Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2015

M. Bagnulo
UC3M
T. Burbridge
BT
S. Crawford
SamKnows
J. Schoenwaelder
V. Bajpai
Jacobs University Bremen
July 3, 2014

Large MeAsurement Platform Protocol
draft-bagnulo-lmap-http-02

Abstract

This documents specifies the LMAP protocol based on HTTP for the Control and Report in Large Scale Measurement Platforms.

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 http://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 January 4, 2015.

Copyright Notice

Table of Contents

1. Introduction

A Large MeAsurement Platform (LMAP) is an infrastructure deployed in the Internet that enables performing measurements from a very large

number of vantage points.

The main components of a LMAP are the following:

- The Measurement Agents (MAs): these are the processes that perform the measurements. The measurements can be both active or passive measurements.
- The Controller: this is the element that controls the MAs. In particular it provides configuration information and it instructs the MA to perform a set of measurements.
- The Collector: this is the repository where the MAs send the results of the measurements that they have performed.

These and other terms used in this document are defined in [I-D.ietf-lmap-framework]. We only include the definition of the main elements in this document so it is self-contained and can be read without the need to consult other documents. The reader is referred to the terminology draft for further details.

In order for a LMAP to work, the following protocols are required:

- Measurement protocols: These are the protocols used between the MA and the Measurement Peer in active measurements. These are the actual packets being used for the measurement operations.
- Control Protocol. This is the protocol between the Controller and the MAs. This protocol is used to convey measurement Instruction(s) from the Controller to the MA as well as logging, failure and capabilities information from the MA to the Controller.
- Report Protocol. This is the protocol between the MAs and the Collector. This protocol conveys information about the results of the measurements performed by the MA to the Collector.

Both the Control protocol and the Report protocol have essentially two parts:a transport and a data model. The data model represents the information about measurement instructions and logging/failure/ capabilities (in the Control protocol) and the information about measurement results (in the Report protocol) that is being exchanged between the parties. The transport is the underlying protocol used to exchange that information. This document specifies the use of

HTTP 1.1 [RFC7230] [RFC7231] [RFC7232] [RFC7233] [RFC7234] [RFC7235] as a transport for the Control and the Report protocol. This document also defines the data model for the Control and Report protocols. The data model described in this document follows the information model described in [I-D.ietf-lmap-information-model]. The Measurement protocols are out of the scope for this document.

At this stage, the goal of this document is to explore different options that can be envisioned to use the HTTP protocol to exchange LMAP information and to foster discussion about which one to use (if any). Because of that, the document contains several discussion

paragraphs that explore different alternative approaches to perform the same function.

2. Overview

This section provides an overview of the architecture envisioned for a LMAP using HTTP as transport protocol. As we described in the previous section, a LMAP is formed by a large number of MAs, one or more Controllers and one or more Collectors. We assume that before the MAs are deployed, it is possible to pre-configure some information in them. Typically this includes information about the MA itself (like its identifier), security information (like some certificates) and information about the Controller(s) available in the measurement platform. Once that the MA is deployed it will retrieve additional configuration information from the pre configured Controller. After obtaining the configuration information, the MA is ready to receive Instructions from the Controller and initiate measurement tasks. The MA will perform the following operations:

- It will obtain Instructions from one of the configured Controllers. These Instructions include information about the set of measurement tasks to be performed, a schedule for the execution of the measurements as well as a set of report channels. This information is downloaded by the MA from the Controller. The MA will periodically check whether there are new Instructions available from the Controller. This document specifies how the MA uses the HTTP protocol to retrieve information from the Controller.
- The MA will execute measurement tasks either by passively

listening to traffic or by actively sending and receiving measurement packets. How this is done is out of the scope of this document.

- After one or more measurements have been performed, the MA reports the results to the Collector. The timing of these uploads is specified in the measurement Instruction i.e. each measurement specified in a measurement Instruction contains a report information, defining when the MA should report the results back to the Collector. This document specifies how the MA uses the HTTP protocol to upload the measurement results to the Collector.
- In addition, the MA will periodically report back to the Controller information about its capabilities (like the number of interfaces it has, the corresponding IP addresses, the set of measurement methods it supports, etc) and also logging information (whether some of the requested measurement tasks failed and related information).

