Network Working Group
Internet-Draft
Expires: December 8, 2004

S. De Cnodder
Alcatel
N. Jonnala
Future Soft
M. Chiba
Cisco Systems, Inc.
June 9, 2004

RADIUS Dynamic Authorization Server MIB
draft-decnodder-radext-dynauth-server-mib-01.txt

Status of this Memo

Copyright Notice

Abstract

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 the RADIUS dynamic authorization server (DAS) functions that support the dynamic authorization extensions as defined in RFC 3576.

Table of Contents

1. Requirements notation

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 [RFC2119].

2. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. It is becoming increasingly important to support Dynamic Authorization extensions on the NAS devices to handle the Disconnect and CoA messages as described in [RFC3576] . As a result, the effective management of RADIUS Dynamic Authorization entities is of considerable importance. It complements the managed objects used for managing RADIUS authentication and accounting clients as described in [RFC2618] and [RFC2620], respectively.

3. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC2578 [RFC2578], STD 58, RFC2579 [RFC2579] and STD 58, RFC2580 [RFC2580].

4. Terminology

Dynamic Authorization Server (DAS)

The component that resides on the NAS which processes the Disconnect and CoA requests sent by the Dynamic Authorization Client as described in [RFC3576].

Dynamic Authorization Client (DAC)

The component which sends the Disconnect and CoA requests to the Dynamic Authorization Server as described in [RFC3576].

Dynamic Authorization Server Port

The UDP port on which the Dynamic Authorization server listens for the Disconnect and CoA requests sent by the Dynamic Authorization Client.

5. Overview

The RADIUS dynamic authorization extensions defined in [RFC3576], distinguishes between the client function and the server function. In RADIUS dynamic authorization, clients send Disconnect-Requests and CoA-Requests, and servers reply with Disconnect-Acks, CoA-Acks, and CoA-NAKs. Typically NAS devices implement the DAS function, and thus would be expected to implement the RADIUS dynamic authorization server MIB, while DACs implement the client function, and thus would be expected to implement the RADIUS dynamic authorization client MIB.

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

This memo describes the MIB for dynamic authorization servers and relates to the following document as follows:

[RFC2618] describes the MIB for a RADIUS authentication client.

[RFC2619] describes the MIB for a RADIUS authentication server.

[RFC2620] describes the MIB for a RADIUS accounting client.

[RFC2621] describes the MIB for a RADIUS accounting server.

[DYNCLNT] describes the MIB for a RADIUS dynamic authorization client.

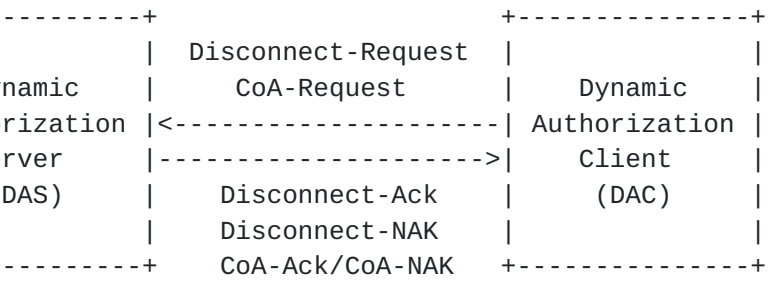

Figure 1: Mapping of clients and servers.

This MIB module for the dynamic authorization server contains the following:

1. Two scalar objects

2. One Dynamic Authorization Client Table. This table contains one row for each DAC that the DAS shares a secret with.

6. RADIUS Dynamic Authorization Server MIB Definitions

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

IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY,
       Counter32, Integer32, mib-2      FROM SNMPv2-SMI
       SnmpAdminString                  FROM SNMP-FRAMEWORK-MIB
       InetAddressType, InetAddress FROM INET-ADDRESS-MIB
       MODULE-COMPLIANCE, OBJECT-GROUP  FROM SNMPv2-CONF;

radiusDynAuthServerMIB MODULE-IDENTITY
       LAST-UPDATED "200404260000Z" -- 26 April 2004
       ORGANIZATION "IETF RADEXT Working Group"
       CONTACT-INFO
              " Stefaan De Cnodder
                Alcatel
                Francis Wellesplein 1
                B-2018 Antwerp
                Belgium

                Phone: +32 3 240 85 15
                EMail: stefaan.de_cnodder@alcatel.be

                Nagi Reddy Jonnala
                Future Soft
                480 - 481, Anna Salai
                Nandanam, Chennai
                India

                EMail: nagi_reddy.jonnala@alcatel.be

                Murtaza Chiba
                Cisco Systems, Inc.
                170 West Tasman Dr.
                San Jose CA, 95134

                Phone: +1 408 525 7198
                EMail: mchiba@cisco.com "
       DESCRIPTION
           "The MIB module for entities implementing the server
            side of the Dynamic Authorization extensions Remote
            Access Dialin User Service (RADIUS) protocol.

