ION Security Application Data Model
draft-birrane-dtn-adm-ionsec-01

Abstract

   This document describes the Application Data Model (ADM) for ION
   Security in compliance with the template provided by
   [I-D.birrane-dtn-adm].

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at https://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on September 12, 2019.

Table of Contents

1. Introduction

   An Application Data Model (ADM) provides a guaranteed interface for
   the management of an application or protocol in accordance with the
   Asynchronous Management Architecture (AMA) defined in
   [I-D.birrane-dtn-ama]. The ADM described in this document complies
   with the ADM Template provided in [I-D.birrane-dtn-adm] as encoded
   using the JSON syntax.

   The IONSEC Admin ADM provides the set of information necessary to
   configure and manage the ION security policy database on the local
   computer that is running ION. This information includes both
   authentication from Licklider Transmission Protocol (LTP) and Bundle
   Protocol Security (BPSEC).

1.1. Technical Notes

   o This document describes Version 0.0 of the IONSEC Admin ADM.

   o The AMM Resource Identifier (ARI) for this ADM is NOT correctly
      set. A sample ARI is used in this version of the specification
      and MAY change in future versions of this ADM until an ARI
      registry is established. This notice will be removed at that
      time.

   o Agent applications MAY choose to ignore the name, description, or
      other annotative information associated with the component
      definitions within this ADM where such items are only used to
      provide human-readable information or are otherwise not necessary
      to manage a device.

1.2. Scope

This ADM specifies those components of the Asynchronous Management Model (AMM) common to the manqgement of any instance of an ION node.

Any Manager software implementing this ADM MUST perform the responsibilities of an AMA Manager as outlined in [I-D.birrane-dtn-adm] as they relate to the objects included in this document.

Any Agent software implementing this ADM MUST perform the responsibilities of an AMA Agent as outlined in [I-D.birrane-dtn-adm] as they relate to the objects included in this document.

1.3. Requirements Language

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 RFC 2119 [RFC2119].

2. Structure and Design of this ADM

The IONSEC Admin ADM's structure is in accordance to [I-D.birrane-dtn-adm]. This ADM contains metadata, table templates, and controls. Table Templates are column templates that will be followed by any instance of this table available in the network. They may not be created dynamically within the network by Managers. Controls are predefined and sometimes parameterized opcodes that can be run on an Agent. Controls are preconfigured in Agents and Managers as part of ADM support. There are no variables, report templates, macros, edd, constants, or operators in this ADM at this time. The contents of this ADM are derived from the main functions and data that are needed to configure the security policy database on the local computer that is running ION and includes both Bundle Protocol Security and Licklider Transmission Protocol Authentication.

All ADMs have metadata that includes the name, namespace, and version of the ADM as well as the name of the organization that is issuing that particular ADM. This is important for identification purposes of the ADMs and to ensure version control.

The controls that were chosen to be expressed in this document are related to adding, deleting, and modifying security keys. The controls also deal with LTP segment authentication and LTP segment signing rules. The table templates expressed in this document show all of the keys and rules that are in the security policy database.

3. Naming and Identification

   This section outlines the namespaces used to uniquely identify ADM
   objects in this specification.

3.1. Namespace and Nicknames

   In accordance with [I-D.birrane-dtn-adm], every ADM is assigned a
   moderated Namespace. In accordance with [I-D.birrane-dtn-amp], these
   namespaces may be enumerated for compactness. The namespace and ADM
   identification for these objects is defined as follows.

```
         +-----------------+---------------------+
         |    Identifier   |        Value        |
         +-----------------+---------------------+
         |    Namespace    | DTN/ION/ionsecadmin |
         |                 |                     |
         | ADM Enumeration |          8          |
         +-----------------+---------------------+
```

