Network Working Group
Request for Comments: 2619
Category: Standards Track

G. Zorn
B. Aboba
Microsoft

June 1999

RADIUS Authentication Server MIB

Status of this Memo

Copyright Notice

Abstract

This memo defines a set of extensions which instrument RADIUS authentication server functions. These extensions represent a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. Using these extensions IP-based management stations can manage RADIUS authentication servers.

1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects used for managing RADIUS authentication servers.

RADIUS authentication servers are today widely deployed by dialup Internet Service Providers, in order to provide authentication services. As a result, the effective management of RADIUS authentication servers is of considerable importance.

2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [1].
- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIv2, is described in STD 58, RFC 2578 [5], RFC 2579 [6] and RFC 2580 [7].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

3. Overview

The RADIUS authentication protocol, described in [16], distinguishes between the client function and the server function. In RADIUS authentication, clients send Access-Requests, and servers reply with Access-Accepts, Access-Rejects, and Access-Challenges. Typically NAS devices implement the client function, and thus would be expected to implement the RADIUS authentication client MIB, while RADIUS authentication servers implement the server function, and thus would be expected to implement the RADIUS authentication server MIB.

However, it is possible for a RADIUS authentication entity to perform both client and server functions. For example, a RADIUS proxy may act as a server to one or more RADIUS authentication clients, while simultaneously acting as an authentication client to one or more authentication servers. In such situations, it is expected that RADIUS entities combining client and server functionality will support both the client and server MIBs.

3.1. Selected objects

This MIB module contains fourteen scalars as well as a single table:

(1) the RADIUS Authentication Client Table contains one row for each RADIUS authentication client that the server shares a secret with.

Each entry in the RADIUS Authentication Client Table includes twelve columns presenting a view of the activity of the RADIUS authentication server.