3. Naming Considerations

In this section we define how the different elements of the LMAP architecture are identified and named.

The Controller and the Collectors can be assumed to have both an IP address and a Fully Qualified Domain Name (FQDN). It is natural to use these as identifiers for these elements. In this document we will use FQDNs, but IP addresses can be used as well.

The MAs on the other hand, are likely to be executed in devices located in the end user premises and are likely to be located behind a NAT box. It is reasonable to assume they have neither a public IP address nor a FQDN. We propose then that the MAs are identified using an Universally Unique IDentifier URN as defined in RFC 4122 [RFC4122]. In particular each MA has a version 4 UUID, which is randomly or pseudo randomly generated.

DISCUSSION:

MA ID Configuration: Some open issues related to this are: a) whether the MA ID is configured before of after the MA is

deployed, b) if configured after deployment whether the MA ID is generated locally and posted or fetched from the Controller and c) whether this is within the scope of this (or other) specification if any. These issues seem also to be related to the nature of the MA platform (wether the MA is a software downloaded into a general purpose device or it is a special purpose hardware box). Consider the case that the MA is located in a special purpose hardware box, then having the MA ID pre configure before deployment requires a per device customization that is expensive. It would be more costly efficient to reuse an existent (hopefully) unique identifier available in the hardware (such as a MAC address) to serve as a one-time pre configured identifier to be used to fetch (or post a self generated) the MA ID from the Controller once the MA is deployed. The requirement for such one-time identifier is that they must be unique (which is not always true for the MACs). About the local generation of the MA ID (as opposed to fetch it from the Controller), the generation process performed in the MA MUST be idempotent, i.e. if the MA was factory-reset then the server would still see it with the same MA ID when it came back up. This is probably easier to achieve if it is generated in the Controller and then fetched by the MA. Finally, it is not clear at this stage if this needs to be specified in this document or in the information model document or left open to the implementers. Group identifiers. In some cases, like the case of measurements in mobile devices, it may be important because of privacy considerations for the MA not to have a unique identifier. It is

possible then to assign "Group identifiers" to a set of devices that share relevant characteristics from the measurement perspective (e.g. devices from the same operator, with the same type of contract or other relevant feature). In this case, the MAs within the same group would retrieve common measurement Instructions from the controller by presenting the same Group ID and would report results including the Group ID in the report. This would imply that it would not be possible for the platform to correlate specific measurement data with any given MA. The downside of this is that some MAs may be over-represented while other under-represented in the measurement data and it would not be possible to detect this case (for instance a given MA may have reported 20 results while another one only one). In order to deal with this issue, the MA behaviour must be programmed accordingly (e.g. the MA should not perform more than one measurement every

given period of time).In addition, it should be noted that privacy is only achieved in a holistic way. This means that really anonymity of the MA is incompatible with strong authentication. In particular, if a measurement platform's goal is to keep MAs anonymous, it cannot require any form of strong authentication (other than weak group authentication e.g. a password shared by a group), which has security implications. In particular, the threat for report forgery (i.e. enabling an attacker to submit forged reports as discussed in the security considerations) increases.

There are additional naming considerations related to:

- The measurements. In order to enable a Controller to properly convey a measurement schedule, it must be possible for the Controller to specify a measurement to be performed while providing the needed input parameters. While this is critical, it is out of the scope of this document. There is a proposed registry for metrics/measurements in [I-D.bagnulo-ippm-new-registry-independent])
- The resources being exchanged, namely, the configuration information, the measurement Instructions and the reports. These are being discussed in the upcoming sections.

4. Information model

The information model for LMAP is described [I-D.ietf-lmap-information-model]. It contains basically two models one for the control information (i.e. the Instructions from the Controller to the MA) and a model for the Report information. We briefly describe their overall structure here.

The control information (or Instruction) has the following five elements:

- The Set of Measurement Task Configurations: This element defines the measurements/test that the MA will perform without defining the schedule when they will be performed.
- The Set of Report Channels: This element defines the set of collectors as well as the reporting schedules for the reports.

- The Set of Measurement Schedules for Repeated Tasks: defines the schedules for the repeated measurements, by referencing the measurement tasks defined in the second element.
- Suppression information

Summary of Report information model here.

