```
Network Working Group                              Sol Victoria Anselmo
Internet-Draft                        Santiago Andrés Larroudé Alvarez
Intended status: Experimental                Malena Vasquez Courrier
November 22, 2022. ITBA                                Andres Podgorny
```

MASS Management Protocol Version 1

Status of this Memo

This document specifies an experimental management protocol for the exchange of requests with a server and client based on the SOCK5 protocol [RFC 2118]. This protocol should help in server behavior modifications or server performance metrics retrieval.

1. Introduction

With the development of more and more communication services, there is a need to provide standard protocols to communicate with these services in a secure and transparent way.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119]. Unless otherwise noted, the decimal numbers appearing in packet format diagrams represent the length of the corresponding field, in octets. Where a given octet must take on a specific value, the syntax X'hh' is used to denote the value of the single octet in that field. When the word 'Variable' is used, it indicates that the corresponding field has a variable length defined either by an associated (one or two octet) length field, or by a data type field.

3. Procedure for clients

When a client wants to connect to a server provider of the MASS management protocol, it must open a TCP connection to an appropriate port on the server system. The service is by default located on TCP port 8080. If the connection request is successful, the client can be authenticated with the chosen method, and then send a request. The server evaluates the request, replies with the appropriate reply and closes the connection.

The client connects to the server, and sends a version identifier/ method selection message:

```
+-------+----------+----------+
|  VER  | NMETHODS | METHODS  |
+-------+----------+----------+
| X'01' |    1     | 1 to 255 |
+-------+----------+----------+
```

The VER field is set to X’01’ for this version of the protocol. The NMETHODS field contains the number of method identifier octets that appear in the METHODS field. The server selects from one of the methods given in METHODS, and sends a METHOD selection message:

VER	METHOD
X’01’	1

If the selected METHOD is X’FF’, none of the methods listed by the client are acceptable, and the client MUST close the connection. The values currently defined for METHOD are:

- o X'00' NO AUTHENTICATION REQUIRED
- o X'02' USERNAME/PASSWORD
- o X'FF' NO ACCEPTABLE METHODS

The client and server then enter a method-specific negotiation. Descriptions of the method-dependent negotiations appear in separate memos. The method USERNAME/PASSWORD negotiation is described in [RFC 1929]. Compliant implementations SHOULD support USERNAME/PASSWORD authentication methods.

4. Requests

Once the negotiation was completed, the client sends the request details. The SOCKS is formed as follows:

REQUEST INDEX	PARAMETERS
1	Variable

- o REQUEST INDEX: refers to the selection between the actions and data retrievals the server offer. When REQUEST INDEX is X’00’ server MUST reply with a list of supported indexes.
- o PARAMETERS: refers to the needed parameters to accomplish the request. The quantity, size and content of the parameters are directly dependant of the REQUEST INDEX, this information will be detailed below. Parameters have fixed or variable length. In case of a variable length, the parameter MUST follow the next structure:

LENGTH	DATA
1	Variable (up to 255)

Where LENGTH contains the number of octets of the DATA that follow, there is no terminating NULL octet. And DATA contains the parameter information. The server will evaluate the request based on the REQUEST INDEX and PARAMETERS, and will return one reply message with the appropriate data.

Information retrieval indexes:

- X’00’ A list of all the REQUEST INDEX that the server supports.
- X’01’ List of active users on the server.
- X’02’ Amount of historic connections to the server.
- X’03’ Amount of actual concurrent connections to the server.
- X’04’ Max concurrent connections to the server.
- X’05’ Amount of historic byte transferred by the server.
- X’06’ Amount of historic authentication attempts to the server.
- X’07’ Amount of historic connections attempts to the server.
- X’08’ Average bytes per single server read.
- X’09’ Average bytes per single server write.

Behavior modification indexes:

- X’0A’ Add a new user. PARAMETERS:
  Content: Username. Type: ASCII characters. Size: Variable.
  Content: Password. Type: ASCII characters. Size: Variable.
- X’0B’ Delete a user. PARAMETERS:
  Content: Username. Type: ASCII characters. Size: Variable.
- X’0C’ Disable server authentication. PARAMETERS:
  Content: New authentication status. Size: 1 octet. Where X’00’ indicates to turn on authentication and any other value means to turn authentication off.
- X’0D’ Disable password dissectors. PARAMETERS:
  Content: New password dissectors status. Size: 1 octet. Where X’00’ indicates to turn on password dissectors and any other value means to turn password dissectors off. Server MUST support the index X’00’. The rest of the indexes are not compulsory, but RECOMMENDED.

5. Replies

The reply will have a STATUS and fields related to the request of the client. The fields, if any, will contain the main reply information. The size and content of the fields are directly dependant of the REQUEST INDEX, this information will be detailed below. Fields have fixed or variable length. In case of a variable length, it should follow the next structure: The reply MUST follow:

STATUS	FIELDS
1	Variable

LENGTH	DATA
1	Variable (up to 255)

Where LENGTH contains the number of octets of the DATA that follow, there is no terminating NUL octet. The MASS management server MUST terminate the TCP connection shortly after sending the reply. STATUS describes the return value of the server related to a request.
The possible status are:

- X'00': Success
- X'01': Server error
- X'02': Index not supported
- X'03': Failed to create a new user because server reached the maximum number of users

In case of a failure status, server must reply with the status. The MASS management server MUST terminate the TCP connection shortly after sending the reply. In case of success status, the fields on the reply directly depend of the REQUEST INDEX.

Information retrieval indexes:

- X'00': Number of fields: 1. FIELDS:
  - Content: List of supported indexes. Type: byte array Size: Variable. Each byte of the data represents a single supported index. Possible return status: X'00', X'01'.
- X'01': Number of fields: Variable. This is a special case where the first field will be used to inform the amount of fields that follow. FIELDS:
  - Content: Amount of fields that follow. Each of these fields represent a user. Type: byte. Size: 1 octet.
  - Content: Username. Type: ASCII characters. Size: Variable.
- X'02': Number of fields: 1. FIELDS:
  - Content: Historic connections to the server. Type: unsigned int. Size: 4 octets.
- X'03': Number of fields: 1. FIELDS:
  - Content: Actual concurrent connections. Type: unsigned int. Size: 4 octets.
- X'04': Number of fields: 1. FIELDS:
  - Content: Max concurrent connections. Type: unsigned int. Size: 4 octets.
- X'05': Number of fields: 1. FIELDS:
  - Content: Historic byte transferred. Type: unsigned int. Size: 4 octets.
- X'06': Number of fields: 1. FIELDS:
  - Content: Historic authentication attempts. Type: unsigned int. Size: 4 octets.

- o X’07’: Number of fields: 1. FIELDS:
  Content: Historic connections attempts. Type: unsigned int. Size: 4 octets.
- o X’08’: Number of fields: 1. FIELDS:
  Content: Average byte size per server read. Type: unsigned int. Size: 4 octets.
- o X’09’: Number of fields: 1. FIELDS:
  Content: Average byte size per server write. Type: unsigned int. Size: 4 octets.

Behavior modification indexes:

- o X’0A’ Number of fields: 0. Possible return status: X’00’, X’01’, X’02’, X’03’.
- o X’0B’ Number of fields: 0.
- o X’0C’ Number of fields: 0.
- o X’0D’ Number of fields: 0.

6. More information

This protocol was for educational purposes. We were given the task in the class named: “Protocolos de Comunicación”