4. Definitions

```
RADIUS-AUTH-SERVER-MIB DEFINITIONS ::= BEGIN

IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY,
       Counter32, Integer32,
       IpAddress, TimeTicks, mib-2      FROM SNMPv2-SMI
       SnmpAdminString                  FROM SNMP-FRAMEWORK-MIB
       MODULE-COMPLIANCE, OBJECT-GROUP  FROM SNMPv2-CONF;

radiusAuthServMIB MODULE-IDENTITY
       LAST-UPDATED "9906110000Z"
       ORGANIZATION "IETF RADIUS Working Group."
       CONTACT-INFO
              " Bernard Aboba
                Microsoft
```

```
                One Microsoft Way
                Redmond, WA  98052
                US

                Phone: +1 425 936 6605
                EMail: bernarda@microsoft.com"
        DESCRIPTION
              "The MIB module for entities implementing the server
               side of the Remote Access Dialin User Service (RADIUS)
               authentication protocol."
        REVISION "9906110000Z"    --  11 Jun 1999
        DESCRIPTION "Initial version as published in RFC 2619"
        ::= { radiusAuthentication 1 }

radiusMIB OBJECT-IDENTITY
        STATUS  current
        DESCRIPTION
              "The OID assigned to RADIUS MIB work by the IANA."
         ::= { mib-2 67 }

radiusAuthentication  OBJECT IDENTIFIER ::= {radiusMIB 1}

radiusAuthServMIBObjects     OBJECT IDENTIFIER ::=
                                                { radiusAuthServMIB 1 }

radiusAuthServ      OBJECT IDENTIFIER ::= { radiusAuthServMIBObjects 1 }

radiusAuthServIdent OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
              "The implementation identification string for the
               RADIUS authentication server software in use on the
               system, for example; 'FNS-2.1'"
        ::= {radiusAuthServ 1}

radiusAuthServUpTime OBJECT-TYPE
        SYNTAX      TimeTicks
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
              "If the server has a persistent state (e.g., a process),
               this value will be the time elapsed (in hundredths of a
               seco) since the server process was started.
               For software without persistent state, this value will
               be zero."
        ::= {radiusAuthServ 2}
```

```
radiusAuthServResetTime OBJECT-TYPE
       SYNTAX      TimeTicks
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
             "If the server has a persistent state (e.g., a process)
              and supports a 'reset' operation (e.g., can be told to
              re-read configuration files), this value will be the
              time elapsed (in hundredths of a second) since the
              server was 'reset.'  For software that does not
              have persistence or does not support a 'reset' operation,
              this value will be zero."
       ::= {radiusAuthServ 3}

radiusAuthServConfigReset OBJECT-TYPE
       SYNTAX INTEGER { other(1),
                        reset(2),
                        initializing(3),
                        running(4)}
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
              "Status/action object to reinitialize any persistent
               server state.  When set to reset(2), any persistent
               server state (such as a process) is reinitialized as if
               the server had just been started.  This value will
               never be returned by a read operation.  When read, one of
               the following values will be returned:
                   other(1) - server in some unknown state;
                   initializing(3) - server (re)initializing;
                   running(4) - server currently running."
       ::= {radiusAuthServ 4}

-- New Stats proposed by Dale E. Reed Jr (daler@iea-software.com)

radiusAuthServTotalAccessRequests OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
               "The number of packets received on the
                authentication port."
        ::= { radiusAuthServ 5}

radiusAuthServTotalInvalidRequests OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
```

```
        DESCRIPTION
              "The number of RADIUS Access-Request packets
               received from unknown addresses."
        ::= { radiusAuthServ 6 }

radiusAuthServTotalDupAccessRequests OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The number of duplicate RADIUS Access-Request
              packets received."
       ::= { radiusAuthServ 7 }

radiusAuthServTotalAccessAccepts OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The number of RADIUS Access-Accept packets sent."
       ::= { radiusAuthServ 8 }

radiusAuthServTotalAccessRejects OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of RADIUS Access-Reject packets sent."
      ::= { radiusAuthServ 9 }

radiusAuthServTotalAccessChallenges OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of RADIUS Access-Challenge packets sent."
      ::= { radiusAuthServ 10 }

radiusAuthServTotalMalformedAccessRequests OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of malformed RADIUS Access-Request
             packets received. Bad authenticators
             and unknown types are not included as
             malformed Access-Requests."
      ::= { radiusAuthServ 11 }
```

```
radiusAuthServTotalBadAuthenticators OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of RADIUS Authentication-Request packets
             which contained invalid Signature attributes received."
      ::= { radiusAuthServ 12 }

radiusAuthServTotalPacketsDropped OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of incoming packets
             silently discarded for some reason other
             than malformed, bad authenticators or
             unknown types."
      ::= { radiusAuthServ 13 }

radiusAuthServTotalUnknownTypes OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of RADIUS packets of unknown type which
             were received."
      ::= { radiusAuthServ 14 }

-- End of new

radiusAuthClientTable OBJECT-TYPE
       SYNTAX     SEQUENCE OF RadiusAuthClientEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
             "The (conceptual) table listing the RADIUS authentication
              clients with which the server shares a secret."
       ::= { radiusAuthServ 15 }

radiusAuthClientEntry OBJECT-TYPE
       SYNTAX     RadiusAuthClientEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
             "An entry (conceptual row) representing a RADIUS
              authentication client with which the server shares a
              secret."
```

```
       INDEX      { radiusAuthClientIndex }
       ::= { radiusAuthClientTable 1 }

RadiusAuthClientEntry ::= SEQUENCE {
       radiusAuthClientIndex                          Integer32,
       radiusAuthClientAddress                        IpAddress,
       radiusAuthClientID                          SnmpAdminString,
       radiusAuthServAccessRequests                   Counter32,
       radiusAuthServDupAccessRequests                Counter32,
       radiusAuthServAccessAccepts                    Counter32,
       radiusAuthServAccessRejects                    Counter32,
       radiusAuthServAccessChallenges                 Counter32,
       radiusAuthServMalformedAccessRequests          Counter32,
       radiusAuthServBadAuthenticators                Counter32,
       radiusAuthServPacketsDropped                   Counter32,
       radiusAuthServUnknownTypes                     Counter32
}

radiusAuthClientIndex OBJECT-TYPE
       SYNTAX     Integer32 (1..2147483647)
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
             "A number uniquely identifying each RADIUS
              authentication client with which this server
              communicates."
       ::= { radiusAuthClientEntry 1 }

radiusAuthClientAddress OBJECT-TYPE
       SYNTAX     IpAddress
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
             "The NAS-IP-Address of the RADIUS authentication client
              referred to in this table entry."
       ::= { radiusAuthClientEntry 2 }

radiusAuthClientID OBJECT-TYPE
       SYNTAX     SnmpAdminString
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
             "The NAS-Identifier of the RADIUS authentication client
              referred to in this table entry. This is not necessarily
              the same as sysName in MIB II."
       ::= { radiusAuthClientEntry 3 }

-- Server Counters
```

```
--
-- Responses = AccessAccepts + AccessRejects + AccessChallenges
--
-- Requests - DupRequests - BadAuthenticators - MalformedRequests -
-- UnknownTypes -  PacketsDropped - Responses = Pending
--
-- Requests - DupRequests - BadAuthenticators - MalformedRequests -
-- UnknownTypes - PacketsDropped = entries logged

radiusAuthServAccessRequests OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The number of packets received on the authentication
              port from this client."
       ::= { radiusAuthClientEntry  4 }

radiusAuthServDupAccessRequests OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The number of duplicate RADIUS Access-Request
              packets received from this client."
       ::= { radiusAuthClientEntry  5 }

radiusAuthServAccessAccepts OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The number of RADIUS Access-Accept packets
              sent to this client."
       ::= { radiusAuthClientEntry  6 }

radiusAuthServAccessRejects OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
             "The number of RADIUS Access-Reject packets
              sent to this client."
       ::= { radiusAuthClientEntry  7 }

radiusAuthServAccessChallenges OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
```

```
        STATUS current
        DESCRIPTION
              "The number of RADIUS Access-Challenge packets
               sent to this client."
        ::= { radiusAuthClientEntry  8 }

radiusAuthServMalformedAccessRequests OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "The number of malformed RADIUS Access-Request
               packets received from this client.
               Bad authenticators and unknown types are not included as
               malformed Access-Requests."
        ::= { radiusAuthClientEntry  9 }

radiusAuthServBadAuthenticators OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "The number of RADIUS Authentication-Request packets
               which contained invalid Signature attributes received
               from this client."
        ::= { radiusAuthClientEntry  10 }

radiusAuthServPacketsDropped OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
              "The number of incoming packets from this
               client silently discarded for some reason other
               than malformed, bad authenticators or
               unknown types."
       ::= { radiusAuthClientEntry  11 }

radiusAuthServUnknownTypes OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "The number of RADIUS packets of unknown type which
               were received from this client."
        ::= { radiusAuthClientEntry  12 }
```

```
-- conformance information

radiusAuthServMIBConformance
              OBJECT IDENTIFIER ::= { radiusAuthServMIB 2 }
radiusAuthServMIBCompliances
              OBJECT IDENTIFIER ::= { radiusAuthServMIBConformance 1 }
radiusAuthServMIBGroups
              OBJECT IDENTIFIER ::= { radiusAuthServMIBConformance 2 }


-- compliance statements

radiusAuthServMIBCompliance MODULE-COMPLIANCE
            STATUS  current
            DESCRIPTION
                  "The compliance statement for authentication servers
                   implementing the RADIUS Authentication Server MIB."
            MODULE  -- this module
            MANDATORY-GROUPS { radiusAuthServMIBGroup }

            OBJECT        radiusAuthServConfigReset
            WRITE-SYNTAX  INTEGER { reset(2) }
            DESCRIPTION  "The only SETable value is 'reset' (2)."

            ::= { radiusAuthServMIBCompliances 1 }

-- units of conformance

radiusAuthServMIBGroup OBJECT-GROUP
      OBJECTS {radiusAuthServIdent,
               radiusAuthServUpTime,
               radiusAuthServResetTime,
               radiusAuthServConfigReset,
               radiusAuthServTotalAccessRequests,
               radiusAuthServTotalInvalidRequests,
               radiusAuthServTotalDupAccessRequests,
               radiusAuthServTotalAccessAccepts,
               radiusAuthServTotalAccessRejects,
               radiusAuthServTotalAccessChallenges,
               radiusAuthServTotalMalformedAccessRequests,
               radiusAuthServTotalBadAuthenticators,
               radiusAuthServTotalPacketsDropped,
               radiusAuthServTotalUnknownTypes,
               radiusAuthClientAddress,
               radiusAuthClientID,
               radiusAuthServAccessRequests,
               radiusAuthServDupAccessRequests,
               radiusAuthServAccessAccepts,
```

```
                radiusAuthServAccessRejects,
                radiusAuthServAccessChallenges,
                radiusAuthServMalformedAccessRequests,
                radiusAuthServBadAuthenticators,
                radiusAuthServPacketsDropped,
                radiusAuthServUnknownTypes
               }
      STATUS  current
      DESCRIPTION
            "The collection of objects providing management of
             a RADIUS Authentication Server."
      ::= { radiusAuthServMIBGroups 1 }

END
```