Summary of Capability and Status information model here.

Summary of Logging information model here.

5. Transport protocol

5.1. Pre-configured information

As we mentioned earlier, the MAs contain pre-configured information before being deployed. The pre-configured information is the following:

- The UUID for the MA. This should be pre-configured so that the Controller is aware of the MA and can feed configuration information and measurement Instructions to it.
- Information about one or more Controllers. The MA MUST have enough information to create the URL for the Instruction resources. This includes the the FQDN of each of the Controller or the IP addresses of the Controller, as well as the well-known path prefix and its identifier.
- The certificate for the Certification authority that is used in the platform to generate the certificates for the Controller and the Collector. See the Security considerations section below.
- The security related information for the MA (it can be a certificate for the MA and the corresponding private key, or simply a key/password depending on the security method used, see the security considerations section below).

5.2. Control Protocol

The Control protocol is used by the MA to retrieve Instruction information from the Controller. In this section we describe how to use HTTP to transport Instructions. The Instruction information is

structured as defined in the LMAP Information model

[I-D.ietf-lmap-information-model] as described in the previous section. The MA uses the Control protocol to retrieve all the resources described above, namely, the Agent information, the Set of Measurement Task Configurations, the Set of Report Channels, the Set of Measurement Schedules for Repeated Tasks and the Set of Measurement Schedules for Isolated Tasks. The main difference from the HTTP perspective is that the MA MUST have the URL for the Agent Information resource pre-configured as described in the previous section, while the URLs for all the other resources are contained in the Agent Information resource itself.

5.2.1. Retrieving Instructions

In order to retrieve the Instruction resources from the Controller the MA can use either the GET or the POST method using the corresponding URL.

5.2.1.1. Using the GET method

One way of using the GET method to retrieve configuration information is to explicitly name the configuration information resources and then apply the GET method. The MA retrieves its Instruction when it is first connected to the network and periodically after that. The frequency for the periodical retrieval is contained in the Agent Information (???).

The URL for the Agent Information resource is formed as the FQDN of the Controller, a well-known path prefix and the MA UUID. The well-known path prefix is /.well-known/lmap/ma-info. The URL for the remaining resources that compose the Instruction are contained in the Agent Information.

Agent Information retrieval: In order to retrieve the Agent information the MA uses the HTTP GET method follows:

```
GET /.well-known/lmap/ma-info/ < ma-iid> HTTP/1.1
Host: FQDN or IP of the Controller
Accept: application/json (as per [RFC7159])
```

The Agent Information should contain the Configuration Retrieval Schedule (i.e. how often the MA should retrieve configuration information) and also the Measurement Instruction Retrieval Schedule (i.e. how often the MA should retrieve the Measurement Instruction from the Controller). COMMENT: this is missing from the Data Model

The retrieval of the remaining resources of the Instruction using the GET method is analogous, only that the URL is extracted from the

Agent Information file rather than constructed with pre-configured information.

The format for the response should be described here

Periodical Instruction retrieval: After having downloaded the initial Instruction information, the MA will periodically look for updated Instruction information. The frequency with which the MA polls for the new Instructions from the Controller is contained in the last Agent Information downloaded. In order to retrieve the Agent Information, the MA uses the GET method as follows:

```
GET /.well-known/lmap/ma-info/ma-iid/ HTTP/1.1
Host: FQDN or IP of the Controller
Accept: application/json (as per [RFC7159])
If-None-Match: the eTag of the last retrieved Agent Information
(an alternative option here is to use If-Modified-Since, not sure
which one is best)
```

For the other Instruction resources, the GET method is applied in the same way just that the URL used are the ones retrieved in the last Agent Information.

The format for the response should be described here

Alternatively, instead of explicitly naming the Instruction resources for each MA, it is possible to perform a query using the GET method as well. In this case, the MA could perform a GET for the following URI http://controller.example.org/?ma=maid & q=ma-info (similar queries can be constructed for the other Instruction resources). (I am not sure how to express in this case the condition that the MA wishes to retrieve the configuration if it is newer than the last one it downloaded.)

