AMP/CAMP Walkthough
Ed Birrane

A set of slides because Ed can’t project his screen.

A few reminders

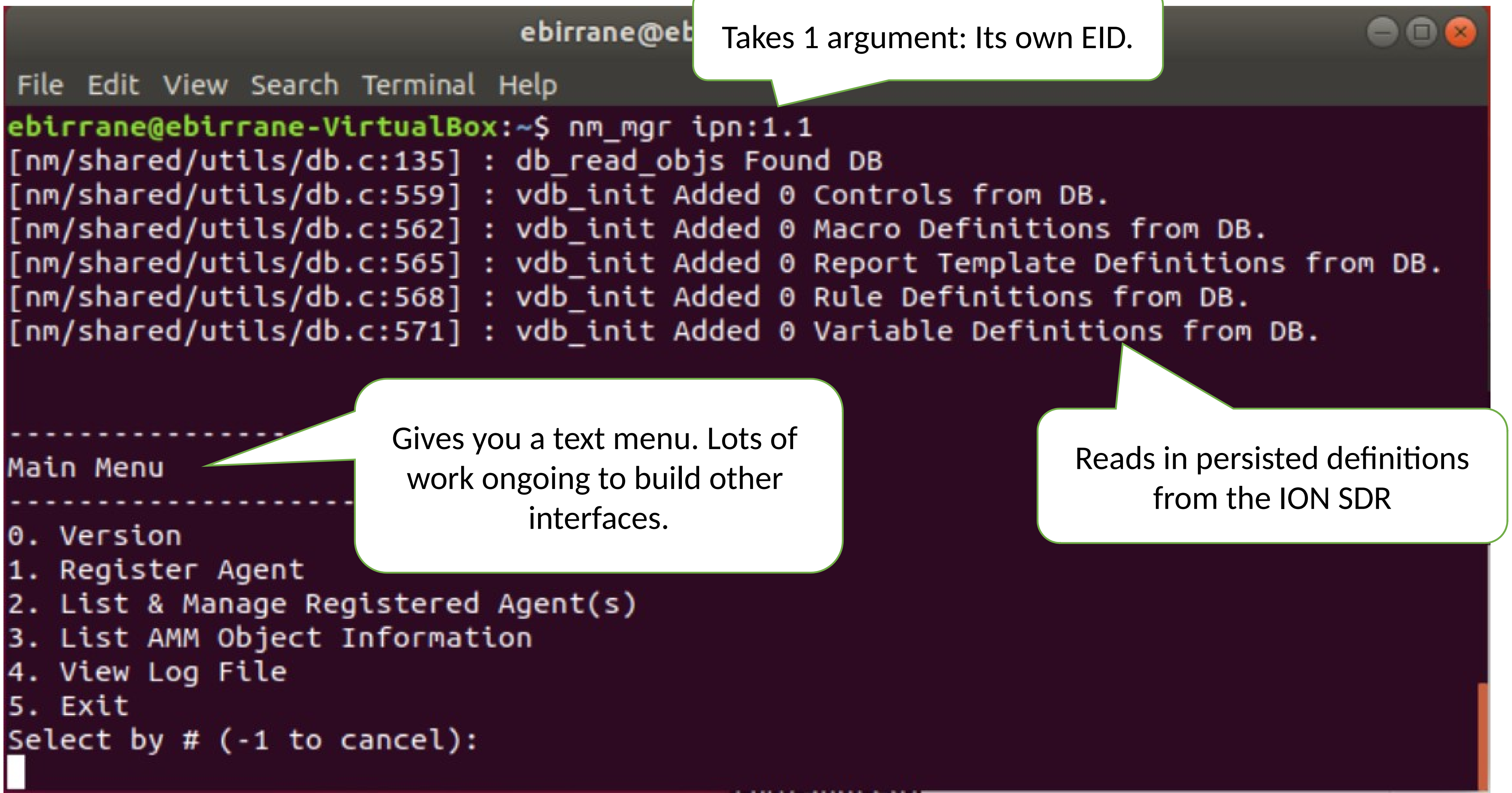
- Everything presented here is either
    - Codified in an I-D (personal)
    - Part of the ION open-source software distribution (available on sourceforge)

- We distinguish between AMA, ADM, and AMP.
    - AMA: the overall architecture. We have AMA managers and AMA agents
    - ADM: A data model. The ADM draft also includes a JSON encoding of the model.
    - AMP: A binary protocol. The CBOR encoding of the model.

- Within the ION open source distribution
    - ./nm is where AMP lives
    - ./nm/docs/adms is where JSON ADMs live
    - ./nm/contrib/CAmpPython is where CAMP lives (tarball you inflate and install)

- Building and installing ION will..
    - Build and install a text-interface AMA manager (nm_mgr)
    - Build and install an AMA agent (nm_agent)

Starting The Manager (once ION running)



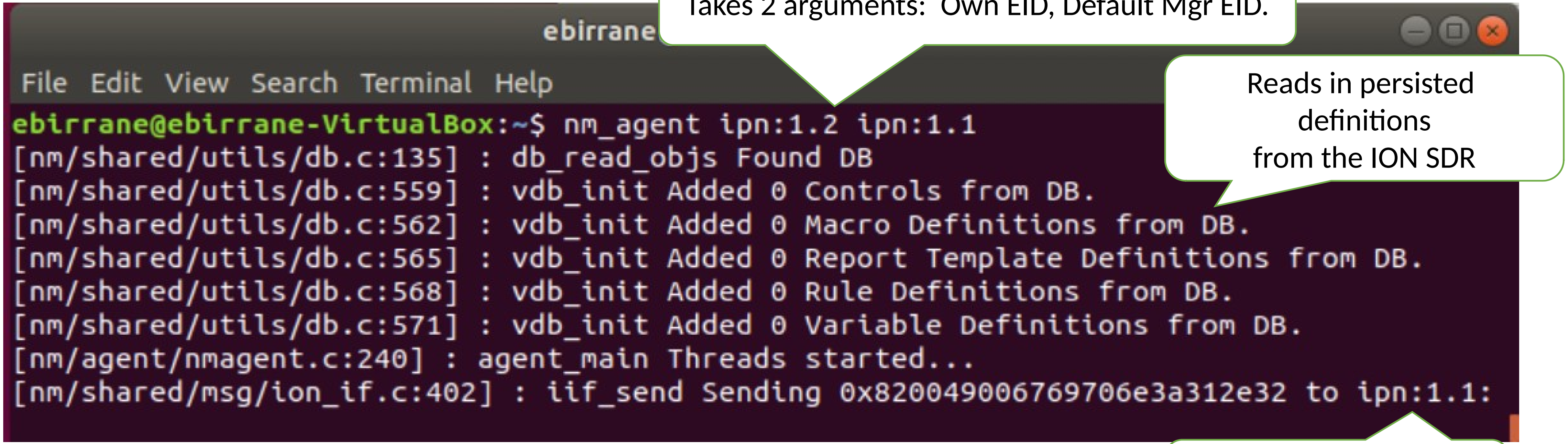
```
ebirrane@ebirrane-VirtualBox:~$ nm_mgr ipn:1.1
[nm/shared/utils/db.c:135] : db_read_objs Found DB
[nm/shared/utils/db.c:559] : vdb_init Added 0 Controls from DB.
[nm/shared/utils/db.c:562] : vdb_init Added 0 Macro Definitions from DB.
[nm/shared/utils/db.c:565] : vdb_init Added 0 Report Template Definitions from DB.
[nm/shared/utils/db.c:568] : vdb_init Added 0 Rule Definitions from DB.
[nm/shared/utils/db.c:571] : vdb_init Added 0 Variable Definitions from DB.


---------------
Main Menu
--------------------
0. Version
1. Register Agent
2. List & Manage Registered Agent(s)
3. List AMM Object Information
4. View Log File
5. Exit
Select by # (-1 to cancel):
```

Takes 1 argument: Its own EID.

Gives you a text menu. Lots of work ongoing to build other interfaces.

Reads in persisted definitions from the ION SDR

Starting the Agent (Once ION is running)



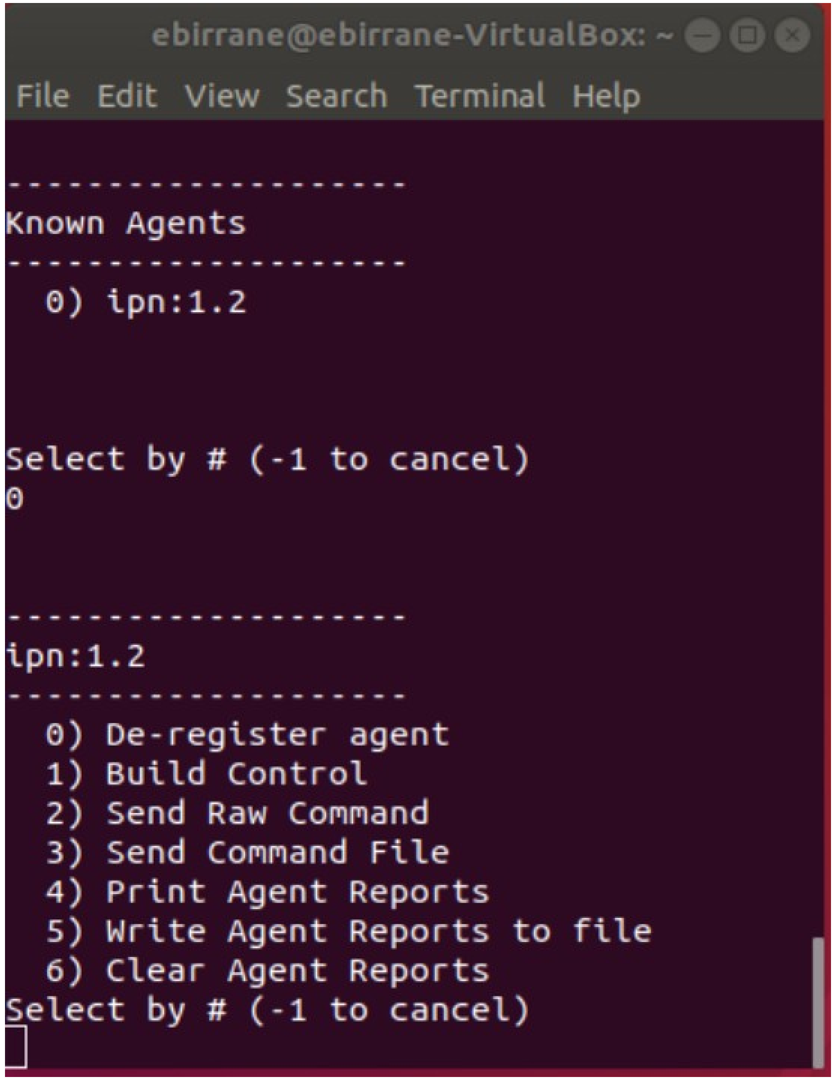
```
ebirrane@ebirrane-VirtualBox:~$ nm_agent ipn:1.2 ipn:1.1
[nm/shared/utils/db.c:135] : db_read_objs Found DB
[nm/shared/utils/db.c:559] : vdb_init Added 0 Controls from DB.
[nm/shared/utils/db.c:562] : vdb_init Added 0 Macro Definitions from DB.
[nm/shared/utils/db.c:565] : vdb_init Added 0 Report Template Definitions from DB.
[nm/shared/utils/db.c:568] : vdb_init Added 0 Rule Definitions from DB.
[nm/shared/utils/db.c:571] : vdb_init Added 0 Variable Definitions from DB.
[nm/agent/nmagent.c:240] : agent_main Threads started...
[nm/shared/msg/ion_if.c:402] : iif_send Sending 0x820049006769706e3a312e32 to ipn:1.1:
```

Takes 2 arguments: Own EID, Default Mgr EID.