            Copyright (C) The Internet Society (date).  This version
            of this MIB module is part of RFC yyyy;  see the RFC
```

```
            itself for full legal notices."
   -- RFC Ed.: replace yyyy with actual RFC number & remove this note

          REVISION "200404260000Z" -- 26 April 2004
          DESCRIPTION "Initial version as published in RFC XXXX"
          ::= { radiusDynamicAuthorization 1 }

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

   -- Ed. Note: The next available OID 3 is picked for
   -- radiusDynamicAuthorization.

   radiusDynamicAuthorization    OBJECT IDENTIFIER ::= { radiusMIB 3 }

   radiusDynAuthServerMIBObjects OBJECT IDENTIFIER ::=
                                        { radiusDynAuthServerMIB 1 }

   radiusDynAuthServer           OBJECT IDENTIFIER ::=
                                 { radiusDynAuthServerMIBObjects 1 }

   radiusDynAuthServerInvalidClientAddresses OBJECT-TYPE
         SYNTAX Counter32
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
               "The number of RADIUS dynamic authorization messages
                (both Disconnect and CoA) received from unknown
                addresses."
         ::= { radiusDynAuthServer 1 }

   radiusDynAuthServerIdentifier OBJECT-TYPE
         SYNTAX SnmpAdminString
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
                "The NAS-Identifier of the RADIUS dynamic authorization
                 server."
         ::= { radiusDynAuthServer 2 }

   radiusDynAuthClientTable OBJECT-TYPE
         SYNTAX SEQUENCE OF RadiusDynAuthClientEntry
         MAX-ACCESS not-accessible
         STATUS     current
         DESCRIPTION
```

```
              "The (conceptual) table listing the RADIUS dynamic
               authorization clients with which the server shares a
               secret."
         ::= { radiusDynAuthServer 3 }

   radiusDynAuthClientEntry OBJECT-TYPE
          SYNTAX     RadiusDynAuthClientEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                "An entry (conceptual row) representing the Dynamic
                 Authorization Client with which the server shares a
                 secret."
          INDEX      { radiusDynAuthClientIndex }
          ::= { radiusDynAuthClientTable 1 }

   RadiusDynAuthClientEntry ::= SEQUENCE {
          radiusDynAuthClientIndex                    Integer32,
          radiusDynAuthClientAddressType              InetAddressType,
          radiusDynAuthClientAddress                  InetAddress,
          radiusDynAuthServDisconRequests             Counter32,
          radiusDynAuthServDupDisconRequests          Counter32,
          radiusDynAuthServDisconAcks                 Counter32,
          radiusDynAuthServDisconNaks                 Counter32,
          radiusDynAuthServDisconUserSessRemoved      Counter32,
          radiusDynAuthServMalformedDisconRequests    Counter32,
          radiusDynAuthServDisconBadAuthenticators    Counter32,
          radiusDynAuthServDisconPacketsDropped       Counter32,
          radiusDynAuthServCoARequests                Counter32,
          radiusDynAuthServDupCoARequests             Counter32,
          radiusDynAuthServCoAAcks                    Counter32,
          radiusDynAuthServCoANaks                    Counter32,
          radiusDynAuthServCoAUserSessChanged         Counter32,
          radiusDynAuthServMalformedCoARequests       Counter32,
          radiusDynAuthServCoABadAuthenticators       Counter32,
          radiusDynAuthServCoAPacketsDropped          Counter32,
          radiusDynAuthServUnknownTypes               Counter32

   -- Ed. Note: Do we need counter for silently discarded replay
   -- packets. Please note that if the Event-Time-stamp is outside
   -- the time window then the request can be silently discarded.

   -- Ed. Note :  Do we need counters for error causes defined in
   -- RFC-3576. */

   }
```

```
   radiusDynAuthClientIndex OBJECT-TYPE
          SYNTAX     Integer32 (1..2147483647)
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                "A number uniquely identifying each RADIUS
                 dynamic authorization client  with which this Dynamic
                 Authorization Server communicates."
          ::= { radiusDynAuthClientEntry 1 }

   radiusDynAuthClientAddressType OBJECT-TYPE
          SYNTAX     InetAddressType
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                "The type of IP-Address of the RADIUS Dynamic
                 Authorization Client referred to in this table entry."
          ::= { radiusDynAuthClientEntry 2 }

