                    RADIUS Authentication Client MIB

Status of this Memo

Copyright Notice

Abstract

   This memo defines a set of extensions which instrument RADIUS
   authentication client 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 clients.

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 clients.

   Today a wide range of network devices, including routers and NASes,
   act as RADIUS authentication clients in order to provide
   authentication and authorization services. As a result, the effective
   management of RADIUS authentication clients is of considerable
   importance.

2. The SNMP Management Framework

   The SNMP Management Framework presently consists of five major
   components:

   o   An overall architecture, described in RFC 2571 [1].

   o   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].

   o   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].

   o   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].

   o   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 two scalars as well as a single table:

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

   Each entry in the RADIUS Authentication Server Table includes fifteen
   columns presenting a view of the activity of the RADIUS
   authentication client.

4. Definitions

RADIUS-AUTH-CLIENT-MIB DEFINITIONS ::= BEGIN

IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY,
        Counter32, Integer32, Gauge32,
        IpAddress, TimeTicks, mib-2   FROM SNMPv2-SMI
        SnmpAdminString   FROM SNMP-FRAMEWORK-MIB
        MODULE-COMPLIANCE, OBJECT-GROUP FROM SNMPv2-CONF;

radiusAuthClientMIB MODULE-IDENTITY
        LAST-UPDATED "9906110000Z" -- 11 Jun 1999
        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 client
                   side of the Remote Access Dialin User Service (RADIUS)
                   authentication protocol."
            REVISION "9906110000Z"   -- 11 Jun 1999
            DESCRIPTION "Initial version as published in RFC 2618"
            ::= { radiusAuthentication 2 }

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

radiusAuthentication OBJECT IDENTIFIER ::= {radiusMIB 1}

radiusAuthClientMIBObjects   OBJECT IDENTIFIER ::=
                                        { radiusAuthClientMIB 1 }

radiusAuthClient OBJECT IDENTIFIER ::= { radiusAuthClientMIBObjects 1 }

radiusAuthClientInvalidServerAddresses OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "The number of RADIUS Access-Response packets
               received from unknown addresses."
        ::= { radiusAuthClient 1 }

radiusAuthClientIdentifier OBJECT-TYPE
        SYNTAX SnmpAdminString
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "The NAS-Identifier of the RADIUS authentication client.
               This is not necessarily the same as sysName in MIB II."
        ::= { radiusAuthClient 2 }

radiusAuthServerTable OBJECT-TYPE
        SYNTAX   SEQUENCE OF RadiusAuthServerEntry
        MAX-ACCESS not-accessible

```
        STATUS      current
        DESCRIPTION
              "The (conceptual) table listing the RADIUS authentication
               servers with which the client shares a secret."
        ::= { radiusAuthClient 3 }

radiusAuthServerEntry OBJECT-TYPE
        SYNTAX      RadiusAuthServerEntry
        MAX-ACCESS not-accessible
        STATUS      current
        DESCRIPTION
              "An entry (conceptual row) representing a RADIUS
               authentication server with which the client shares
               a secret."
        INDEX       { radiusAuthServerIndex }
        ::= { radiusAuthServerTable 1 }

RadiusAuthServerEntry ::= SEQUENCE {
        radiusAuthServerIndex                       Integer32,
        radiusAuthServerAddress                     IpAddress,
        radiusAuthClientServerPortNumber            Integer32,
        radiusAuthClientRoundTripTime               TimeTicks,
        radiusAuthClientAccessRequests              Counter32,
        radiusAuthClientAccessRetransmissions       Counter32,
        radiusAuthClientAccessAccepts               Counter32,
        radiusAuthClientAccessRejects               Counter32,
        radiusAuthClientAccessChallenges            Counter32,
        radiusAuthClientMalformedAccessResponses    Counter32,
        radiusAuthClientBadAuthenticators           Counter32,
        radiusAuthClientPendingRequests               Gauge32,
        radiusAuthClientTimeouts                    Counter32,
        radiusAuthClientUnknownTypes                Counter32,
        radiusAuthClientPacketsDropped              Counter32
}

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

radiusAuthServerAddress OBJECT-TYPE
        SYNTAX      IpAddress
        MAX-ACCESS read-only
```

              STATUS   current
              DESCRIPTION
                    "The IP address of the RADIUS authentication server
                     referred to in this table entry."
              ::= { radiusAuthServerEntry 2 }

radiusAuthClientServerPortNumber OBJECT-TYPE
              SYNTAX Integer32 (0..65535)
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                    "The UDP port the client is using to send requests to
                     this server."
              ::= { radiusAuthServerEntry 3 }

radiusAuthClientRoundTripTime OBJECT-TYPE
              SYNTAX TimeTicks
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                    "The time interval (in hundredths of a second) between
                     the most recent Access-Reply/Access-Challenge and the
                     Access-Request that matched it from this RADIUS
                     authentication server."
              ::= { radiusAuthServerEntry 4 }