Reads in persisted definitions
from the ION SDR

Sends register message
to default manager.

What can you manage on an Agent?


ebirrane@ebirrane-VirtualBox: ~
File Edit View Search Terminal Help

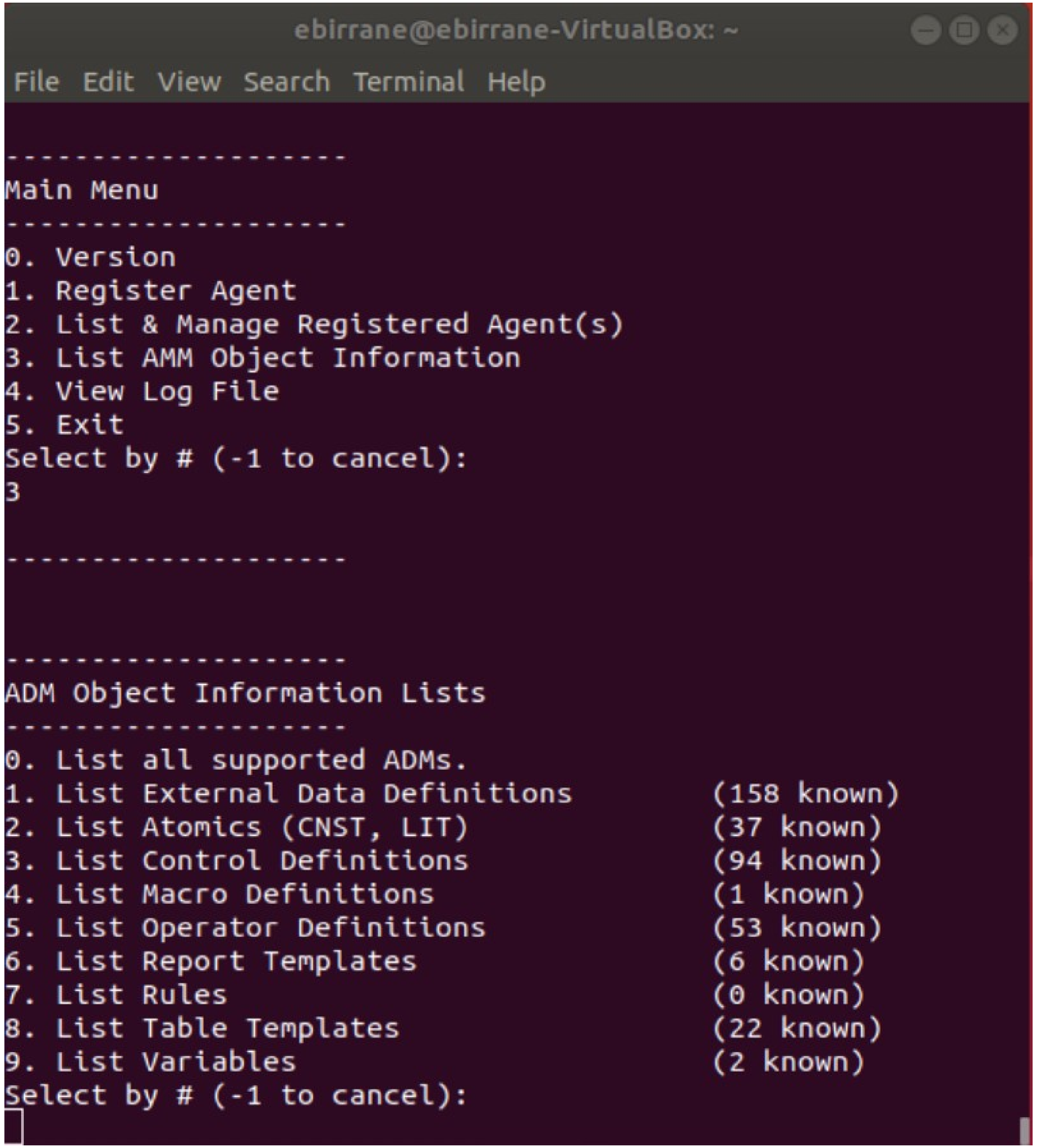
```
--------------------
Known Agents
--------------------
  0) ipn:1.2


Select by # (-1 to cancel)
0


--------------------
ipn:1.2
--------------------
  0) De-register agent
  1) Build Control
  2) Send Raw Command
  3) Send Command File
  4) Print Agent Reports
  5) Write Agent Reports to file
  6) Clear Agent Reports
Select by # (-1 to cancel)
```

- **De-Register**
  - Forget about this agent.
- **Build Control**
  - Tell the agent to do something (80% of use of this mgr)
- **Send Raw Command / Send Command File**
  - Send pre-built commands (hex) to agent.
  - This is an expert feature to aid in regression tests.
- **Print Agent Reports**
  - To screen, reports received by an agent.
- **Write Agent Reports to file**
  - Write to file.
- **Clear Agent Reports**
  - Remove from memory.

What does ION Support

- Agent ADM
  - Everything needed to control an AMA agent.
- ADMs for standard protocols
  - BP
  - LTP
  - BPSEC (SBSP)
- ION-Specific ADMs
  - Ionadmin
  - Bpadmin
  - Ipnadmin
  - Ionsecadmin
  - Ltpadmin

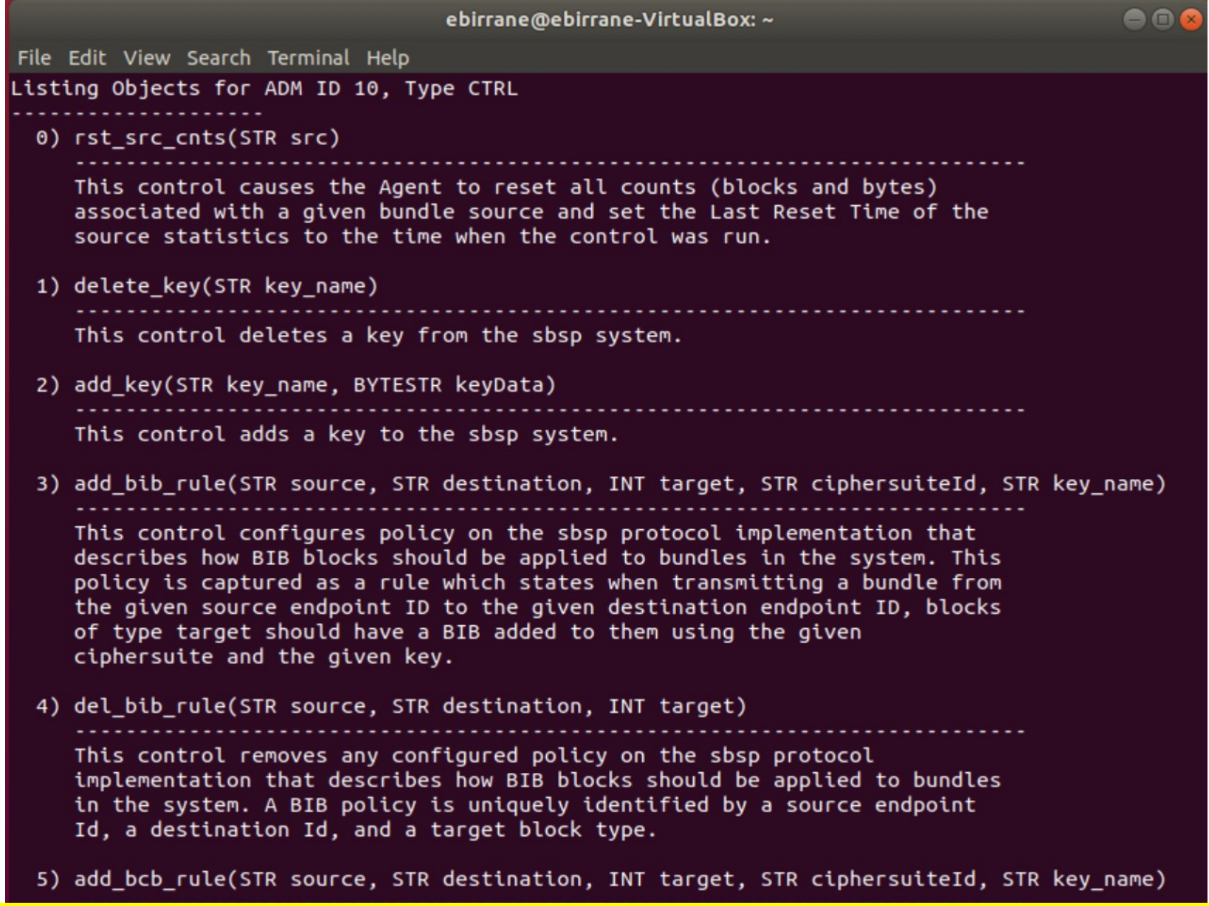
- What is an ADM again?
  - Encoding-agnostic model. Think MIB or YANG.
  - Has features not present in YANG
  - Can convert ADM <-> YANG where there is feature compatibility.


ebirrane@ebirrane-VirtualBox: ~
File Edit View Search Terminal Help

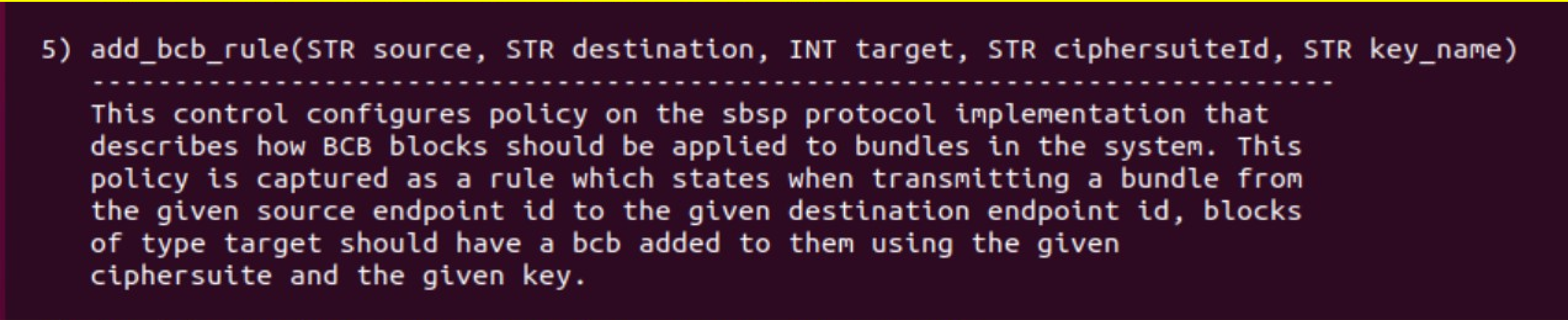
```
--------------------
Main Menu
--------------------
0. Version
1. Register Agent
2. List & Manage Registered Agent(s)
3. List AMM Object Information
4. View Log File
5. Exit
Select by # (-1 to cancel):
3


--------------------



--------------------
ADM Object Information Lists
--------------------
0. List all supported ADMs.
1. List External Data Definitions        (158 known)
2. List Atomics (CNST, LIT)              (37 known)
3. List Control Definitions              (94 known)
4. List Macro Definitions                (1 known)
5. List Operator Definitions             (53 known)
6. List Report Templates                 (6 known)
7. List Rules                            (0 known)
8. List Table Templates                  (22 known)
9. List Variables                        (2 known)
Select by # (-1 to cancel):
```

ION Security Controls

- 7 controls for security
  - Defined in the security ADM
  - Code for this auto-generated from the ADM files.
  - We will see this in a few slides when we get to CAMP.
- Example: Add BCB Rule
  - Will add a BCB rule as if it were commanded from the ionsecadmin CLI.


