Network Working Group
Request for Comments: 931
Supersedes: RFC 912

Mike StJohns
TPSC

January 1985

Authentication Server

STATUS OF THIS MEMO

This RFC suggests a proposed protocol for the ARPA-Internet community, and requests discussion and suggestions for improvements. This is the second draft of this proposal (superseding RFC 912) and incorporates a more formal description of the syntax for the request and response dialog, as well as a change to specify the type of user identification returned. Distribution of this memo is unlimited.

INTRODUCTION

The Authentication Server Protocol provides a means to determine the identity of a user of a particular TCP connection. Given a TCP port number pair, it returns a character string which identifies the owner of that connection on the server's system. Suggested uses include automatic identification and verification of a user during an FTP session, additional verification of a TAC dial up user, and access verification for a generalized network file server.

OVERVIEW

This is a connection based application on TCP. A server listens for TCP connections on TCP port 113 (decimal). Once a connection is established, the server reads one line of data which specifies the connection of interest. If it exists, the system dependent user identifier of the connection of interest is sent out the connection. The service closes the connection after sending the user identifier.

RESTRICTIONS

Queries are permitted only for fully specified connections. The local/foreign host pair used to fully specify the connection are taken from the query connection. This means a user on Host A may only query the server on Host B about connections between A and B.

QUERY/RESPONSE FORMAT

The server accepts simple text query requests of the form

<local-port>, <foreign-port>

where <local-port> is the TCP port (decimal) on the target (server) system, and <foreign-port> is the TCP port (decimal) on the source (user) system.

For example:

23, 6191

The response is of the form

<local-port>, <foreign-port> : <response-type> : <additional-info>

where <local-port>,<foreign-port> are the same pair as the query, <response-type> is a keyword identifying the type of response, and <additional info> is context dependent.

For example:

```
23, 6191 : USERID : MULTICS : StJohns.DODCSC.a
23, 6193 : USERID : TAC : MCSJ-MITMUL
23, 6195 : ERROR : NO-USER
```

RESPONSE TYPES

A response can be one of two types:

USERID

In this case, <additional-info> is a string consisting of an operating system name, followed by a ":", followed by user identification string in a format peculiar to the operating system indicated. Permitted operating system names are specified in RFC-923, "Assigned Numbers" or its successors. The only other names permitted are "TAC" to specify a BBN Terminal Access Controller, and "OTHER" to specify any other operating system not yet registered with the NIC.

ERROR

For some reason the owner of <TCP-port> could not be determined, <additional-info> tells why. The following are suggested values of <additional-info> and their meanings.

INVALID-PORT

Either the local or foreign port was improperly specified.

NO-USER

The connection specified by the port pair is not currently in use.

UNKNOWN-ERROR

Can't determine connection owner; reason unknown. Other values may be specified as necessary.

CAVEATS

Unfortunately, the trustworthiness of the various host systems that might implement an authentication server will vary quite a bit. It is up to the various applications that will use the server to determine the amount of trust they will place in the returned information. It may be appropriate in some cases restrict the use of the server to within a locally controlled subnet.

APPLICATIONS

1) Automatic user authentication for FTP

A user-FTP may send a USER command with no argument to the server-FTP to request automatic authentication. The server-FTP will reply with a 230 (user logged in) if it can use the authentication. It will reply with a 530 (not logged in) if it cannot authenticate the user. It will reply with a 500 or 501 (syntax or parameter problem) if it does not implement automatic authentication. Please note that no change is needed to currently implemented servers to handle the request for authentication; they will reject it normally as a parameter problem. This is a suggested implementation for experimental use only.

2) Verification for privileged network operations. For example, having the server start or stop special purpose servers.

3) Elimination of "double login" for TAC and other TELNET users.

This will be implemented as a TELNET option.

FORMAL SYNTAX

```
<request>     ::= <port-pair> <CR> <LF>

<port-pair>   ::= <integer-number> "," <integer-number>

<reply>       ::= <reply-text> <CR> <LF>

<reply-text>  ::= <error-reply> | <auth-reply>

<error-reply> ::= <port-pair> ":" ERROR ":" <error-type>

<auth-reply>  ::= <port-pair> ":" USERID ":" <opsys> ":" <user-id>

<error-type>  ::= INVALID-PORT | NO-USER | UNKNOWN-ERROR

<opsys>       ::= TAC | OTHER | MULTICS | UNIX ...etc.
                  (See "Assigned Numbers")
```

Notes on Syntax:

1) White space (blanks and tab characters) between tokens is not important and may be ignored.

2) White space, the token separator character (":"), and the port pair separator character (",") must be quoted if used within a token. The quote character is a back-slash, ASCII 92 (decimal) ("\"). For example, a quoted colon is "\:". The back-slash must also be quoted if its needed to represent itself ("\\").

Notes on User Identification Format:

The user identifier returned by the server should be the standard one for the system. For example, the standard Multics identifier consists of a PERSONID followed by a ".", followed by a PROJECTID, followed by a ".", followed by an INSTANCE TAG of one character. An instance tag of "a" identifies an interactive user, and instance tag of "m" identifies an absentee job (batch job) user, and an instance tag of "z" identifies a daemon (background) user.

Each set of operating system users must come to a consensus as to

what the OFFICIAL user identification for their systems will be. Until they register this information, they must use the "OTHER" tag to specify their user identification.

Notes on User Identification Translation:

Once you have a user identifier from a remote system, you must then have a way of translating it into an identifier that meaningful on the local system. The following is a sketchy outline of table driven scheme for doing this.

The table consists of four columns, the first three are used to match against, the fourth is the result.

USERID	Opsys	Address	Result
MCSJ-MITMUL	TAC	26.*.*.*	StJohns
*	MULTICS	192.5.42.*	=
*	OTHER	10.0.0.42	anonymous
MSJ	ITS	10.3.0.44	StJohns

The above table is a sample one for a Multics system on MILNET at the Pentagon. When an authentication is returned, the particular application using the userid simply looks for the first match in the table. Notice the second line. It says that any authentication coming from a Multics system on Net 192.5.42 is accepted in the same format.

Obviously, various users will have to be registered to use this facility, but the registration can be done at the same time the use receives his login identity from the system.