   radiusDynAuthClientAddress OBJECT-TYPE
          SYNTAX     InetAddress
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                "The IP-Address value of the RADIUS Dynamic
                 Authorization Client referred to in this table entry."
          ::= { radiusDynAuthClientEntry 3 }

   radiusDynAuthServDisconRequests OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of RADIUS Disconnect-Requests received
                 from this Dynamic Authorization Client."
          ::= { radiusDynAuthClientEntry  4 }

   radiusDynAuthServDupDisconRequests OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of duplicate RADIUS Disconnect-Request
                 packets received from this Dynamic Authorization
                 Client."
          ::= { radiusDynAuthClientEntry  5 }

   radiusDynAuthServDisconAcks  OBJECT-TYPE
```

```
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of RADIUS Disconnect-ACK packets
                 sent to this Dynamic Authorization Client"
          ::= { radiusDynAuthClientEntry  6 }

   radiusDynAuthServDisconNaks OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of RADIUS Disconnect-NAK packets
                 sent to this Dynamic Authorization Client."
          ::= { radiusDynAuthClientEntry  7 }

   radiusDynAuthServDisconUserSessRemoved OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of user sessions removed for the
                 Disconnect-Requests received from this
                 Dynamic Authorization Client. Depending on site
                 specific policies, a single Disconnect request
                 can remove multiple user sessions."
          ::= { radiusDynAuthClientEntry  8 }

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

   radiusDynAuthServDisconBadAuthenticators OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of RADIUS Disconnect-Request packets
                 which contained invalid Signature attributes
                 received from this Dynamic Authorization Client."
```

```
          ::= { radiusDynAuthClientEntry  10 }

   radiusDynAuthServDisconPacketsDropped OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of incoming Disconnect-Requests
                 from this Dynamic Authorization Client silently
                 discarded for some reason other than malformed,
                 bad authenticators or unknown types."
          ::= { radiusDynAuthClientEntry  11 }

   radiusDynAuthServCoARequests OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of CoA requests received from this
                 Dynamic Authorization Client."
          ::= { radiusDynAuthClientEntry  12 }

   radiusDynAuthServDupCoARequests OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of duplicate RADIUS CoA-Request
                 packets received from this Dynamic Authorization
                 client."
          ::= { radiusDynAuthClientEntry  13 }

   radiusDynAuthServCoAAcks  OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of RADIUS CoA-ACK packets
                 sent to this Dynamic Authorization Client."
          ::= { radiusDynAuthClientEntry 14 }

   radiusDynAuthServCoANaks OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of RADIUS CoA-NAK packets
                 sent to this Dynamic Authorization Client."
```

```
        ::= { radiusDynAuthClientEntry  15 }

   radiusDynAuthServCoAUserSessChanged OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of user sessions authorization
                 changed for the CoA-Requests received from this
                 Dynamic Authorization Cient. Depending on site
                 specific policies, a single CoA request can change
                 multiple user sessions authorization"
          ::= { radiusDynAuthClientEntry  16 }

   radiusDynAuthServMalformedCoARequests OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of malformed RADIUS CoA-Request
                 packets received from this Dynamic Authorization
                 Client. Bad authenticators and unknown types are not
                 included as malformed CoA-Requests."
          ::= { radiusDynAuthClientEntry  17 }

   radiusDynAuthServCoABadAuthenticators OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of RADIUS CoA-Request packets which
                 contained invalid Signature attributes received
                 from this Dynamic Authorization client."
          ::= { radiusDynAuthClientEntry  18 }

   radiusDynAuthServCoAPacketsDropped OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of incoming CoA packets from this
                 Dynamic Authorization Client silently discarded
                 for some reason other than malformed, bad
                 authenticators or unknown types."
          ::= { radiusDynAuthClientEntry  19 }

   radiusDynAuthServUnknownTypes OBJECT-TYPE
          SYNTAX Counter32
```

```
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                "The number of incoming packets of unknown types
                 which were received on the Dynamic Authorization port."
          ::= { radiusDynAuthClientEntry  20 }

   -- conformance information

   radiusDynAuthServerMIBConformance
          OBJECT IDENTIFIER ::= { radiusDynAuthServerMIB 2 }
   radiusDynAuthServerMIBCompliances
          OBJECT IDENTIFIER ::= { radiusDynAuthServerMIBConformance 1 }
   radiusDynAuthServerMIBGroups
          OBJECT IDENTIFIER ::= { radiusDynAuthServerMIBConformance 2 }

   -- compliance statements

   radiusAuthServerMIBCompliance MODULE-COMPLIANCE
          STATUS  current
          DESCRIPTION
                "The compliance statement for entities implementing
                 the RADIUS Dynamic Authorization Server."
          MODULE  -- this module
          MANDATORY-GROUPS { radiusDynAuthServerMIBGroup }
          ::= { radiusDynAuthServerMIBCompliances 1 }