5. References

[1] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.

[2] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

[3] Rose, M., and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[5] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[7] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[8] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[9] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[10] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

[11] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.

[12] Blumenthal, U., and B. Wijnen, "User-based Security Model for Version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.

[13] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

[14] Levi, D., Meyer, P., and B. Stewart, "SNMP Applications", RFC 2573, April 1999.

[15] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.

[16] Rigney, C., Rubens, A., Simpson W. and S. Willens, "Remote Authentication Dial In User Service (RADIUS)", RFC 2138, April 1997.

6. Security Considerations

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

There are a number of managed objects in this MIB that may contain sensitive information. These are:

radiusAuthClientAddress
: This can be used to determine the address of the RADIUS authentication client with which the server is communicating. This information could be useful in impersonating the client.

radiusAuthClientID
: This can be used to determine the client ID of the authentication client with which the server is communicating. This information could be useful in

impersonating the client.

It is thus important to control even GET access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [12] and the View-based Access Control Model RFC 2575 [15] is recommended. Using these security features, customer/users can give access to the objects only to those principals (users) that have legitimate rights to GET or SET (change/create/delete) them.

7. Acknowledgments

The authors acknowledge the contributions of the RADIUS Working Group in the development of this MIB. Thanks to Narendra Gidwani of Microsoft, Allan C. Rubens of MERIT, Carl Rigney of Livingston and Peter Heitman of American Internet Corporation for useful discussions of this problem space.

8. Authors' Addresses

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: 425-936-6605
EMail: bernarda@microsoft.com

Glen Zorn
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: 425-703-1559
EMail: glennz@microsoft.com

9. Intellectural Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards- related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

10. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.