify
4. Save changes

### Database Migration

Run the migration to add new fields:

```
npx prisma migrate dev --name add-atlas-authentication
```

Or apply manually:

```
ALTER TABLE "AudioProcessor" ADD COLUMN "username" TEXT;  
ALTER TABLE "AudioProcessor" ADD COLUMN "password" TEXT;
```

## Technical Details

---

### Authentication Flow

1. User adds processor with credentials
2. Credentials are encrypted and stored in database
3. When connecting, credentials are decrypted
4. HTTP Basic Auth header is created: `Authorization: Basic base64(username:password)`
5. Header is included in all requests to processor

### Files Modified

- `prisma/schema.prisma` - Added username/password fields

- `src/lib/atlas-auth.ts` - New authentication utilities
- `src/app/api/audio-processor/route.ts` - Handle credentials in CRUD operations
- `src/app/api/audio-processor/test-connection/route.ts` - Enhanced connection testing
- `src/components/AudioProcessorManager.tsx` - UI for credential input

## Support

---

For issues or questions:

1. Check processor documentation for default credentials
2. Verify network connectivity to processor
3. Test web interface access in browser
4. Review application logs for detailed error messages

## Future Enhancements

---

Planned improvements:

- [ ] Proper AES-256 encryption for passwords
- [ ] OAuth 2.0 support for cloud-connected processors
- [ ] Certificate-based authentication
- [ ] Multi-factor authentication
- [ ] Credential rotation policies
- [ ] Integration with secret management systems (HashiCorp Vault, AWS Secrets Manager)