5.2.1.2. Using the POST method

An alternative to retrieve Instruction resources is to use the POST method to perform a query (similar to the query using GET). In this case there is no explicit naming of the Instruction information of each MA, but a general Instruction resource and the POST method is used to convey a query for the Instruction information of a particular MA. For the case of the Agent Information resource, this would look like as follows:

```
POST /.well-known/lmap/ma-info/ma-iid/ HTTP/1.1
Host: controller.example.com
```

```
Content-Type: application/lmap-maid+json
Accept: application/lmap-config+json
```



```
{
"ma-id" : "550e8400-e29b-11d4-a716-446655440000",
}
```

The reply for this query would contain the actual configuration information as follows:

```
HTTP/1.1 200 OK
Content-Length: xxx
Content-Type: application/lmap-config+json
{
// whatever config goes here
}
```

In this case, the URLs contained in the Agent information can be generic and not MA specific, since the MA will use the POST method including its own identifier when retrieving the Instruction resources.

The argument for this approach is that this is much more extensible since the POST can carry complex information and there is no need to "press" arguments into the strict hierarchy of URIs.

We need to describe how to use this to retrieve newer information in the periodic case.

5.2.2. Handling communication failures

The cases that the MA is unable to retrieve the Instructions are handled as follows:

- The MA will use a timeout for the communication of TIMEOUT seconds. The value of TIMEOUT MUST be configurable via the aforementioned Configuration Information retrieval protocol. The default value for the TIMEOUT is 3 seconds. If after the timeout, the communication with the Controller has not been established, the MA will retry doing an exponential backoff and doing a round robin between the different Controllers it has available.
- If a HTTP error message (5xx) is received from the Controller as a

response to the GET request, the MA will retry doing an exponential back-off and doing a round robin between the different Controllers it has available. The 5xx error codes indicate that this Controller is currently incapable of performing the requested operation.

5.2.3. Pushing Information from the Controller to the MA

The previous sections described how the MA periodically polls the Controller to retrieve Instruction information. The frequency of the downloads is configurable. The question is whether this is enough or a mechanism for pushing Instruction information is needed. Such method would enable to contact the MA in any moment and take actions like triggering a measurement right away or for instance to stop an ongoing measurement (e.g. because it is disturbing the network). The need for such a mechanism is likely to depend on the use case of the platform. Probably the ISP use case is more likely to require this feature than the regulator/benchmarking use case. It is probably useful then to provide this as an optional feature.

The main challenge in order to provide this feature is that the MAs are likely to be placed behind NATs, so it is not possible for the Controller to initiate a communication with the MA unless there is a binding in the NAT to forward the packets to the MA. There are several options that can be considered to enable this communication:

- The MA can use one of the NAT control protocols, such as PCP or UPNP. If this approach is used, the MA will create a binding in the NAT opening a hole. After that, the MA should inform the Controller about which is the IP address and port available for communication. It would be possible to re-use existing protocols to forward this information. The problem with this is that the NAT may not support these protocols or they may not be activated. In any case, a solution should try to use them in the case they are available.
- If it is not possible to use a NAT control protocol, then the MA can open a hole in the NAT by establishing a connection to the Controller and keeping it open. This allows the Controller to

push information to the MA through that connection. One concern with this approach is that the MA is playing the role of the client and the Controller is playing the role of the server (the MA is initiating the TCP connection), but it would be the Controller who would use the PUT method towards the MA reversing the roles. An alternative approach is that the MA has a long running GET pending which is answered by the server if the measurement Instruction changes (or the server times out, in which case the MA restarts the long running GET. More discussion is needed about whether one of these options is acceptable or not. In addition, this would imply that the Controller should maintain as many open sessions as MAs it is managing, which imposes additional burden in the Controller. There are security considerations as well, but these are covered in the Security Considerations section below.

5.3. Report protocol

The MA after performing the measurements reports the results to a collector. There can be more than one collector within a LMAP framework. Each collector is identified by its FQDN or IP address which is retrieved as part of the Agent information from a pre-configured controller as previously discussed. The number of Collectors that the MA uploads the results to as well as the schedule when it does so is defined in the measurement Instruction previously downloaded from the Controller. The MA themselves are identified by a UUID.

There are two options that can be considered for the MA to upload reports to the Collector either to use the PUT method or to use the POST method.

