The Basic Configuration Model for SNMPv2

19 March 1995

draft-ietf-snmpv2-bcm-ds-00.txt

Jeffrey D. Case
SNMP Research, Inc.
case@snmp.com

Keith McCloghrie
Cisco Systems, Inc.
kzm@cisco.com

Marshall T. Rose
Dover Beach Consulting, Inc.
mrose@dbc.mtview.ca.us

Steven Waldbusser
Carnegie Mellon University
waldbusser@cmu.edu

Status of this Memo

1. Introduction

A network management system contains: several (potentially many) nodes, each with a processing entity, termed an agent, which has access to management instrumentation; at least one management station; and, a management protocol, used to convey management information between the agents and management stations. Operations of the protocol are carried out under an administrative framework which defines both authentication and authorization policies.

Network management stations execute management applications which monitor and control network elements. Network elements are devices such as hosts, routers, terminal servers, etc., which are monitored and controlled through access to their management information.

The Administrative Infrastructure for SNMPv2 [1] defines how the administrative framework is applied to realize effective network management in a variety of configurations and environments. It is the purpose of this document, the Basic Configuration Model for SNMPv2, to define one such deployment strategy using the administrative framework.

1.1. A Note on Terminology

For the purpose of exposition, the original Internet-standard Network Management Framework, as described in RFCs 1155, 1157, and 1212, is termed the SNMP version 1 framework (SNMPv1). The current framework is termed the SNMP version 2 framework (SNMPv2).

2. Overview

The model described here is based on the notion of a basic set of well-known permanent configuration information for an SNMPv2 agent. This permanent information provides a basic set of SNMPv2 parties, a single SNMPv2 context, two views and access control information, by which management information held by the agent can be accessed.

This configuration information can be augmented during the lifetime of the agent through the addition of other temporary or permanent parties, contexts, views and access control information. However, the basic set must be configured at installation, and may not be deleted nor reduced in functionality below a minimum level of capability thereafter. By this means, a management application can be assured that the basic set always exists and is always available for use.

The basic set always includes four parties, and may include two more. The four parties are a pair of unauthenticated parties and a pair of authenticated parties. If the agent supports encryption, an additional pair of parties supporting both privacy and authentication can also be included. For each pair of parties, one party of the pair represents the agent's SNMPv2 entity, and the other represents an SNMPv2 entity it communicates with.

The naming conventions for the basic set of parties and contexts are organized to provide a framework for the naming of additional parties and contexts.

3. Naming Conventions

3.1. Party Identities

The convention for naming parties provides for a set of up to six parties to be used together, named as follows:

```
basicAgentNoAuthPartyID.<agentID>.<qualifier>
basicManagerNoAuthPartyID.<agentID>.<qualifier>
basicAgentAuthPartyID.<agentID>.<qualifier>
basicManagerAuthPartyID.<agentID>.<qualifier>
basicAgentPrivPartyID.<agentID>.<qualifier>
basicManagerPrivPartyID.<agentID>.<qualifier>
```

where:

<agentID>
: the 12-octet value of agentID.0 [2] for the agent, encoded one sub-identifier per octet.

<qualifier>
: An arbitrary length octet-string to produce a unique name for a party, encoded one sub-identifier per octet.

basicAgentNoAuthPartyID
: parties without authenticated or privacy, local to the agent,

basicManagerNoAuthPartyID
: parties without authenticated or privacy, remote from the agent,

basicAgentAuthPartyID
: parties with authenticated but not privacy, local to the agent,

basicManagerAuthPartyID
parties with authenticated but not privacy, remote from the agent,

basicAgentPrivPartyID
parties with authenticated and privacy, local to the agent,

basicManagerPrivPartyID
parties with authenticated and privacy, remote from the agent,

The <qualifier> allows unique party names so that each party is used by at most one manager entity at a time. Typically, one manager will communicate with an agent using a particular set of four or six parties which have that agent's agentID [2] and the same qualifier value.

3.2. Context Identities

The convention for naming contexts provides for the use of multiple context local entities as well as various of context temporal domains:

basicContextID.<agentID>.<clt>.<cle>

where:

<agentID>
the 12-octet value of agentID.0 [2] for the agent, encoded one sub-identifier per octet.

