BF2 Server Management API

A lightweight, secure REST API built with Rust and Axum for managing a Battlefield 2 (BF2) server running on Debian. This API provides controlled access to server restart functionality, configuration file uploads, and RCON command execution.

Features

- 🚀 **Server Management**: Restart BF2 server with different profiles
- 📁 **Config Upload**: Secure multipart file upload for server configurations
- 🎮 **RCON Integration**: Execute RCON commands with MD5 authentication
- 🔒 **Security First**: Token authentication, rate limiting, and systemd hardening
- ⚡ **Performance**: Async/await with connection pooling and caching
- 📊 **Monitoring**: Health checks, status reporting, and structured logging

API Endpoints

Public Endpoints

- `GET /health` - Health check (no authentication required)

Authenticated Endpoints

- `GET /status` - Server and RCON status information
- `POST /restart` - Restart BF2 server with specified profile
- `POST /configs/upload` - Upload server configuration files
- `POST /rcon/command` - Execute custom RCON commands
- `POST /rcon/users` - Get current player list (convenience endpoint)

Authentication

The API supports two authentication methods:
- **Authorization Header**: `Authorization: Bearer <token>`
- **Custom Header**: `X-API-Token: <token>`

Quick Start

Prerequisites

- Debian 12 (Bookworm) or compatible Linux system
- Rust (stable) and Cargo
- Existing BF2 server with RCON enabled
- systemd for service management

Installation

1. **Clone and build the project:**
   `bash git clone <repository-url> cd bf2-api cargo build --release`
2. **Install using the provided script:**
   `bash sudo ./install.sh`
3. **Configure the API:**
   `bash sudo vim /etc/bf2-api/config.toml`

Update the configuration with your specific settings:
```toml
[api]
bind = "127.0.0.1:8080"

[security]
token = "your-secure-api-token-here"

[rcon]
host = "127.0.0.1"
```

```
port = 4711
password = "your-rcon-password"
timeout_secs = 10

[paths]
restart_script = "/opt/bf2/scripts/restart-bf2.sh"
config_dir = "/home/bf2/server"
```

1. **Enable and start the service:** `bash sudo systemctl enable bf2-api sudo systemctl start bf2-api sudo systemctl status bf2-api`

Testing

Run the comprehensive test suite:

```
./test-api.sh --token "your-api-token"
```

Configuration

Configuration File (`/etc/bf2-api/config.toml`)

```
[api]
bind = "127.0.0.1:8080"          # API server bind address

[security]
token = "your-secret-token"       # Static API token
# allowlist = ["10.0.0.0/24"]    # Optional IP allowlist

[rcon]
host = "127.0.0.1"               # RCON server address
port = 4711                      # RCON port
password = "rcon-password"       # RCON password
timeout_secs = 10               # Connection timeout

[paths]
restart_script = "/opt/bf2/scripts/restart-bf2.sh"  # Path to restart script
config_dir = "/home/bf2/server"  # Config file upload directory
```

Environment File (`/etc/bf2-api/environment`)

```
CONFIG_PATH=/etc/bf2-api/config.toml
RUST_LOG=bf2_api=info,tower_http=info
BF2SERVERUSER=bf2
BF2_HOME=/home/bf2
```

API Usage Examples

Health Check

```
curl http://localhost:8080/health
```

Get Server Status

```
curl -H "Authorization: Bearer your-token" \
     http://localhost:8080/status
```

Restart Server

```
curl -X POST \
     -H "Authorization: Bearer your-token" \
     -H "Content-Type: application/json" \
     -d '{"profile": "vehicles", "map_name": "Strike_at_Karkand"}'
\
     http://localhost:8080/restart
```

Upload Configuration File

```
curl -X POST \
     -H "Authorization: Bearer your-token" \
     -F "file=@server.cfg" \
     http://localhost:8080/configs/upload
```

Execute RCON Command

```
curl -X POST \
     -H "Authorization: Bearer your-token" \
     -H "Content-Type: application/json" \
     -d '{"command": "exec admin.kickPlayer 3"}' \
     http://localhost:8080/rcon/command
```

Get Player List

```
curl -X POST \
     -H "Authorization: Bearer your-token" \
     http://localhost:8080/rcon/users
```

Security Features

systemd Hardening

The service runs with extensive security hardening:
- No new privileges
- Private temporary filesystem
- Read-only system directories
- Restricted network access
- Capability restrictions
- System call filtering

File Upload Security

- Filename allowlist (`.profile`, `.con`, `.cfg`)
- Path traversal prevention
- Atomic file replacement

- Automatic backup creation
- Proper file ownership

Rate Limiting

- 10 requests per 10 seconds per IP address
- Configurable via middleware

RCON Protocol Implementation

The API implements the BF2 RCON protocol with MD5 authentication:

1. Connect to RCON server (default port 4711)
2. Read banner: `### Digest seed: <seed>`
3. Compute MD5 hash: `md5(seed + password)`
4. Send login: `login <hash>`
5. Execute commands and read responses

Connection pooling with TTL ensures efficient resource usage.

Logging

The API uses structured logging with tracing:
- Request/response logging
- RCON operation logging
- Security event logging
- Error tracking with context

View logs using journalctl:

```
sudo journalctl -u bf2-api -f
```

Monitoring

Health Endpoint

The `/health` endpoint provides a simple liveness check for load balancers.

Status Endpoint

The `/status` endpoint provides detailed information:
- API service status
- RCON connectivity
- Player count
- Restart script availability

Troubleshooting

Common Issues

1. **RCON Connection Failed**
   - Verify BF2 server is running
   - Check RCON password in config
   - Ensure port 4711 is accessible
2. **Permission Denied for Script**
   - Ensure restart script is executable
   - Check file ownership and permissions
   - Verify bf2api user has execute access
3. **Config Upload Failed**
   - Check config directory permissions
   - Ensure bf2api user can write to target directory
   - Verify filename matches allowed extensions

4. **Service Won't Start**
   - Check configuration file syntax
   - Verify all paths exist
   - Review systemd logs: `journalctl -u bf2-api`

Security Analysis

Analyze the systemd security configuration:

```
sudo systemd-analyze security bf2-api
```

This should show a "good" security score with minimal attack surface.

Development

Building from Source

```
cargo build --release
```

Running Tests

```
cargo test
./test-api.sh
```

Code Structure

- `src/main.rs` - Application entry point and server setup
- `src/config/` - Configuration management
- `src/auth/` - Authentication and rate limiting middleware
- `src/rcon/` - RCON client implementation

- `src/handlers/` - API endpoint handlers
- `src/utils/` - Utility functions for file operations

License

This project is licensed under the MIT License - see the LICENSE file for details.

Contributing

1. Fork the repository
2. Create a feature branch
3. Make your changes with tests
4. Submit a pull request

Support

For issues and questions:
1. Check the troubleshooting section
2. Review the logs: `journalctl -u bf2-api`
3. Open an issue on GitHub with logs and configuration (redact sensitive information)

Security Note: Always change the default API token and RCON passwords in production deployments. Consider running the API behind a reverse proxy with TLS termination for external access.