If the PUT method option is used, then the MA need to perform the PUT method using an explicit name for the report resource it is transferring to the Collector. The name of the resource is contained in the Agent Information previously retrieved by the MA

The other option is for the MA to use the POST method to upload the measurement reports to one or more Collectors. In this case,, the POST message body can contain the identifier of the MA and additional information describing the report in addition to the report itself.

One argument to consider is that PUT is idempotent. This means that if the network is bad at some point and the MA is not sure whether its request made it through, it can send it a second (or nth) time, and it is guaranteed that the request will have exactly the same effect as sending it for the first time. POST does not by itself guarantee this. This can be achieved by verifying the report data itself, and contrast it with data already stored int he Collector database.

5.3.1. Handling communication failures

The MA will use a timeout for the communication with the Collector of TIMEOUT seconds. The value of TIMEOUT MUST be configurable via the aforementioned Configuration Information retrieval protocol. The default value for the TIMEOUT is 3 seconds.

If the MA is uploading the report to several Collectors and it manages to establish the communication before TIMEOUT seconds with at least one of them, but not with one or more of the other Collectors, then the MA gives up after TIMEOUT seconds and it MAY issue an alarm. The definition of how to do that operation is out of the scope of this document.

If the MA is uploading the report to only one Collector, and it does not manages to establish a communication before TIMEOUT seconds, then it retry doing an exponential backoff and doing a round robin between the different Collectors it has available.

Similarly, if an HTTP error message (5xx) is received from the Collector as a response to the PUT request, the MA will retry doing an exponential backoff and doing a round robin between the different Collectors it has available. The 5xx error codes indicate that this Collector is currently incapable of performing the requested operation.

In order to support this, the information model must express the difference between a report sent to multiple collectors and multiple collectors used for fallback.

6. LMAP Data Model

This section will contain the data model in json.

6.1. Timing Information

An example immediate timing object with no defined randomness is shown below:

```
-- ma_timing_obj
{
    "id": 1
  , "ma_timing_option": "IMMEDIATE"
  , "ma_randomness_option": null
  , "ma_timing_name": null
}
```

6.2. Channels

An example channel object using the aforementioned timing object is shown below:


```
-- ma_channel_obj
{
    "id": 1
  , "ma_channel_timing_obj_id": 1
  , "ma_channel_connect_always": "true"
  , "ma_channel_target": "controller.example.org"
  , "ma_channel_certificate": "MIIFEzCCAvsCAQEwDQYJ"
  , "ma_channel_interface_name": "eth0"
  , "ma_channel_name": "INSTRUCTION"
}
```

```
-- ma_channel_obj
{
    "id": 2
  , "ma_channel_timing_obj_id": 1
  , "ma_channel_connect_always": "true"
  , "ma_channel_target": "controller.example.org"
  , "ma_channel_certificate": "VtAKQhFM89kOIxn5g..."
  , "ma_channel_interface_name": "eth0"
  , "ma_channel_name": "MA-TO-CONTROLLER"
}

-- ma_channel_obj
{
    "id": 3
  , "ma_channel_timing_obj_id": 1
  , "ma_channel_connect_always": "true"
  , "ma_channel_target": "collector.example.org"
  , "ma_channel_certificate": "X1Ow9+Grmkb9EmVPfqH0..."
  , "ma_channel_interface_name": "eth0"
  , "ma_channel_name": "REPORT"
}
```

6.3. Pre-Configuration

An example pre-config object using the aforementioned channel objects is shown below:

```
-- ma_preconfig_obj
{
    "id": 1
  , "ma_instruction_channel_obj_id": 1
  , "ma_ma_to_controller_channel_obj_id": 2
  , "ma_device_id": "01:23:45:67:89:ab"
  , "ma_agent_id": null
}
```


6.4. Configuration

An example config object using the aforementioned channel objects is shown below:

```
-- ma_config_obj
{
    "id": 1
  , "ma_agent_id": "c54c284a01ee11e48dd310ddb1bd23b5"
  , "ma_group_id": "d7d63d7a01ee11e49b2210ddb1bd23b5"
  , "ma_instruction_channel_obj_id": 1
  , "ma_report_ma_id_flag": "false"
  , "ma_instruction_channel_failure_threshold": 10
}
```