<clt>
identifies the context's temporal domain, encoded as a single sub-identifier, specifically the value of the last sub-identifier of a context temporal domain defined under temporalDomains [2]:

value	meaning
1	currentTime
2	restartTime

<cle>
identifies the context's local entity name, encoded as N sub-identifiers, one per octet of the value of contextLocalEntity [2] associated with the context. If contextLocalEntity is the empty string, then <cle> is effectively omitted from the context identity.

4. Installation Parameters

During the installation of an agent, several parameters must be configured. These are:

(1) a security posture

The choice of security posture determines the extent of the view configured for unauthenticated access. One of three possible choices is selected:

minimum-secure,
semi-secure, or
very-secure.

(2) one or more transport service addresses

These parameters (see partyTDomain and partyTAddress in [2]) may be specified explicitly, or they may be specified implicitly as the same set of network-layer addresses configured for other uses by the device together with the well-known transport-layer "port" information for the appropriate transport domain [3]. The agent listens on each of these transport service addresses for those parties which are included in the basic set and which are local to it.

(3) one or more secrets

These are the authentication/privacy secrets for the configured parties.

One way to accomplish this is to have the installer enter a "password" for each required secret. The password is then algorithmically converted into the required secret by: forming a string of length 1,048,576 octets by repeating the value of the password as often as necessary, truncating accordingly, and using the resulting string as the input to the MD5 algorithm. The resulting digest is the required secret (see Appendix A).

5. Mandatory Installation

An agent must instantiate the following parties, context, views and ACLs at the time the agent is installed. This configuration must persist, except that authentication secrets should be changed after installation.

5.1. Parties

Four parties with <qualifier> as the empty-string:

basicAgentNoAuthPartyID.<agentID>
basicManagerNoAuthPartyID.<agentID>
basicAgentAuthPartyID.<agentID>
basicManagerAuthPartyID.<agentID>

The parties are created with these values:

party	Agent noAuth	Manager noAuth	Agent Auth	Manager Auth
TDomain	tDomain	tDomain	tDomain	tDomain
TAddress	agtAddr	null	agtAddr	null
MaxMessageSize	<mms>	<mms>	<mms>	<mms>
Local	true	false	true	false
AuthProtocol	noAuth	noAuth	v2md5AuthProt	v2md5AuthProtocol
AuthClock	0	0	0	0
AuthPrivate	''H	''H	<aSecret>	<aSecret>
AuthPublic	''H	''H	''H	''H
PrivProtocol	noPriv	noPriv	noPriv	noPriv
PrivPrivate	''H	''H	''H	''H
PrivPublic	''H	''H	''H	''H
StorageType	permanent	permanent	permanent	permanent
Status	active	active	active	active

where:

<mms>
the standard maximum message size for the indicated transport domain [3].

<aSecret>
the configured authentication secret.

5.2. Contexts

One context with <cle> as the empty-string, and the <clt> value for currentTime:

```
basicContextID.<agentID>.1
```

The context is created with these values:

```
LocalEntity      ""
LocalTime        currentTime
ProxyDstParty    0.0
ProxySrcParty    0.0
ProxyContext     0.0
StorageType      readOnly
Status           active
Type             local
```

5.3. Views

Two views are configured as a set of view subtrees, one view for authenticated access and the other for unauthenticated access. The latter is configured according to the selected security posture.

For the "very-secure" posture, three view subtrees:

```
view          subtree #1   subtree #2    subtree #3
----          ----------   ----------    ----------
Index         <all>        <restricted>  <restricted>
Subtree       internet     snmpStats     snmpParties
Mask          ''H          ''H           ''H
Type          included     included      included
StorageType   readOnly     readOnly      readOnly
Status        active       active        active
```

For the "semi-secure" posture, four view subtrees:

view	subtree #1	subtree #2	subtree #3	subtree #4
Index	<all>	<restricted>	<restricted>	<restricted>
Subtree	internet	snmpStats	partyMIB	system
Mask	''H	''H	''H	''H
Type	included	included	included	included
StorageType	readOnly	readOnly	readOnly	readOnly
Status	active	active	active	active

For the "minimum-secure" posture, two view subtrees:

view	subtree #1	subtree #2
Index	<all>	<restricted>
Subtree	internet	internet
Mask	''H	''H
Type	included	included
StorageType	readOnly	readOnly
Status	active	active

5.4. ACLs

Two ACLs with these values:

ACL	ACL #1	ACL #2
Target	Agent NoAuth	Agent Auth
Subject	Manager NoAuth	Manager Auth
Context	the mandatory context	the mandatory context
Privileges	get/getNext/getBulk	get/getNext/getBulk/set
ReadViewIndex	<restricted>	<all>
WriteViewIndex	0	<all>
StorageType	readOnly	readOnly
Status	active	active

6. Optional Installation

If privacy is supported, the agent must also instantiate the following parties and ACLs at the time the agent is installed. This configuration must persist, except that authentication and privacy secrets should be changed after installation.