ebirrane@ebirrane-VirtualBox: ~
File Edit View Search Terminal Help

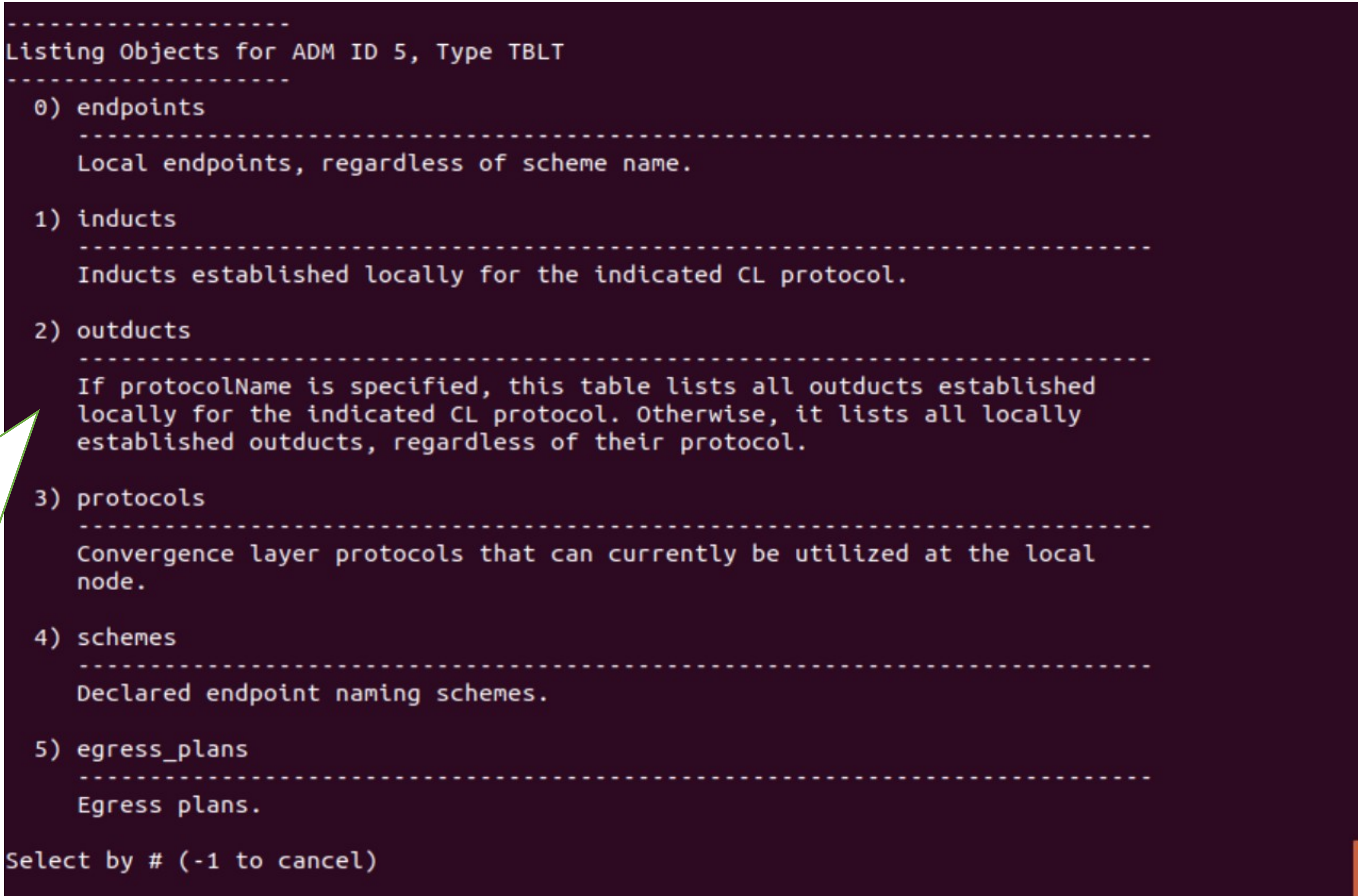
```
Listing Objects for ADM ID 10, Type CTRL
--------------------
  0) rst_src_cnts(STR src)
     --------------------------------------------------------------------------------
     This control causes the Agent to reset all counts (blocks and bytes)
     associated with a given bundle source and set the Last Reset Time of the
     source statistics to the time when the control was run.

  1) delete_key(STR key_name)
     --------------------------------------------------------------------------------
     This control deletes a key from the sbsp system.

  2) add_key(STR key_name, BYTESTR keyData)
     --------------------------------------------------------------------------------
     This control adds a key to the sbsp system.

  3) add_bib_rule(STR source, STR destination, INT target, STR ciphersuiteId, STR key_name)
     --------------------------------------------------------------------------------
     This control configures policy on the sbsp protocol implementation that
     describes how BIB blocks should be applied to bundles in the system. This
     policy is captured as a rule which states when transmitting a bundle from
     the given source endpoint ID to the given destination endpoint ID, blocks
     of type target should have a BIB added to them using the given
     ciphersuite and the given key.

  4) del_bib_rule(STR source, STR destination, INT target)
     --------------------------------------------------------------------------------
     This control removes any configured policy on the sbsp protocol
     implementation that describes how BIB blocks should be applied to bundles
     in the system. A BIB policy is uniquely identified by a source endpoint
     Id, a destination Id, and a target block type.

  5) add_bcb_rule(STR source, STR destination, INT target, STR ciphersuiteId, STR key_name)
```



```
5) add_bcb_rule(STR source, STR destination, INT target, STR ciphersuiteId, STR key_name)
   ----------------------------------------------------------------------
   This control configures policy on the sbsp protocol implementation that
   describes how BCB blocks should be applied to bundles in the system. This
   policy is captured as a rule which states when transmitting a bundle from
   the given source endpoint id to the given destination endpoint id, blocks
   of type target should have a bcb added to them using the given
   ciphersuite and the given key.
```

BP Admin Tables

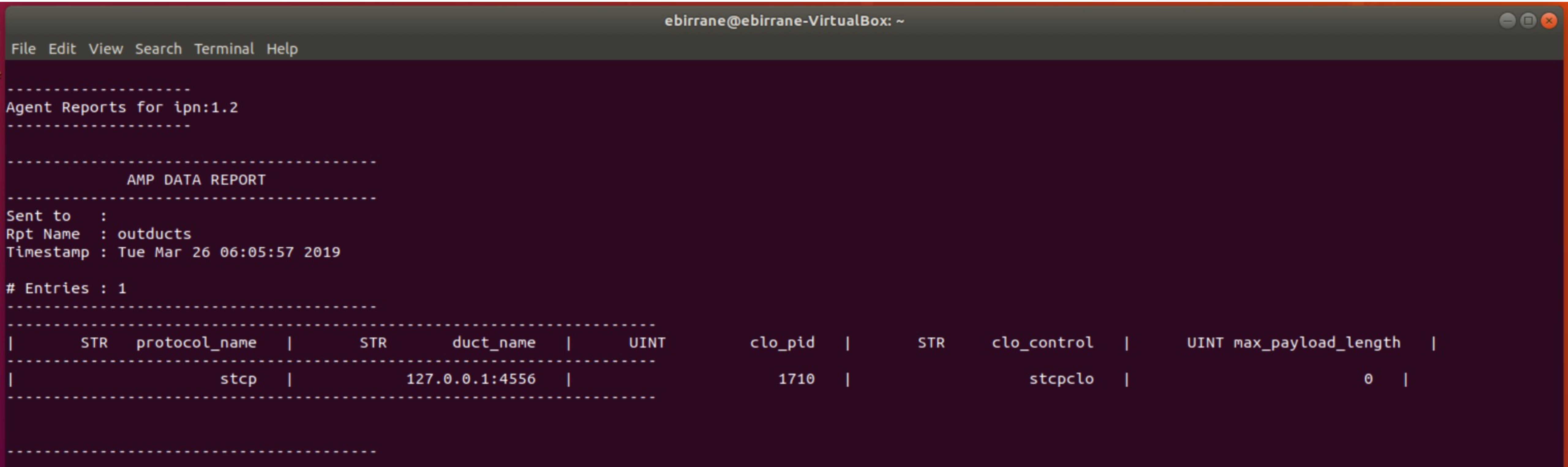
ADM for ION BpAdmin function includes definition of 6 tables.



```
--------------------
Listing Objects for ADM ID 5, Type TBLT
--------------------
  0) endpoints
     ---------------------------------------------------------------------------
     Local endpoints, regardless of scheme name.

  1) inducts
     ---------------------------------------------------------------------------
     Inducts established locally for the indicated CL protocol.

  2) outducts
     ---------------------------------------------------------------------------
     If protocolName is specified, this table lists all outducts established
     locally for the indicated CL protocol. Otherwise, it lists all locally
     established outducts, regardless of their protocol.

  3) protocols
     ---------------------------------------------------------------------------
     Convergence layer protocols that can currently be utilized at the local
     node.

  4) schemes
     ---------------------------------------------------------------------------
     Declared endpoint naming schemes.

  5) egress_plans
     ---------------------------------------------------------------------------
     Egress plans.

Select by # (-1 to cancel)
```

BP Admin Outduct Report

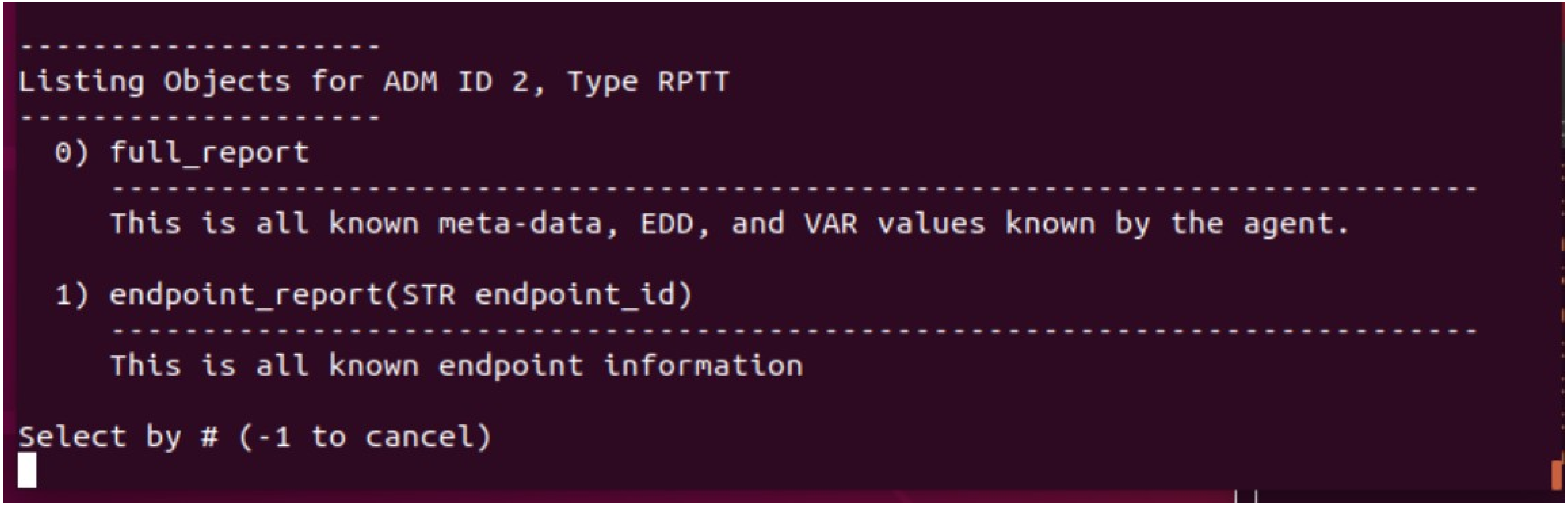
- AMP Agent has a control “Generate Table” which populates a table entry and returns it.
- This is printed by the Mgr as a (poorly ASCII formatted) table.
- Represents default outducts from a simple loopback scenario (./configs/loopback-stcp/loopback.rc)



```
--------------------
Agent Reports for ipn:1.2
--------------------

----------------------------------------
             AMP DATA REPORT
----------------------------------------
Sent to    :
Rpt Name   : outducts
Timestamp : Tue Mar 26 06:05:57 2019

# Entries : 1
----------------------------------------
----------------------------------------------------------------------
|       STR   protocol_name   |       STR       duct_name    |     UINT         clo_pid    |      STR     clo_control    |     UINT max_payload_length    |
----------------------------------------------------------------------
|                      stcp   |            127.0.0.1:4556    |                     1710    |                  stcpclo    |                        0   |
----------------------------------------------------------------------



----------------------------------------
```

