RFC: [Number]

­­

[Protocol Name]

[Date]

Table of Contents

[1. Introduction 3](#_Toc160462398)

[2. Protocol Overview 3](#_Toc160462399)

[3. Protocol Specification 3](#_Toc160462400)

[3.1. Service Location 3](#_Toc160462401)

[3.2. Sequence of Inter-process Communication 3](#_Toc160462402)

[3.3. Representation and Interpretation of Data Exchanged 3](#_Toc160462403)

[3.4. Error Handling 4](#_Toc160462404)

[3.5. Service Session Management 4](#_Toc160462405)

[4. Security Considerations 4](#_Toc160462406)

[5. Glossary 4](#_Toc160462411)

[6. References 4](#_Toc160462412)

# 1. Introduction

The protocol is intended to provide secure and efficient interactions between players and the game server.

This document describes the [Protocol Name], a protocol designed for facilitating communication between clients and servers in a text-based game environment.

# 2. Protocol Overview

[Protocol Name] is a simple client-server protocol that enables players to log on to the game server, interact with the game world, and exchange messages securely.

# 3. Protocol Specification

## 3.1. Service Location

## Clients connect to the server's IP address and port number.

## Communication is secured using SSL/TLS encryption.

## 3.2. Sequence of Inter-process Communication

## Client initiates a secure connection request to the server

## Server listens for incoming client connections over SSL/TLS

## Upon successful connection establishment, client and server perform an SSL/TLS handshake

## Client acknowledges server's acknowledgment over the secure connection

## Communication continues over the established SSL/TLS connection until session termination.

## 3.3. Representation and Interpretation of Data Exchanged

Requests and responses follow a predefined format agreed upon by both client and server.

JSON data format is used for encoding game commands, responses, and other data over the secure connection.

## 3.4. Error Handling

## Error codes and messages are defined for common error scenarios.

Both client and server handle exceptions and errors appropriately over the secure connection.

## 3.5. Service Session Management

Upon connection establishment, a secure service session is initiated.

The server assigns a unique session identifier to each client session securely.

Session management includes maintaining session state and uploaded messages securely over the SSL/TLS connection.

Periodic updates and notifications are sent securely over the SSL/TLS connection.

Session clean-up is performed securely upon session termination.

# 4. Security Considerations

Communication between client and server is encrypted using SSL/TLS, providing confidentiality and integrity for sensitive data.

SSL/TLS certificate management is crucial for ensuring the authenticity of the server and preventing man-in-the-middle attacks.



# Functional Specifications

[ TODO - if applicable ]

## Header Format

## Terminology

## Sequence Numbers

## Establishing a connection

## Closing a connection

## Precedence and security

## Data communication

## Interfaces

## Event processing

# Glossary

[ if applicable ]

# References

[ if applicable ]

(basing layout and content on RFC:793

https://datatracker.ietf.org/doc/html/rfc793)