   -- units of conformance

   radiusDynAuthServerMIBGroup OBJECT-GROUP
          OBJECTS { radiusDynAuthServerInvalidClientAddresses,
                    radiusDynAuthServerIdentifier,
                    radiusDynAuthClientAddressType,
                    radiusDynAuthClientAddress,
                    radiusDynAuthServDisconRequests,
                    radiusDynAuthServDupDisconRequests,
                    radiusDynAuthServDisconAcks,
                    radiusDynAuthServDisconNaks,
                    radiusDynAuthServDisconUserSessRemoved,
                    radiusDynAuthServMalformedDisconRequests,
                    radiusDynAuthServDisconBadAuthenticators,
                    radiusDynAuthServDisconPacketsDropped,
                    radiusDynAuthServCoARequests,
                    radiusDynAuthServDupCoARequests,
                    radiusDynAuthServCoAAcks,
                    radiusDynAuthServCoANaks,
                    radiusDynAuthServCoAUserSessChanged,
                    radiusDynAuthServMalformedCoARequests,
```

```
                  radiusDynAuthServCoABadAuthenticators,
                  radiusDynAuthServCoAPacketsDropped,
                  radiusDynAuthServUnknownTypes
                }
        STATUS  current
        DESCRIPTION
              "The collection of objects providing management of
               a RADIUS Dynamic Authorization Server."
        ::= { radiusDynAuthServerMIBGroups 1 }

   END
```

7. Security Considerations

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

radiusDynAuthClientAddress and radiusDynAuthClientAddressType

This can be used to determine the address of the DAC with which the DAS is communicating. This information could be useful in mounting an attack on the DAC.

radiusDynAuthServerIdentifier

This can be used to determine the Identifier of the DAS. This information could be useful in impersonating the DAS.

The other readable objects are not really considered as being sensitive or vulnerable. These objects are:

radiusDynAuthServerInvalidClientAddresses,
radiusDynAuthServDisconRequests,
radiusDynAuthServDupDisconRequests,
radiusDynAuthServDisconAcks,
radiusDynAuthServDisconNaks,
radiusDynAuthServDisconUserSessRemoved,
radiusDynAuthServMalformedDisconRequests,
radiusDynAuthServDisconBadAuthenticators,
radiusDynAuthServDisconPacketsDropped,
radiusDynAuthServCoARequests,
radiusDynAuthServDupCoARequests,
radiusDynAuthServCoAAcks,
radiusDynAuthServCoANaks,
radiusDynAuthServCoAUserSessChanged,
radiusDynAuthServMalformedCoARequests,
radiusDynAuthServCoABadAuthenticators,
radiusDynAuthServCoAPacketsDropped, and
radiusDynAuthServUnknownTypes.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, 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 module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

8. Acknowledgements

This document reuses some of the work done in earlier RADIUS MIB specifications [RFC2618] and [RFC2620].

The authors would also like to acknowledge the following people for their contributions to this document: Anjaneyulu Pata.

9. References

9.1 Normative References

[DYNCLNT] De Cnodder, S., Jonnala, N. and M. Chiba, "RADIUS Dynamic Auhtorization Client MIB", draft-decnodder-radext-dynauth-client-mib-01.txt, work in progress, June 2004.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC2578] 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.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC2618] Aboba, B. and G. Zorn, "RADIUS Authentication Client MIB", RFC 2618, June 1999.

[RFC2619] Zorn, G. and B. Aboba, "RADIUS Authentication Server MIB", RFC 2619, June 1999.

[RFC2620] Aboba, B. and G. Zorn, "RADIUS Accounting Client MIB", RFC 2620, June 1999.

[RFC2621] Zorn, G. and B. Aboba, "RADIUS Accounting Server MIB", RFC 2621, June 1999.

[RFC3576] Chiba, M., Dommety, G., Eklund, M., Mitton, D. and B. Aboba, "Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)", RFC 3576, July 2003.

9.2 Informative References

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet Standard Management Framework", RFC 3410, December 2002.

Authors' Addresses

Stefaan De Cnodder
Alcatel
Francis Wellesplein 1
B-2018 Antwerp
Belgium

Phone: +32 3 240 85 15
EMail: stefaan.de_cnodder@alcatel.be

Nagi Reddy Jonnala
Future Soft
480 - 481, Anna Salai
Nandanam, Chennai
India

EMail: nagi_reddy.jonnala@alcatel.be

Murtaza Chiba
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose CA, 95134

Phone: +1 408 525 7198
EMail: mchiba@cisco.com

Intellectual Property Statement

Full Copyright Statement

Acknowledgment

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