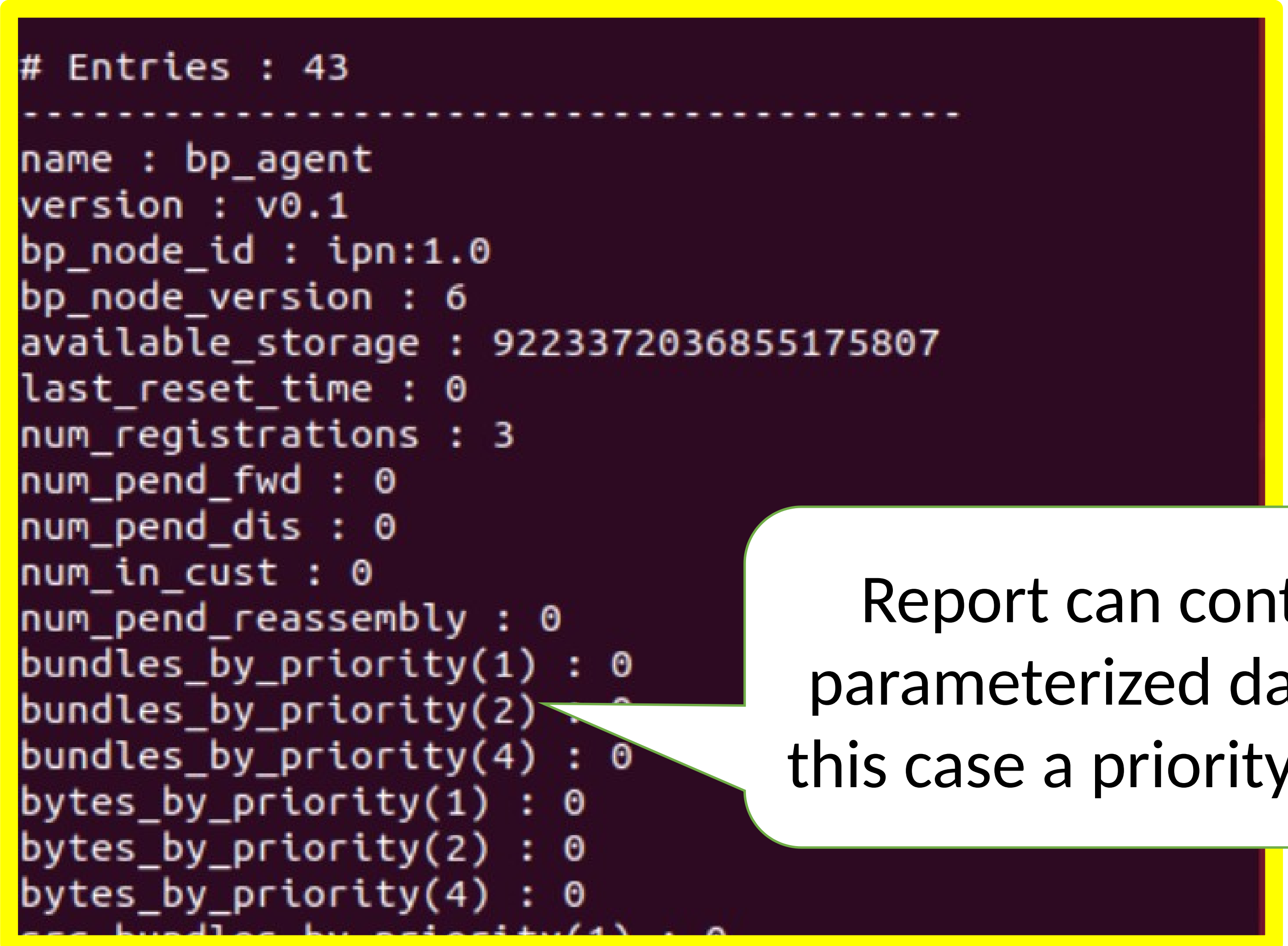
Report Templates for BP Agent

- General BP Agent defines 2 default reports
  - **Full Report (no parameters)** All BP Node data
  - **Endpoint Report (parameterized)** Info for a given endpoint.



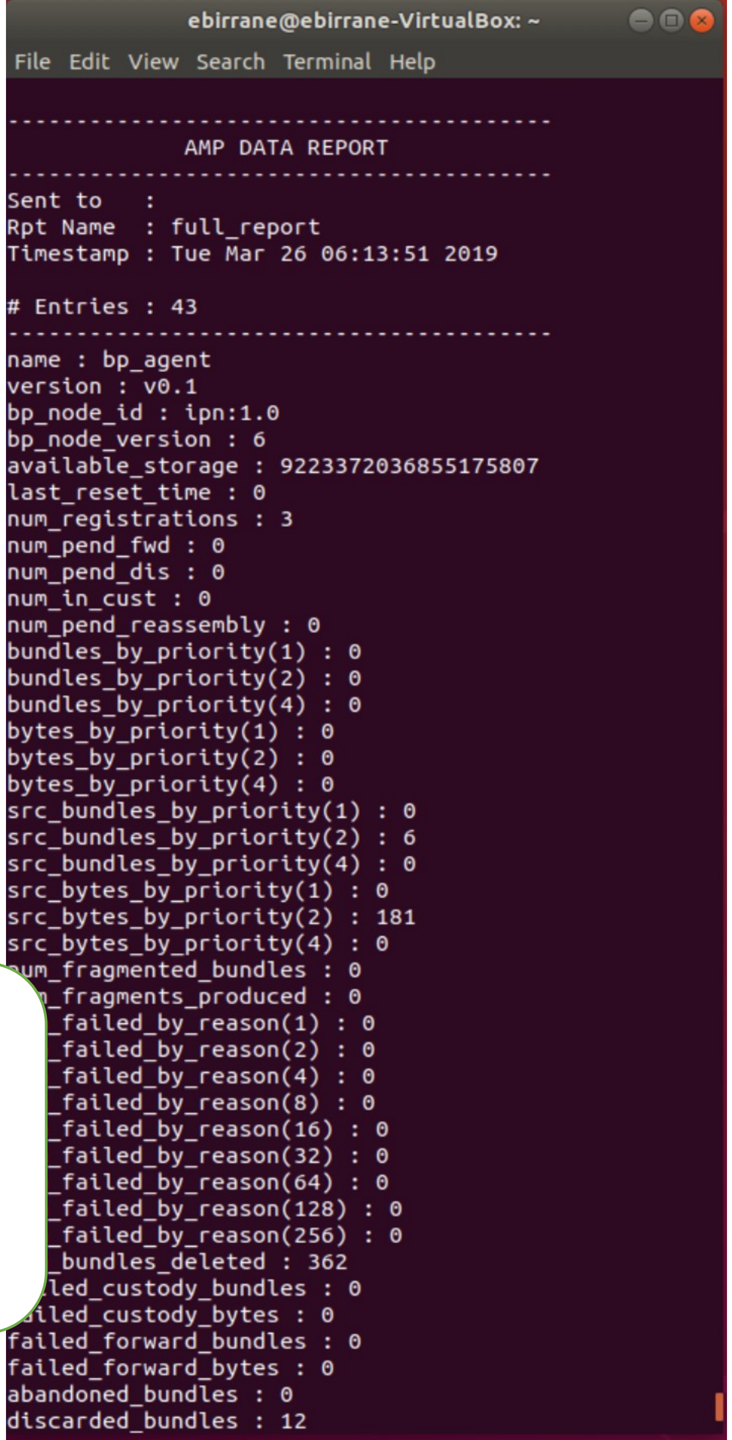
```
--------------------
Listing Objects for ADM ID 2, Type RPTT
--------------------
  0) full_report
     ------------------------------------------------------------------------------
     This is all known meta-data, EDD, and VAR values known by the agent.

  1) endpoint_report(STR endpoint_id)
     ------------------------------------------------------------------------------
     This is all known endpoint information

Select by # (-1 to cancel)
```

BP Full Report Example



```
# Entries : 43
----------------------------------------
name : bp_agent
version : v0.1
bp_node_id : ipn:1.0
bp_node_version : 6
available_storage : 9223372036855175807
last_reset_time : 0
num_registrations : 3
num_pend_fwd : 0
num_pend_dis : 0
num_in_cust : 0
num_pend_reassembly : 0
bundles_by_priority(1) : 0
bundles_by_priority(2)
bundles_by_priority(4) : 0
bytes_by_priority(1) : 0
bytes_by_priority(2) : 0
bytes_by_priority(4) : 0
src_bundles_by_priority(1) : 0
```

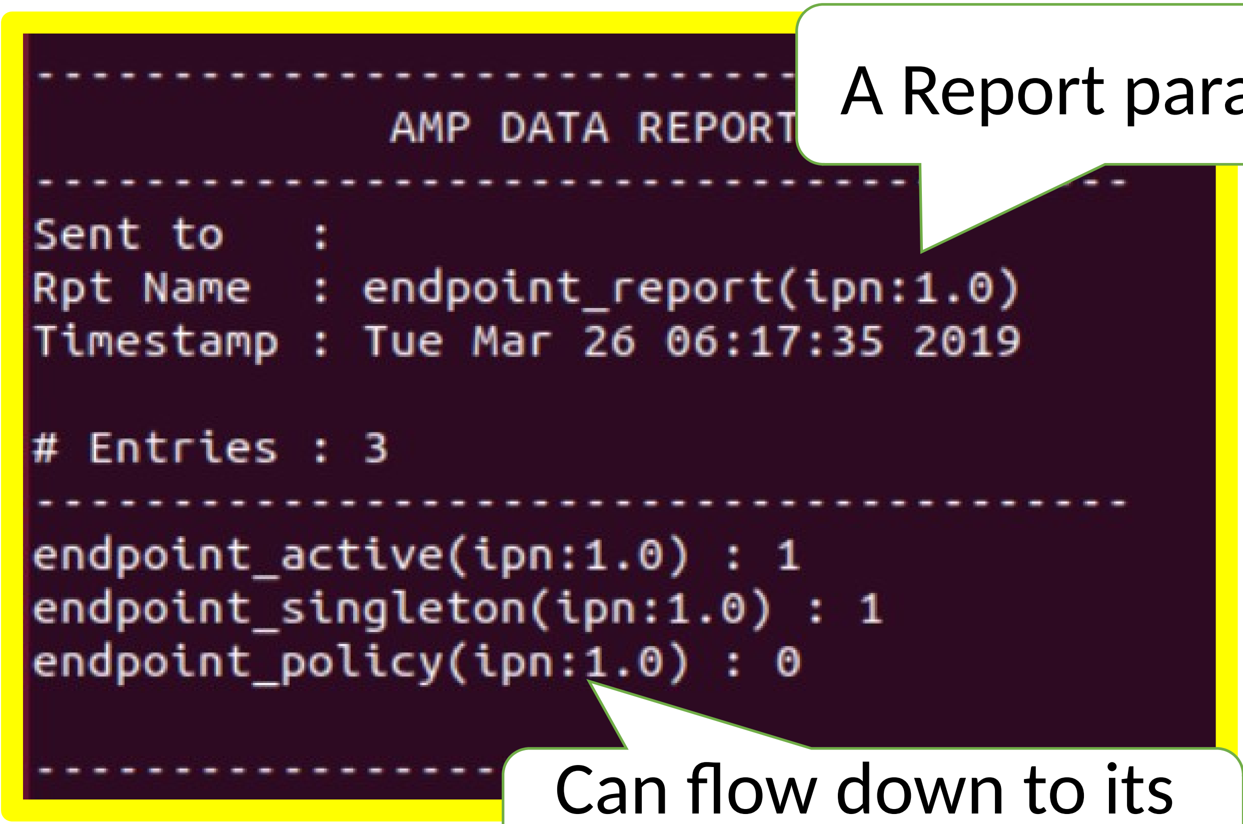
Report can contain parameterized data. In this case a priority mask.