6.1. Parties

Two parties with <qualifier> as the empty-string:

```
basicAgentPrivPartyID.<agentID>
basicManagerPrivPartyID.<agentID>
```

The parties are created with these values:

party	Agent Priv	Manager Priv
TDomain	tDomain	tDomain
TAddress	agtAddr	null
MaxMessageSize	<mms>	<mms>
Local	true	false
AuthProtocol	v2md5AuthProtocol	v2md5AuthProtocol
AuthClock	0	0
AuthPrivate	<aSecret>	<aSecret>
AuthPublic	''H	''H
PrivProtocol	desPrivProtocol	desPrivProtocol
PrivPrivate	<pSecret>	<pSecret>
PrivPublic	''H	''H
StorageType	permanent	permanent
Status	active	active

where:

<mms>
the standard maximum message size for the indicated transport domain.

<aSecret>
the configured authentication secret.

<pSecret>
the configured privacy secret.

6.2. ACLs

One ACL with these values:

ACL	ACL #3
Target	Agent Priv
Subject	Manager Priv
Context	the mandatory context
Privileges	get/getNext/getBulk/set
ReadViewIndex	<all>
WriteViewIndex	<all>
StorageType	readOnly
Status	active

7. Agent Discovery

The basic configuration described in this memo facilitates communication between a manager and a discovered agent. On discovering that an agent executes a particular transport service address, a manager may proceed to:

(1) obtain the value of agentID.0 [2] using maintenance functions. (Note this only applies to non-proxied agents.)

(2) query the agent using:

```
dstParty     basicAgentNoAuthPartyID.<agentID>
srcParty     basicManagerNoAuthPartyID.<agentID>
context      basicContextID.<agentID>.1
```

to discover a <qualifier> for which there are a pair of parties

```
basicAgentAuthPartyID.<agentID>.<qualifier>
basicManagerAuthPartyID.<agentID>.<qualifier>
   and/or
basicAgentPrivPartyID.<agentID>.<qualifier>
basicManagerPrivPartyID.<agentID>.<qualifier>
```

having authentication/privacy secrets known to the manager.

(3) use the set of parties having the determined value of <qualifier> to obtain the set of contexts for which management information is accessible by the agent.

Whenever the manager needs to access the management information of any of these discovered contexts, it may then communicate with the agent using the set of parties having the determined value of <qualifier>.

Note that it is assumed that only one manager performs discovery at a time, and therefore the above procedure does not require the sharing of parties. This is ensured for the authenticated parties by providing the secrets to only one manager.

8. Definitions

```
SNMPv2-BCM-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, snmpModules
        FROM SNMPv2-SMI;

bcmMIB MODULE-IDENTITY
    LAST-UPDATED "9503180000Z"
    ORGANIZATION "IETF SNMPv2 Working Group"
    CONTACT-INFO
            "        Keith McCloghrie

             Postal: Cisco Systems, Inc.
                     170 West Tasman Drive,
                     San Jose, CA 95134-1706
                     US

                Tel: +1 408 526 5260

             E-mail: kzm@cisco.com"
    DESCRIPTION
            "The MIB module for the Basic Configuration Model."
    ::= { snmpModules 5 }
```

```
-- administrative assignments

bcmAdmin        OBJECT IDENTIFIER ::= { bcmMIB 1 }

-- parties

basicPartyID    OBJECT IDENTIFIER ::= { bcmAdmin 1 }

basicAgentNoAuthPartyID
                OBJECT IDENTIFIER ::= { basicPartyID 1 }
basicManagerNoAuthPartyID
                OBJECT IDENTIFIER ::= { basicPartyID 2 }
basicAgentAuthPartyID
                OBJECT IDENTIFIER ::= { basicPartyID 3 }
basicManagerAuthPartyID
                OBJECT IDENTIFIER ::= { basicPartyID 4 }
basicAgentPrivPartyID
                OBJECT IDENTIFIER ::= { basicPartyID 5 }
basicManagerPrivPartyID
                OBJECT IDENTIFIER ::= { basicPartyID 6 }

-- contexts

basicContextID  OBJECT IDENTIFIER ::= { bcmAdmin 2 }


END
```