                  Table 1: Namespace Information

   Given the above ADM enumeration, in accordance with
   [I-D.birrane-dtn-amp], the following AMP nicknames are defined.

```
           +----------+-----------------------------+
           | Nickname |          Collection         |
           +----------+-----------------------------+
           |   160    |  DTN/ION/ionsecadmin/Const  |
           |          |                             |
           |   161    |   DTN/ION/ionsecadmin/Ctrl  |
           |          |                             |
           |   162    |   DTN/ION/ionsecadmin/Edd   |
           |          |                             |
           |   163    |   DTN/ION/ionsecadmin/Mac   |
           |          |                             |
           |   164    |   DTN/ION/ionsecadmin/Oper  |
           |          |                             |
           |   165    |   DTN/ION/ionsecadmin/Rptt  |
           |          |                             |
           |   167    |   DTN/ION/ionsecadmin/Tblt  |
           |          |                             |
           |   169    |   DTN/ION/ionsecadmin/Var   |
           |          |                             |
           |   170    |   DTN/ION/ionsecadmin/Mdat  |
           |          |                             |
           | 171-179  | DTN/ION/ionsecadmin/Reserved|
           +----------+-----------------------------+
```

                   Table 2: IONSEC ADM Nicknames

4. IONSEC ADM JSON Encoding

   The following is the JSON encoding of the IONsec Admin ADM:

```
  {
    "Mdat": [{
        "name": "name",
        "type": "STR",
        "value": "ionsec_admin",
        "description": "The human-readable name of the ADM."
      },
      {
        "name": "namespace",
        "type": "STR",
        "value": "DTN/ION/ionsecadmin",
        "description": "The namespace of the ADM."
      },
      {
        "name": "version",
        "type": "STR",
        "value": "v0.0",
```

```
      "description": "The version of the ADM."
    },
    {
      "name": "organization",
      "type": "STR",
      "value": "JHUAPL",
      "description": "The name of the issuing organization of the
                      ADM."
    }
  ],

  "Tblt": [{
      "name": "ltp_rx_rules",
      "columns": [{
        "type": "UINT",
        "name": "ltp_engine_id"
      }, {
        "type": "UINT",
        "name": "ciphersuite_nbr"
      }, {
        "type": "STR",
        "name": "key_name"
      }],
      "description": "This table lists all LTP segment authentication
                      rules in the security policy database."
    },
    {
      "name": "ltp_tx_rules",
      "columns": [{
        "type": "UINT",
        "name": "ltp_engine_id"
      }, {
        "type": "UINT",
        "name": "ciphersuite_nbr"
      }, {
        "type": "STR",
        "name": "key_name"
      }],
      "description": "This table lists all LTP segment signing rules
                      in the security policy database."
    }
  ],

  "Ctrl": [{
      "name": "key_add",
      "parmspec": [{
        "type": "STR",
        "name": "key_name"
```

```
        }, {
          "type": "BYTESTR",
          "name": "key_value"
        }],
        "description": "This control adds a named key value to the
                        security policy database. The content of
                        file_name is taken as the value of the key.
                        Named keys can be referenced by other elements
                        of the security policy database."
      },
      {
        "name": "key_change",
        "parmspec": [{
          "type": "STR",
          "name": "key_name"
        }, {
          "type": "BYTESTR",
          "name": "key_value"
        }],
        "description": "This control changes the value of the named key,
                        obtaining the new key value from the content of
                        file_name."
      },
      {
        "name": "key_del",
        "parmspec": [{
          "type": "STR",
          "name": "key_name"
        }],
        "description": "This control deletes the key identified by
                        name."
      },
      {
        "name": "ltp_rx_rule_add",
        "parmspec": [{
          "type": "UINT",
          "name": "ltp_engine_id"
        }, {
          "type": "UINT",
          "name": "ciphersuite_nbr"
        }, {
          "type": "STR",
          "name": "key_name"
        }],
        "description": "This control adds a rule specifying the manner
                        in which LTP segment authentication will be
                        applied to LTP segmentsrecieved from the
                        indicated LTP engine. A segment from the
```

```
                            indicated LTP engine will only be deemed
                            authentic if it contains an authentication
                            extension computed via the ciphersuite
                            identified by ciphersuite_nbr using the