ebirrane@ebirrane-VirtualBox: ~
File Edit View Search Terminal Help

```
----------------------------------------
             AMP DATA REPORT
----------------------------------------
Sent to    :
Rpt Name   : full_report
Timestamp  : Tue Mar 26 06:13:51 2019

# Entries : 43
----------------------------------------
name : bp_agent
version : v0.1
bp_node_id : ipn:1.0
bp_node_version : 6
available_storage : 9223372036855175807
last_reset_time : 0
num_registrations : 3
num_pend_fwd : 0
num_pend_dis : 0
num_in_cust : 0
num_pend_reassembly : 0
bundles_by_priority(1) : 0
bundles_by_priority(2) : 0
bundles_by_priority(4) : 0
bytes_by_priority(1) : 0
bytes_by_priority(2) : 0
bytes_by_priority(4) : 0
src_bundles_by_priority(1) : 0
src_bundles_by_priority(2) : 6
src_bundles_by_priority(4) : 0
src_bytes_by_priority(1) : 0
src_bytes_by_priority(2) : 181
src_bytes_by_priority(4) : 0
um_fragmented_bundles : 0
 _fragments_produced : 0
 _failed_by_reason(1) : 0
 _failed_by_reason(2) : 0
 _failed_by_reason(4) : 0
 _failed_by_reason(8) : 0
 _failed_by_reason(16) : 0
 _failed_by_reason(32) : 0
 _failed_by_reason(64) : 0
 _failed_by_reason(128) : 0
 _failed_by_reason(256) : 0
 _bundles_deleted : 362
 led_custody_bundles : 0
 iled_custody_bytes : 0
failed_forward_bundles : 0
failed_forward_bytes : 0
abandoned_bundles : 0
discarded_bundles : 12
```

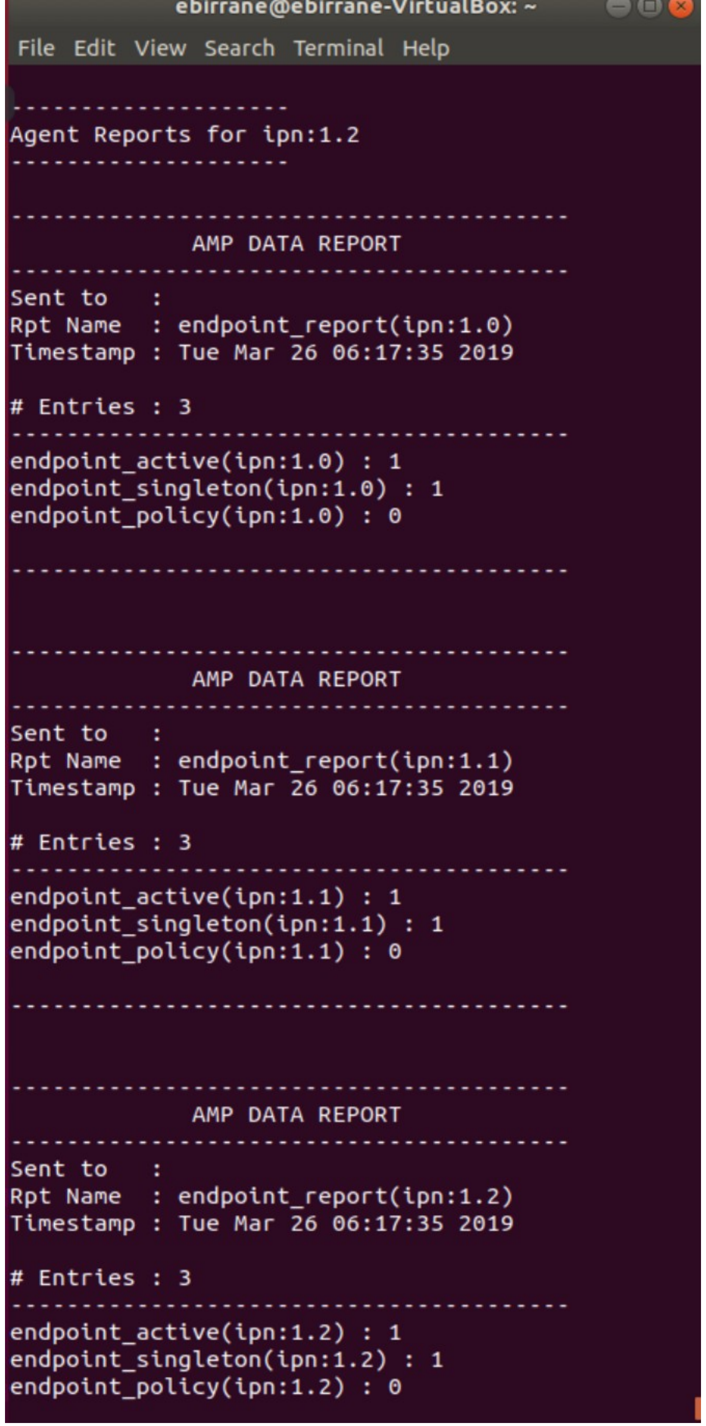
BP Endpoint Report



```
--------------------------------------
           AMP DATA REPORT
--------------------------------------
Sent to   :
Rpt Name  : endpoint_report(ipn:1.0)
Timestamp : Tue Mar 26 06:17:35 2019

# Entries : 3
--------------------------------------
endpoint_active(ipn:1.0) : 1
endpoint_singleton(ipn:1.0) : 1
endpoint_policy(ipn:1.0) : 0

--------------------------------------
```

A Report parameter…

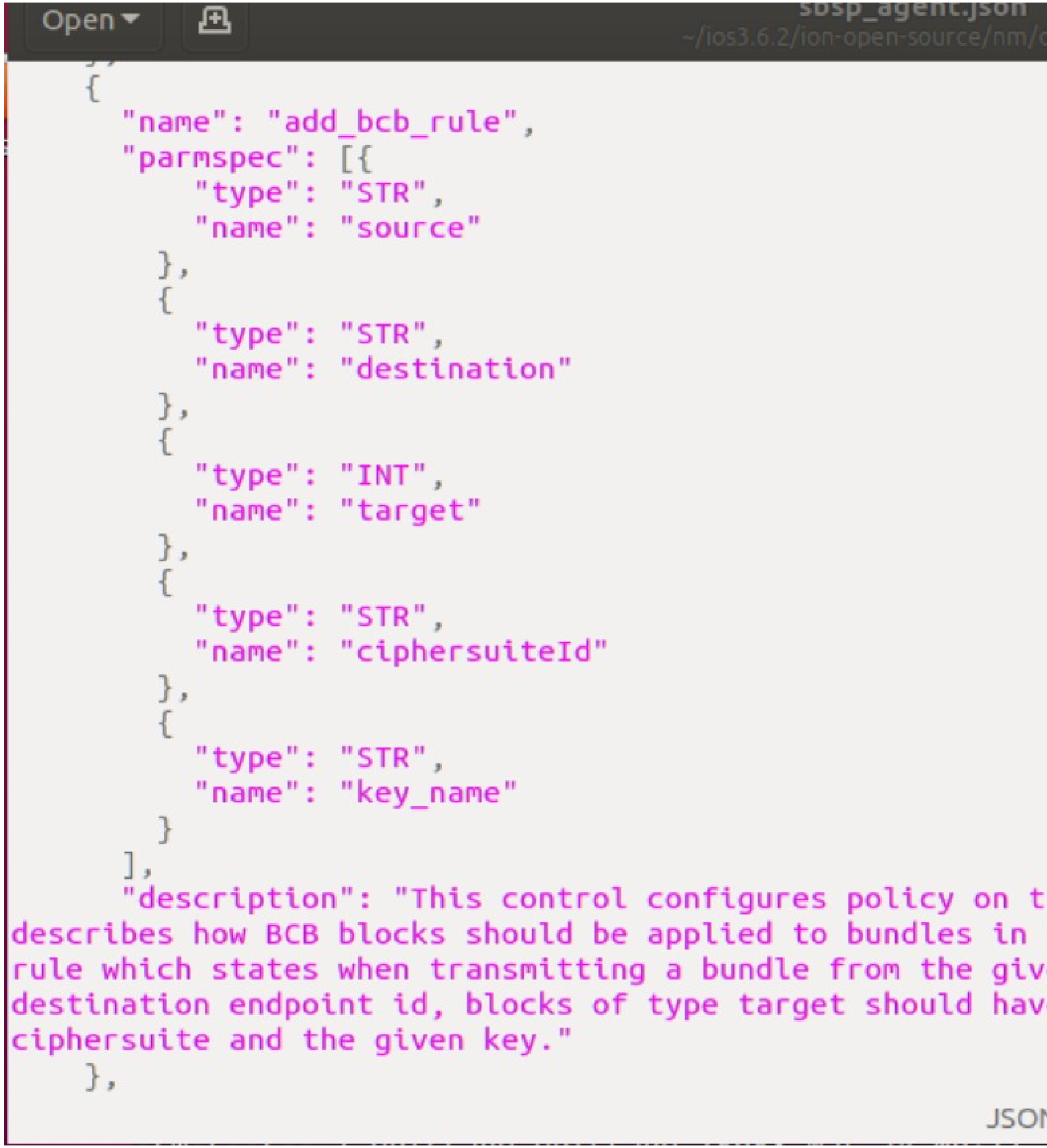
Can flow down to its entries.


ebirrane@ebirrane-VirtualBox: ~
File Edit View Search Terminal Help

```
-------------------
Agent Reports for ipn:1.2
-------------------

----------------------------------------
             AMP DATA REPORT
----------------------------------------
Sent to   :
Rpt Name  : endpoint_report(ipn:1.0)
Timestamp : Tue Mar 26 06:17:35 2019

# Entries : 3
----------------------------------------
endpoint_active(ipn:1.0) : 1
endpoint_singleton(ipn:1.0) : 1
endpoint_policy(ipn:1.0) : 0

----------------------------------------



----------------------------------------
             AMP DATA REPORT
----------------------------------------
Sent to   :
Rpt Name  : endpoint_report(ipn:1.1)
Timestamp : Tue Mar 26 06:17:35 2019

# Entries : 3
----------------------------------------
endpoint_active(ipn:1.1) : 1
endpoint_singleton(ipn:1.1) : 1
endpoint_policy(ipn:1.1) : 0

----------------------------------------



----------------------------------------
             AMP DATA REPORT
----------------------------------------
Sent to   :
Rpt Name  : endpoint_report(ipn:1.2)
Timestamp : Tue Mar 26 06:17:35 2019

# Entries : 3
----------------------------------------
endpoint_active(ipn:1.2) : 1
endpoint_singleton(ipn:1.2) : 1
endpoint_policy(ipn:1.2) : 0
```