9. Appendix A: Password to Key Algorithm

The following code fragment demonstrates the password to key algorithm used when mapping a password to an authentication or privacy key.

```
void password_to_key(password, passwordlen, key)
    u_char *password;       /* IN */
    u_int   passwordlen;    /* IN */
    u_char *key;            /* OUT - caller supplies pointer to 16
                               octet buffer */ {
    MDstruct    MD;
    u_char      *cp, password_buf[64];
    u_long      password_index = 0;
    u_long      count = 0, i;


    MDbegin(&MD);   /* initialize MD5 */

    /* loop until we've done 1 Megabyte */
    while (count < 1048576) {
        cp = password_buf;
        for(i = 0; i < 64; i++){
            *cp++ = password[ password_index++ % passwordlen ];
            /*
             * Take the next byte of the password, wrapping to the
             * beginning of the password as necessary.
             */
        }

        MDupdate(&MD, password_buf, 64 * 8);
        /*
         * 1048576 is divisible by 64, so the last MDupdate will be
         * aligned as well.
         */
        count += 64;
    }
    MDupdate(&MD, password_buf, 0); /* tell MD5 we're done */
    copy_digest_to_buffer(&MD, key);
    return; }
```

10. Acknowledgements

The authors wish to acknowledge the contributions of the SNMPv2 Working Group in general. In particular, the following individuals

Dave Arneson (Cabletron),
Uri Blumenthal (IBM),
Doug Book (Chipcom),
Maria Greene (Ascom Timeplex),
Deirdre Kostik (Bellcore),
Dave Harrington (Cabletron),
Jeff Johnson (Cisco Systems),
Brian O'Keefe (Hewlett Packard),
Dave Perkins (Bay Networks),
Randy Presuhn (Peer Networks),
Shawn Routhier (Epilogue),
Bob Stewart (Cisco Systems),
Kaj Tesink (Bellcore).

deserve special thanks for their contributions.

11. References

[1] Case, J., Galvin, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)", in progress, SNMP Research, Inc., Trusted Information Systems, Cisco Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, March, 1995.

[2] Case, J., Galvin, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)", in progress, SNMP Research, Inc., Trusted Information Systems, Cisco Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, March, 1995.

[3] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2)", in progress, SNMP Research Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., Carnegie Mellon University, March, 1995.

12. Security Considerations

Each authenticated party needs an authentication secret, and those which employ privacy also need a privacy secret. Appendix A specifies an algorithm by which the initial value of such a secret can be entered as a password. Such an algorithm simplifies the coordination required between a manager and an agent at installation time. However, once installation is complete, these secrets should be changed.

The algorithm in Appendix A is provided because human-generated passwords may be less than the 16 octets required by the MD5 authentication and DES privacy protocols, and because brute force attacks can be quite easy on a relatively short ASCII character set. Agent implementations (and agent configuration applications) must ensure that passwords are at least 8 characters in length.

Because these passwords are used (nearly) directly, it is important that they not be easily guessed. It is suggested that they be composed of mixed-case alphanumeric and punctuation characters that don't form words or phrases that might be found in a dictionary. Longer passwords improve the security of the system. Installers may wish to input multiword phrases to make the password string longer while ensuring that it is memorable.

13. Authors' Address

Jeffrey D. Case
SNMP Research, Inc.
3001 Kimberlin Heights Rd.
Knoxville, TN 37920-9716
US

Phone: +1 615 573 1434
Email: case@snmp.com

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive,
San Jose, CA 95134-1706

Phone: +1 408 526 5260
EMail: kzm@cisco.com

Marshall T. Rose
Dover Beach Consulting, Inc.
420 Whisman Court
Mountain View, CA 94043-2186
US

Phone: +1 415 968 1052
Email: mrose@dbc.mtview.ca.us

Steven Waldbusser
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
US

Phone: +1 412 268 6628
Email: waldbusser@cmu.edu

Table of Contents