-- Request/Response statistics
--
-- TotalIncomingPackets = Accepts + Rejects + Challenges + UnknownTypes
--
-- TotalIncomingPackets - MalformedResponses - BadAuthenticators -
-- UnknownTypes - PacketsDropped = Successfully received
--
-- AccessRequests + PendingRequests + ClientTimeouts =
-- Successfully Received
--
--

radiusAuthClientAccessRequests OBJECT-TYPE
              SYNTAX Counter32
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                    "The number of RADIUS Access-Request packets sent
                     to this server. This does not include retransmissions."
              ::= { radiusAuthServerEntry 5 }

radiusAuthClientAccessRetransmissions OBJECT-TYPE

         SYNTAX Counter32
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
               "The number of RADIUS Access-Request packets
                retransmitted to this RADIUS authentication server."
         ::= { radiusAuthServerEntry 6 }

radiusAuthClientAccessAccepts OBJECT-TYPE
         SYNTAX Counter32
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
               "The number of RADIUS Access-Accept packets
                (valid or invalid) received from this server."
         ::= { radiusAuthServerEntry 7 }

radiusAuthClientAccessRejects OBJECT-TYPE
         SYNTAX Counter32
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
               "The number of RADIUS Access-Reject packets
                (valid or invalid) received from this server."
         ::= { radiusAuthServerEntry 8 }

radiusAuthClientAccessChallenges OBJECT-TYPE
         SYNTAX Counter32
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
               "The number of RADIUS Access-Challenge packets
                (valid or invalid) received from this server."
         ::= { radiusAuthServerEntry 9 }

-- "Access-Response" includes an Access-Accept, Access-Challenge
-- or Access-Reject

radiusAuthClientMalformedAccessResponses OBJECT-TYPE
         SYNTAX Counter32
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
               "The number of malformed RADIUS Access-Response
                packets received from this server.
                Malformed packets include packets with
                an invalid length. Bad authenticators or
                Signature attributes or unknown types are not

                included as malformed access responses."
          ::= { radiusAuthServerEntry 10 }

radiusAuthClientBadAuthenticators OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "The number of RADIUS Access-Response packets
               containing invalid authenticators or Signature
               attributes received from this server."
          ::= { radiusAuthServerEntry 11 }

radiusAuthClientPendingRequests OBJECT-TYPE
        SYNTAX Gauge32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "The number of RADIUS Access-Request packets
               destined for this server that have not yet timed out
               or received a response. This variable is incremented
               when an Access-Request is sent and decremented due to
               receipt of an Acess-Accept, Access-Reject or
               Access-Challenge, a timeout or retransmission."
           ::= { radiusAuthServerEntry 12 }

radiusAuthClientTimeouts OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The number of authentication timeouts to this server.
               After a timeout the client may retry to the same
               server, send to a different server, or
               give up. A retry to the same server is counted as a
               retransmit as well as a timeout. A send to a different
               server is counted as a Request as well as a timeout."
          ::= { radiusAuthServerEntry 13 }

radiusAuthClientUnknownTypes OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "The number of RADIUS packets of unknown type which
               were received from this server on the authentication port."
          ::= { radiusAuthServerEntry 14 }

radiusAuthClientPacketsDropped OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of RADIUS packets of which were
             received from this server on the authentication port
             and dropped for some other reason."
      ::= { radiusAuthServerEntry 15 }

-- conformance information

radiusAuthClientMIBConformance
            OBJECT IDENTIFIER ::= { radiusAuthClientMIB 2 }
radiusAuthClientMIBCompliances
            OBJECT IDENTIFIER ::= { radiusAuthClientMIBConformance 1 }
radiusAuthClientMIBGroups
            OBJECT IDENTIFIER ::= { radiusAuthClientMIBConformance 2 }

-- compliance statements

radiusAuthClientMIBCompliance MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION
            "The compliance statement for authentication clients
             implementing the RADIUS Authentication Client MIB."
      MODULE -- this module
            MANDATORY-GROUPS { radiusAuthClientMIBGroup }

      ::= { radiusAuthClientMIBCompliances 1 }

-- units of conformance

radiusAuthClientMIBGroup OBJECT-GROUP
      OBJECTS { radiusAuthClientIdentifier,
                radiusAuthClientInvalidServerAddresses,
                radiusAuthServerAddress,
                radiusAuthClientServerPortNumber,
                radiusAuthClientRoundTripTime,
                radiusAuthClientAccessRequests,
                radiusAuthClientAccessRetransmissions,
                radiusAuthClientAccessAccepts,
                radiusAuthClientAccessRejects,
                radiusAuthClientAccessChallenges,
                radiusAuthClientMalformedAccessResponses,

```
                radiusAuthClientBadAuthenticators,
                radiusAuthClientPendingRequests,
                radiusAuthClientTimeouts,
                radiusAuthClientUnknownTypes,
                radiusAuthClientPacketsDropped
              }
        STATUS  current
        DESCRIPTION
              "The basic collection of objects providing management of
               RADIUS Authentication Clients."
        ::= { radiusAuthClientMIBGroups 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, Performance Systems International, 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 no management objects defined in this MIB that have a MAX-
   ACCESS clause of read-write and/or read-create. So, if this MIB is
   implemented correctly, then there is no risk that an intruder can
   alter or create any management objects of this MIB via direct SNMP
   SET operations.

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

   radiusAuthServerAddress
           This can be used to determine the address of the RADIUS
           authentication server with which the client is
           communicating. This information could be useful in
           mounting an attack on the authentication server.

   radiusAuthClientServerPortNumber This can be used to determine the
           port number on which the RADIUS authentication client is
           sending. This information could be useful in impersonating
           the client in order to send data to the authentication

          server.

     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 Wy
     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. Intellectual 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