The sbsp_agent.json

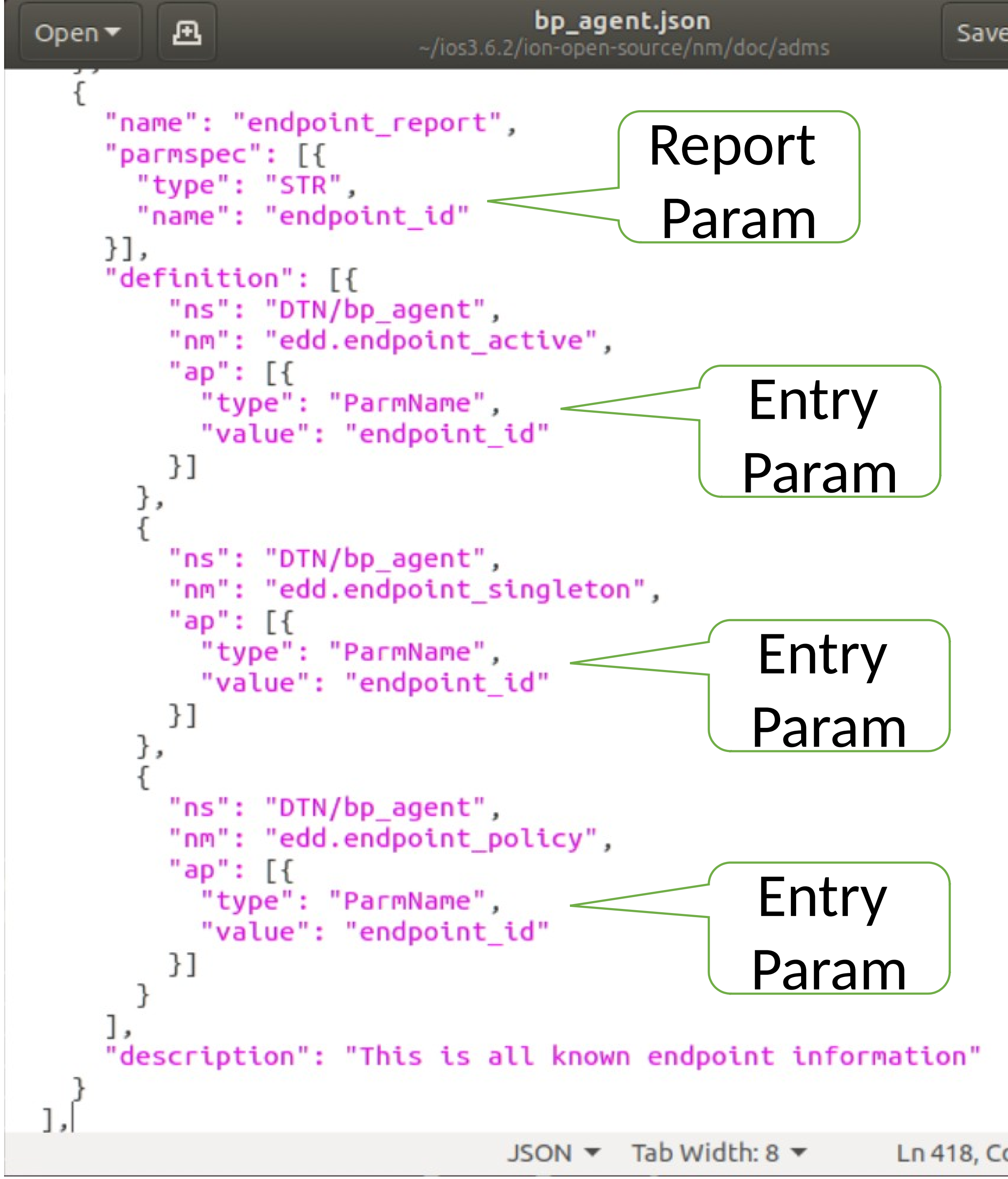
- ION Supports JSON-formatted ADMs
  - Template given in ADM I-D.
  - JSON encoding given in ADM I-D
- Example:
  - Control “Add BCB Rule”
  - 5 parameters
- When sending from SOURCE to DEST a block of type TARGET use a BCB with the given CIPHERSUITE and KEY NAME.



```
    {
      "name": "add_bcb_rule",
      "parmspec": [{
          "type": "STR",
          "name": "source"
        },
        {
          "type": "STR",
          "name": "destination"
        },
        {
          "type": "INT",
          "name": "target"
        },
        {
          "type": "STR",
          "name": "ciphersuiteId"
        },
        {
          "type": "STR",
          "name": "key_name"
        }
      ],
      "description": "This control configures policy on t
describes how BCB blocks should be applied to bundles in
rule which states when transmitting a bundle from the giv
destination endpoint id, blocks of type target should hav
ciphersuite and the given key."
    },
```

BP Endpoint Report

- Example of JSON for parameterized Report
- BP Endpoint Report



```
  {
    "name": "endpoint_report",
    "parmspec": [{
      "type": "STR",
      "name": "endpoint_id"
    }],
    "definition": [{
        "ns": "DTN/bp_agent",
        "nm": "edd.endpoint_active",
        "ap": [{
          "type": "ParmName",
          "value": "endpoint_id"
        }]
      },
      {
        "ns": "DTN/bp_agent",
        "nm": "edd.endpoint_singleton",
        "ap": [{
          "type": "ParmName",
          "value": "endpoint_id"
        }]
      },
      {
        "ns": "DTN/bp_agent",
        "nm": "edd.endpoint_policy",
        "ap": [{
          "type": "ParmName",
          "value": "endpoint_id"
        }]
      }
    ],
    "description": "This is all known endpoint information"
  }
],
```

Report Param

Entry Param

Entry Param

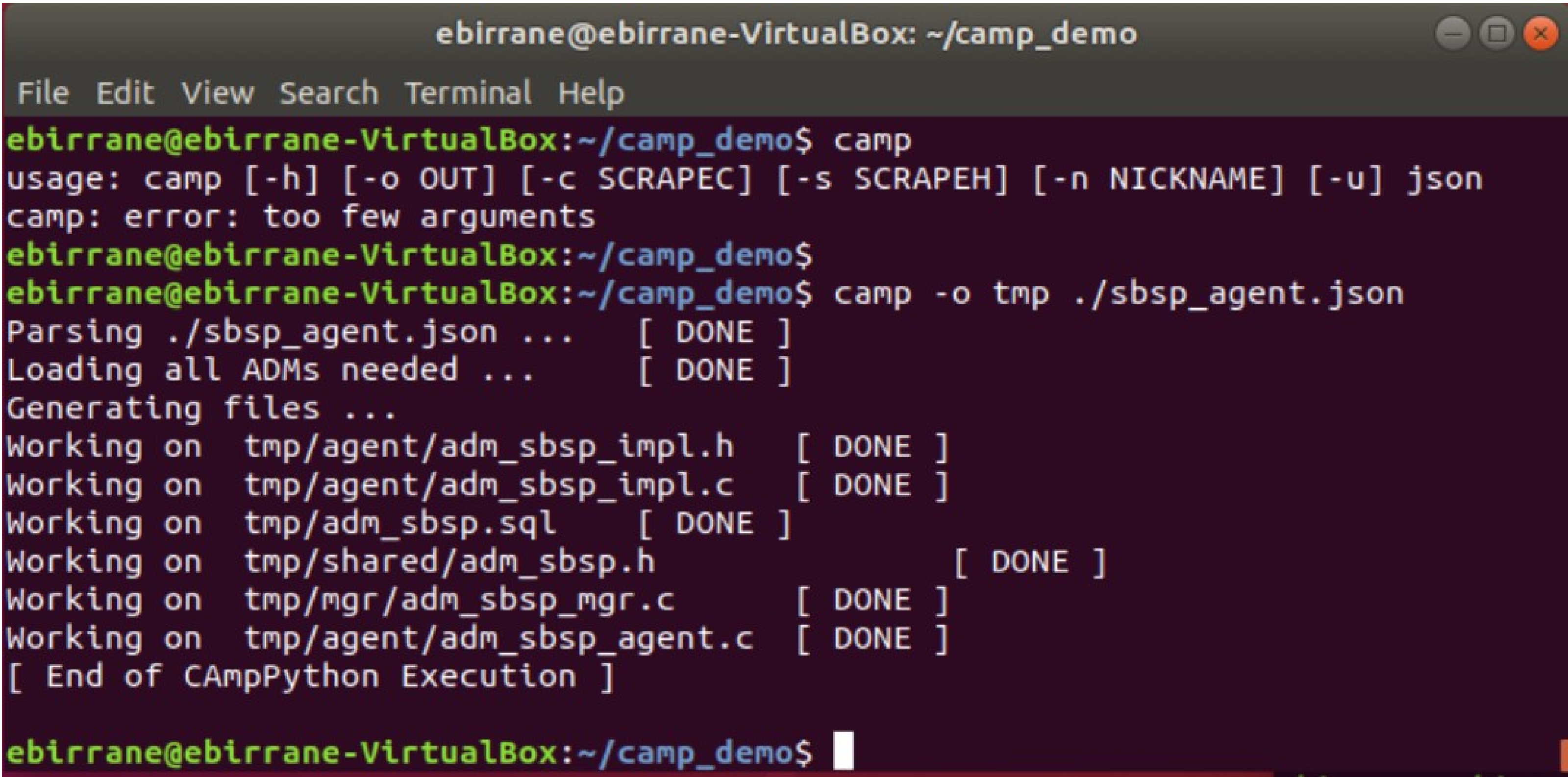
Entry Param

CAMP Python Scripts

- CAMP
  - C API Generator for AMP
  - Generated ION .c .h files to support new ADMs as they are added.
- User generates and adds code for functions, tables, and data collection.

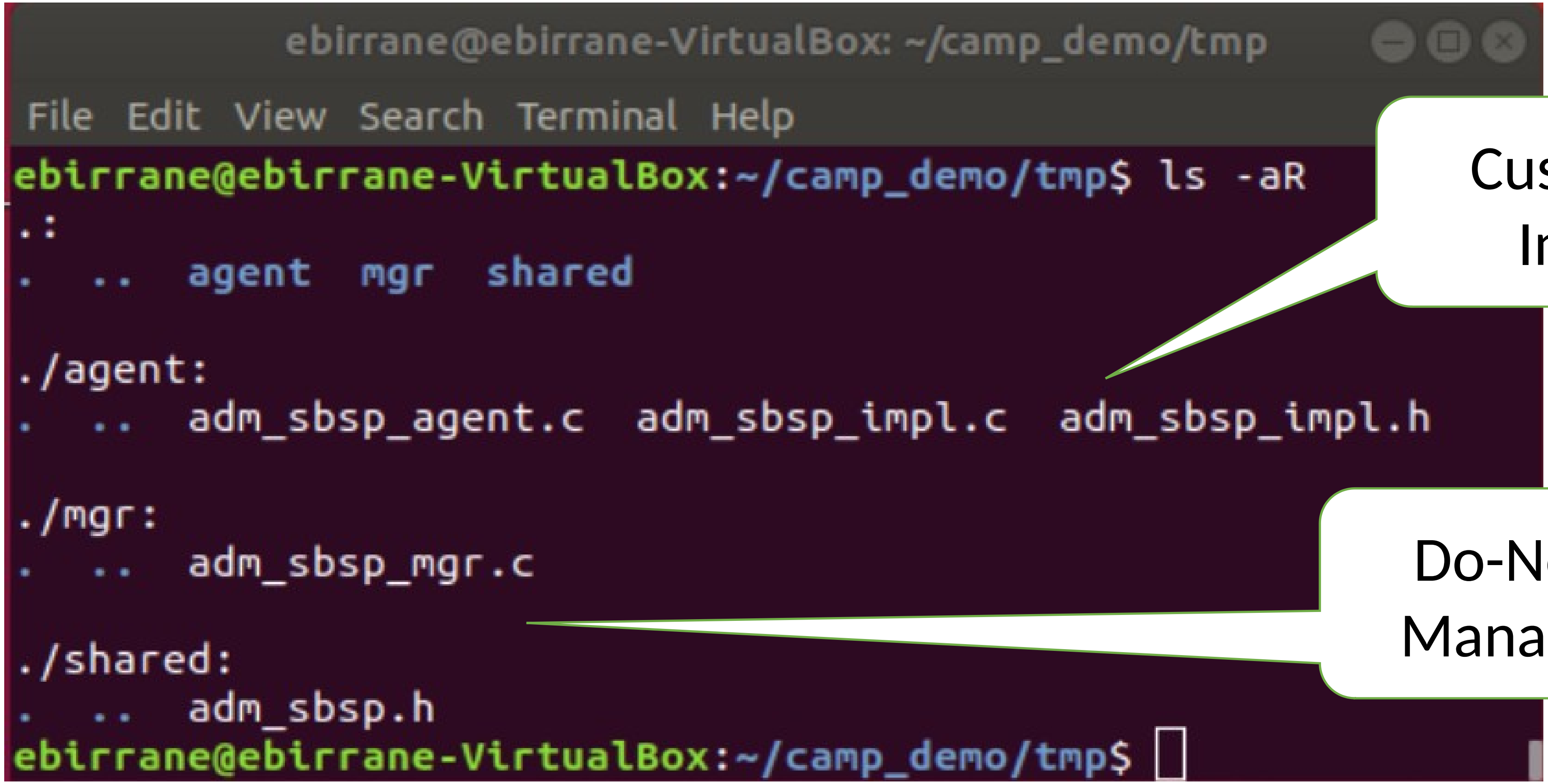
At minimum takes in JSON ADM. Can also scrape previous .h/.c to round-trip code.