                            applicable key value. If ciphersuite_nbr is 255
                            then the applicable key value is a hard-coded
                            constant and key_name must be omitted; otherwise
                            key_nameis required and the applicable key
                            value is the current value of the key named
                            key_name in the local security policy database.
                            Valid values of ciphersuite_nbr are: 0:
                            HMAC-SHA1-80 1: RSA-SHA256 255: NULL"
      },
      {
        "name": "ltp_rx_rule_change",
        "parmspec": [{
          "type": "UINT",
          "name": "ltp_engine_id"
        }, {
          "type": "UINT",
          "name": "ciphersuite_nbr"
        }, {
          "type": "STR",
          "name": "key_name"
        }],
        "description": "This control changes the parameters of the LTP
                        segment authentication rule for the indicated
                        LTP engine."
      },
      {
        "name": "ltp_rx_rule_del",
        "parmspec": [{
          "type": "UINT",
          "name": "ltp_engine_id"
        }],
        "description": "This control deletes the LTP segment
                        authentication rule for the indicated LTP
                        engine."
      },
      {
        "name": "ltp_tx_rule_add",
        "parmspec": [{
          "type": "UINT",
          "name": "ltp_engine_id"
        }, {
          "type": "UINT",
          "name": "ciphersuite_nbr"
        }, {
```

```
        "type": "STR",
        "name": "key_name"
      }],
      "description": "This control adds a rule specifying the manner
                      in which LTP segments transmitted to the
                      indicated LTP engine mustbe signed. Signing a
                      segment destined for the indicated LTP engine
                      entails computing an authentication extension
                      via the ciphersuite identified by
                      ciphersuite_nbr using the applicable key value.
                      If ciphersuite_nbr is 255 then the applicable
                      key value is a hard-coded constant and key_name
                      must be omitted; otherwise key_nameis required
                      and the applicable key value is the current
                      value of the key named key_name in the local
                      security policy database.Valid values of
                      ciphersuite_nbr are: 0:HMAC_SHA1-80
                      1: RSA_SHA256 255: NULL"
    },
    {
      "name": "ltp_tx_rule_change",
      "parmspec": [{
        "type": "UINT",
        "name": "ltp_engine_id"
      }, {
        "type": "UINT",
        "name": "ciphersuite_nbr"
      }, {
        "type": "STR",
        "name": "key_name"
      }],
      "description": "This control changes the parameters of the LTP
                      segment signing rule for the indicated LTP
                      engine."
    },
    {
      "name": "ltp_tx_rule_del",
      "parmspec": [{
        "type": "UINT",
        "name": "ltp_engine_id"
      }],
      "description": "This control deletes the LTP segment signing
                      rule for the indicated LTP engine."
    },
    {
      "name": "list_keys",
      "description": "This control lists the names of keys available
                      in the key policy database."
```

```
      },
      {
        "name": "list_ltp_rx_rules",
        "description": "This control lists all LTP segment
                        authentication rules in the security policy
                        database."
      },
      {
        "name": "list_ltp_tx_rules",
        "description": "This control lists all LTP segment signing
                        rules in the security policy database."
      }
    ]
  }
```

5. IANA Considerations

   At this time, this protocol has no fields registered by IANA.

6. References

6.1. Informative References

   [I-D.birrane-dtn-ama]
              Birrane, E., "Asynchronous Management Architecture",
              draft-birrane-dtn-ama-07 (work in progress), June 2018.

6.2. Normative References

   [I-D.birrane-dtn-adm]
              Birrane, E., DiPietro, E., and D. Linko, "AMA Application
              Data Model", draft-birrane-dtn-adm-02 (work in progress),
              June 2018.

   [I-D.birrane-dtn-amp]
              Birrane, E., "Asynchronous Management Protocol", draft-
              birrane-dtn-amp-04 (work in progress), June 2018.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

Authors' Addresses

    Edward J. Birrane
    Johns Hopkins Applied Physics Laboratory

    Email: Edward.Birrane@jhuapl.edu

    Evana DiPietro
    Johns Hopkins Applied Physics Laboratory

    Email: Evana.DiPietro@jhuapl.edu

    David Linko
    Johns Hopkins Applied Physics Laboratory

    Email: David.Linko@jhuapl.edu