Produces files to compile in to Agent and Mgr.


ebirrane@ebirrane-VirtualBox: ~/camp_demo
File Edit View Search Terminal Help

```
ebirrane@ebirrane-VirtualBox:~/camp_demo$ camp
usage: camp [-h] [-o OUT] [-c SCRAPEC] [-s SCRAPEH] [-n NICKNAME] [-u] json
camp: error: too few arguments
ebirrane@ebirrane-VirtualBox:~/camp_demo$
ebirrane@ebirrane-VirtualBox:~/camp_demo$ camp -o tmp ./sbsp_agent.json
Parsing ./sbsp_agent.json ...    [ DONE ]
Loading all ADMs needed ...      [ DONE ]
Generating files ...
Working on  tmp/agent/adm_sbsp_impl.h    [ DONE ]
Working on  tmp/agent/adm_sbsp_impl.c    [ DONE ]
Working on  tmp/adm_sbsp.sql     [ DONE ]
Working on  tmp/shared/adm_sbsp.h                [ DONE ]
Working on  tmp/mgr/adm_sbsp_mgr.c       [ DONE ]
Working on  tmp/agent/adm_sbsp_agent.c   [ DONE ]
[ End of CAmpPython Execution ]

ebirrane@ebirrane-VirtualBox:~/camp_demo$ 
```

CAMP Produces 5 files per ADM

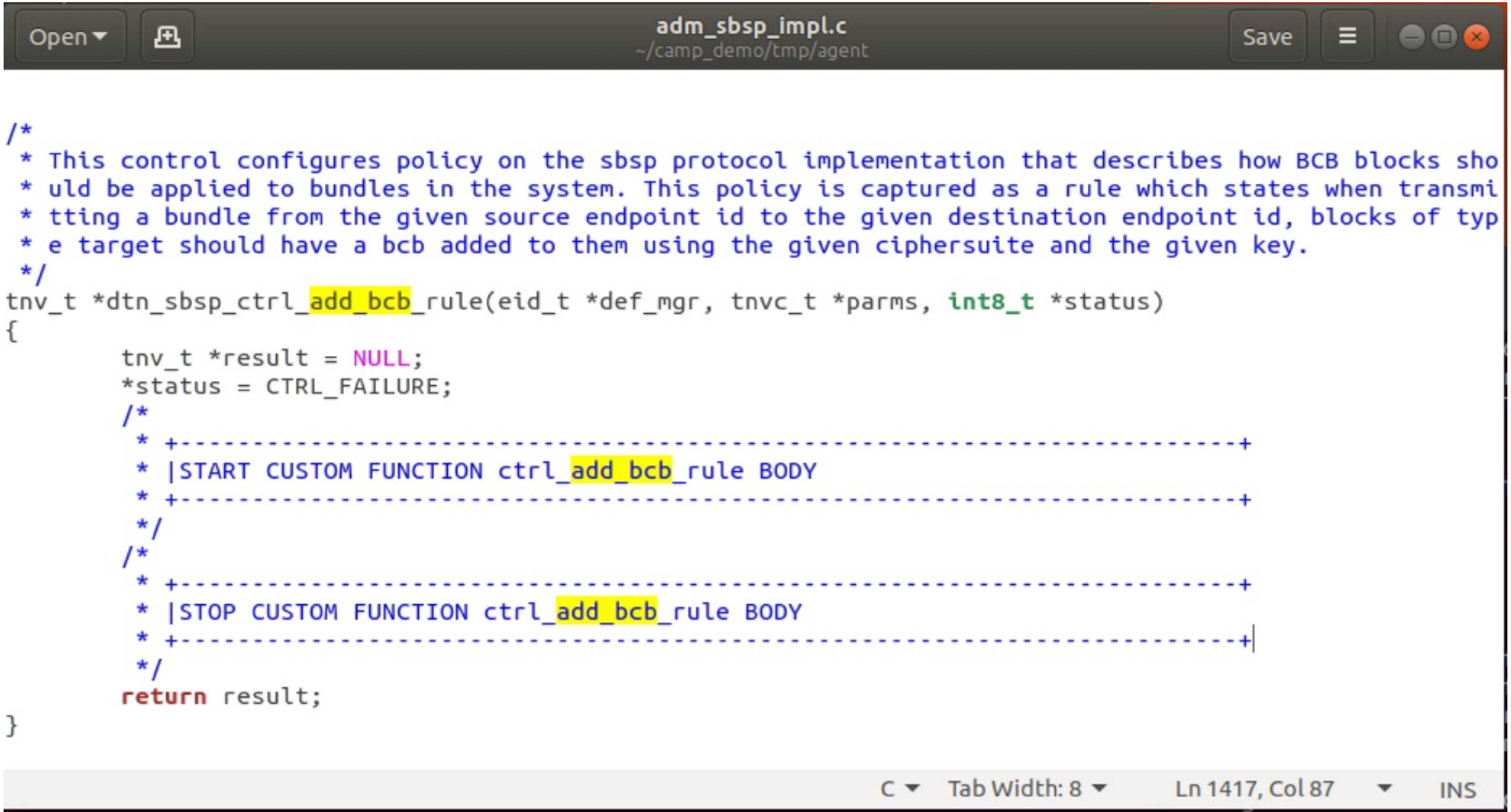

ebirrane@ebirrane-VirtualBox: ~/camp_demo/tmp
File Edit View Search Terminal Help

```
ebirrane@ebirrane-VirtualBox:~/camp_demo/tmp$ ls -aR
.:
.   ..   agent   mgr   shared

./agent:
.   ..   adm_sbsp_agent.c   adm_sbsp_impl.c   adm_sbsp_impl.h

./mgr:
.   ..   adm_sbsp_mgr.c

./shared:
.   ..   adm_sbsp.h
ebirrane@ebirrane-VirtualBox:~/camp_demo/tmp$ 
```

Customizable Agent Implementation.

Do-Not-Edit files for Manager and Shared.

CAMP auto-generated function example:

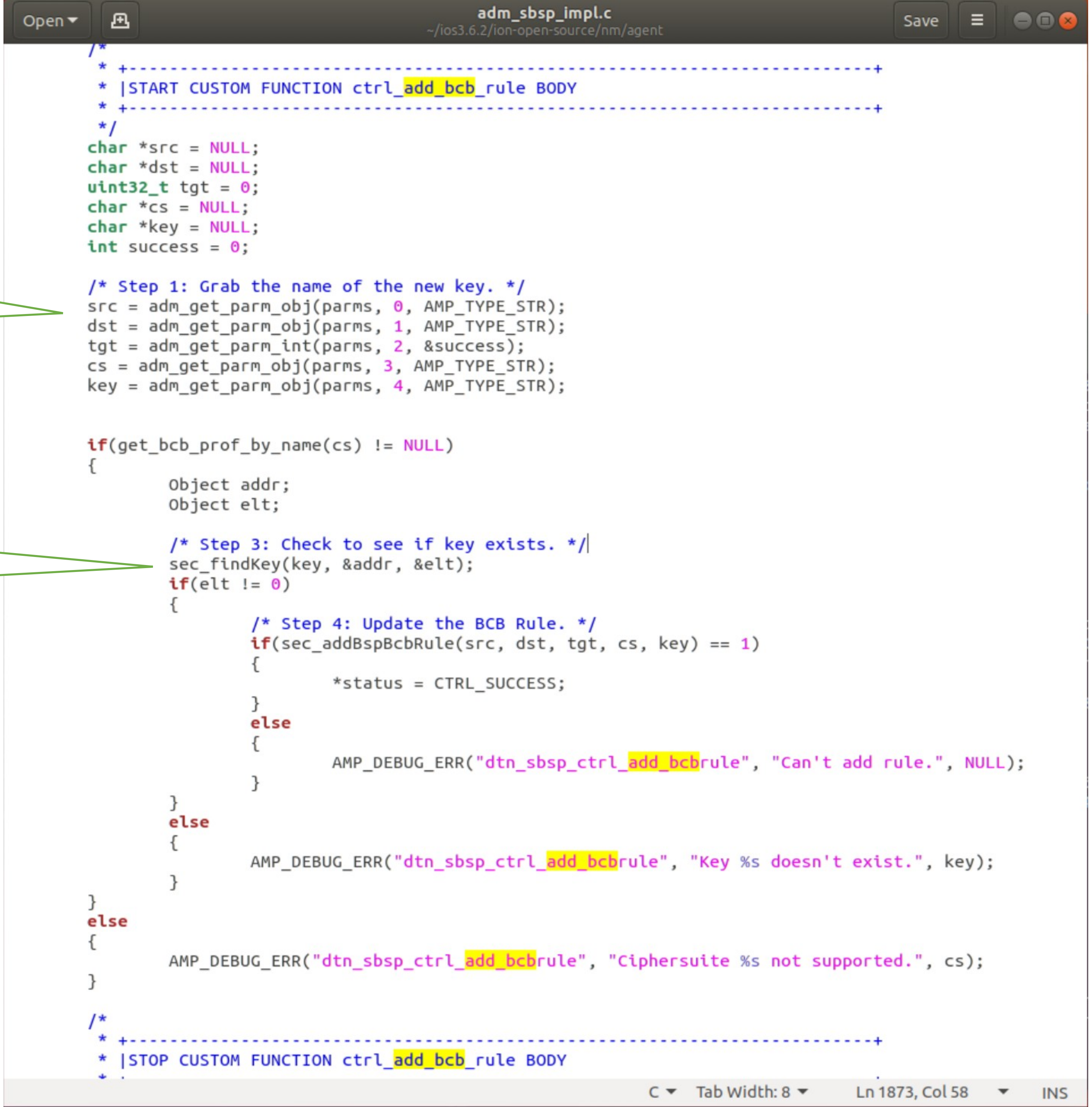


```
/*
 * This control configures policy on the sbsp protocol implementation that describes how BCB blocks sho
 * uld be applied to bundles in the system. This policy is captured as a rule which states when transmi
 * tting a bundle from the given source endpoint id to the given destination endpoint id, blocks of typ
 * e target should have a bcb added to them using the given ciphersuite and the given key.
 */
tnv_t *dtn_sbsp_ctrl_add_bcb_rule(eid_t *def_mgr, tnvc_t *parms, int8_t *status)
{
        tnv_t *result = NULL;
        *status = CTRL_FAILURE;
        /*
         * +-------------------------------------------------------------------------+
         * |START CUSTOM FUNCTION ctrl_add_bcb_rule BODY
         * +-------------------------------------------------------------------------+
         */
        /*
         * +-------------------------------------------------------------------------+
         * |STOP CUSTOM FUNCTION ctrl_add_bcb_rule BODY
         * +-------------------------------------------------------------------------+
         */
        return result;
}
```

- Example of populated function to add a BCB rule.

ION APIs for param verification.

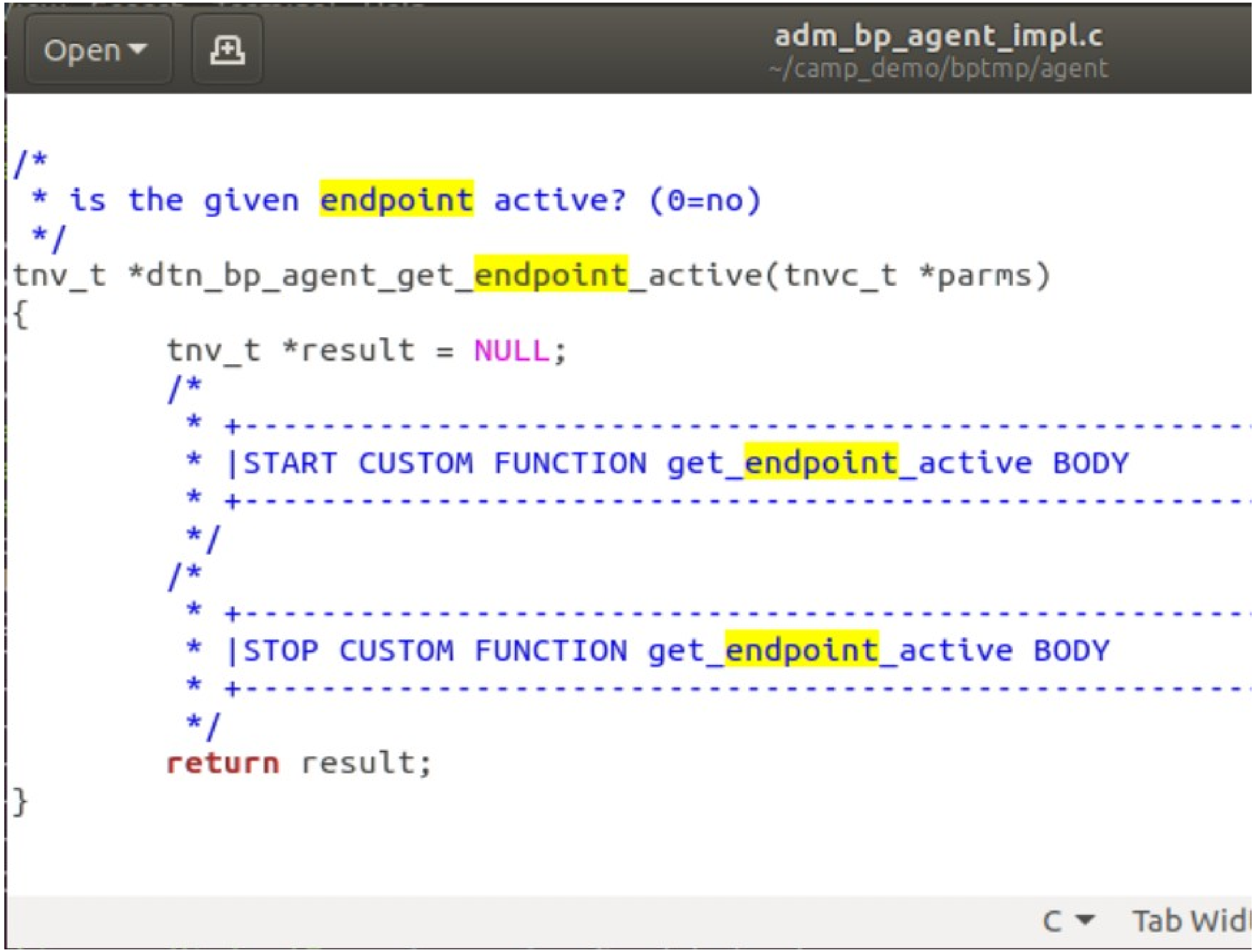
Wraps calls to ionsecadmin functions.



```c
/*
 * +----------------------------------------------------------------------------+
 * |START CUSTOM FUNCTION ctrl_add_bcb_rule BODY
 * +----------------------------------------------------------------------------+
 */
char *src = NULL;
char *dst = NULL;
uint32_t tgt = 0;
char *cs = NULL;
char *key = NULL;
int success = 0;

/* Step 1: Grab the name of the new key. */
src = adm_get_parm_obj(parms, 0, AMP_TYPE_STR);
dst = adm_get_parm_obj(parms, 1, AMP_TYPE_STR);
tgt = adm_get_parm_int(parms, 2, &success);
cs = adm_get_parm_obj(parms, 3, AMP_TYPE_STR);
key = adm_get_parm_obj(parms, 4, AMP_TYPE_STR);


if(get_bcb_prof_by_name(cs) != NULL)
{
        Object addr;
        Object elt;

        /* Step 3: Check to see if key exists. */
        sec_findKey(key, &addr, &elt);
        if(elt != 0)
        {
                /* Step 4: Update the BCB Rule. */
                if(sec_addBspBcbRule(src, dst, tgt, cs, key) == 1)
                {
                        *status = CTRL_SUCCESS;
                }
                else
                {
                        AMP_DEBUG_ERR("dtn_sbsp_ctrl_add_bcbrule", "Can't add rule.", NULL);
                }
        }
        else
        {
                AMP_DEBUG_ERR("dtn_sbsp_ctrl_add_bcbrule", "Key %s doesn't exist.", key);
        }
}
else
{
        AMP_DEBUG_ERR("dtn_sbsp_ctrl_add_bcbrule", "Ciphersuite %s not supported.", cs);
}

/*
 * +----------------------------------------------------------------------------+
 * |STOP CUSTOM FUNCTION ctrl_add_bcb_rule BODY
```

Data collection for BP Endpoint Active

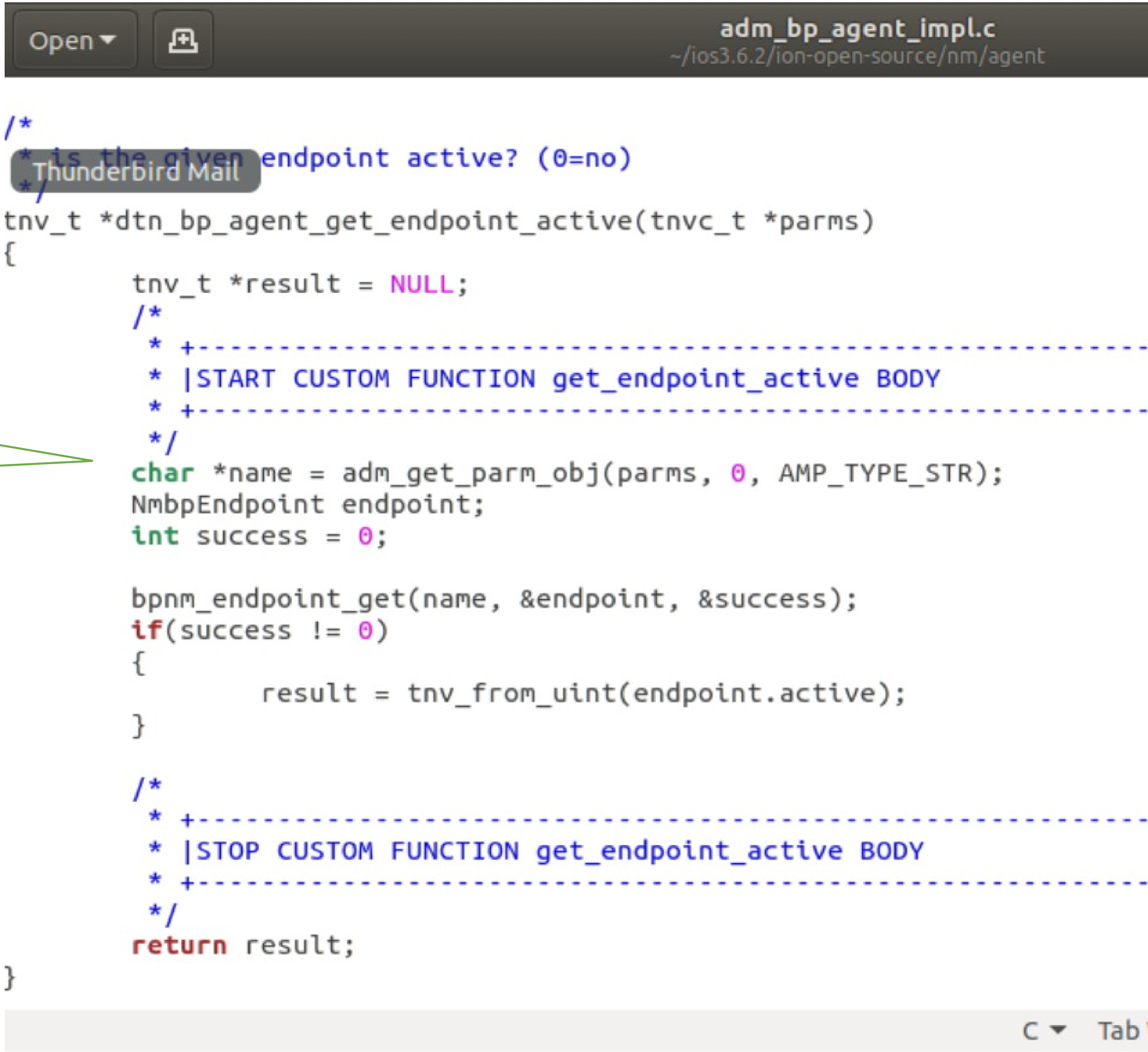
- Previous example of BP endpoint report.
- Report is fully auto-generated by CAMP.
- User only needs to specify how to collect the data contained in the report.



```
/*
 * is the given endpoint active? (0=no)
 */
tnv_t *dtn_bp_agent_get_endpoint_active(tnvc_t *parms)
{
        tnv_t *result = NULL;
        /*
         * +-----------------------------------------------------
         * |START CUSTOM FUNCTION get_endpoint_active BODY
         * +-----------------------------------------------------
         */
        /*
         * +-----------------------------------------------------
         * |STOP CUSTOM FUNCTION get_endpoint_active BODY
         * +-----------------------------------------------------
         */
        return result;
}
```

- Example of populated function to collect active state of given endpoint name.

Takes param and uses it for associative lookup of its active state.



```
/*
 * is the given endpoint active? (0=no)
 */
tnv_t *dtn_bp_agent_get_endpoint_active(tnvc_t *parms)
{
        tnv_t *result = NULL;
        /*
         * +-----------------------------------------------------------------
         * |START CUSTOM FUNCTION get_endpoint_active BODY
         * +-----------------------------------------------------------------
         */
        char *name = adm_get_parm_obj(parms, 0, AMP_TYPE_STR);
        NmbpEndpoint endpoint;
        int success = 0;

        bpnm_endpoint_get(name, &endpoint, &success);
        if(success != 0)
        {
                result = tnv_from_uint(endpoint.active);
        }

        /*
         * +-----------------------------------------------------------------
         * |STOP CUSTOM FUNCTION get_endpoint_active BODY
         * +-----------------------------------------------------------------
         */
        return result;
}
```