6.5. Instruction

An example instruction object is shown below:

```
-- ma_instruction_obj
{
    "id": 1
  , "ma_supression_obj_id": 1
}
```

6.5.1. Measurement Supression

An example supression object used by the aforementioned instruction object is shown below:

```
-- ma_supression_obj
{
    "id": 1
  , "ma_supression_enabled": "true"
  , "ma_supression_start": 1404309159
  , "ma_supression_end": 1404309193
}
```

6.5.2. Measurement Task Configurations

An example task object used by the aforementioned instruction object is shown below:

```
-- ma_task_obj
{
    "id": 1
  , "instruction_obj_id": 1
  , "supression_obj_id": 1
  , "ma_task_name": "UDP latency"
  , "ma_task_registry": "urn:ietf:ippm..."
  , "ma_task_options": "..." # omitted for brevity reasons
  , "ma_task_cycle_id": "1"
}
```

6.5.3. Measurement Schedules

An example schedule object used by the aforementioned instruction object is shown below:

```
-- ma_schedule_obj
{
    "id": 1
  , "instruction_obj_id": 1
  , "supression_obj_id": 1
  , "timing_obj_id": 1
  , "ma_schedule_name": "A schedule with immediate timing"
}

-- ma_sched_task_obj
{
    "id": 1
  , "schedule_obj_id": 1
  , "ma_schedule_task_name": "A schedule for UDP latency task"
}
```

6.5.4. Report Channels

An example schedule report object used by the aforementioned instruction object is shown below:

```
-- ma_sched_report_obj
{
    "id": 1
  , "ma_schedule_obj_id": 1
  , "channel_obj_id": 3
  , "ma_schedule_task_report_channel_name": "A report channel"
  , "ma_schedule_task_filter": null
}
```

6.6. MA to Controller

An example log object is shown below:

```
-- ma_log_obj
{
    "id": 1
  , "ma_log_agent_id": "0e49b32b01fa11e4bcaf10ddb1bd23b5"
  , "ma_log_event_time": 1404313752
  , "ma_log_code": "200"
  , "ma_log_description": "OK"
}
```

6.7. Capability and Status

An example status object is shown below:

```
-- ma_status_obj
{
    "id": 1
  , "ma_agent_id": "c54c284a01ee11e48dd310ddb1bd23b5"
  , "ma_device_id": "01:23:45:67:89:ab"
  , "ma_hardware": "TL-MR3020"
  , "ma_software": "Busybox"
  , "ma_firmware": "4560"
  , "ma_last_measurement": 1404315031
  , "ma_last_report": 1404315053
  , "ma_last_instruction": 140431312
  , "ma_last_configuration": 140423245
}
```

An example capability object is shown below:

```
-- ma_capability_obj
{
    "id": "1"
  , "ma_status_obj_id": 1
  , "ma_measurement_id": "c56cb44a028c11e495d910ddb1bd23b5"
  , "ma_measurement_version": "v1.0"
}
```

An example interface object is shown below:


```
-- ma_interface_obj
{
    "id": 1
  , "ma_status_obj_id": 1
  , "ma_interface_name": "eth0"
  , "ma_interface_type": "100baseTX"
  , "ma_interface_speed": "100Mbps"
  , "ma_link_layer_address": "01:23:45:67:89:ab"
}
```

An example ip address object used by the aforementioned interface object is shown below:

```
-- ip_address
{
    "id": 1
  , "interface_obj_id": 1
  , "value": "192.168.1.10"
  , "ma_interface_if_ip_address": 1
  , "ma_interface_if_dns_server": 0
  , "ma_interface_if_gateway": 0
}
```

6.8. Reporting

An example report object is shown below:

```
-- ma_report_obj
{
    "id": 1
  , "ma_report_agent_id": "c54c284a01ee11e48dd310ddb1bd23b5"
  , "ma_report_group_id": "d7d63d7a01ee11e49b2210ddb1bd23b5"
  , "ma_report_date": 1404316528
```

```
}

-- ma_report_task_obj
{
    "id": 1
  , "ma_task_obj_id": 1
  , "ma_report_obj_id": 1
  , "ma_report_task_column_headers": "...,...,..."
}
```



```
-- ma_result_row_obj
{
    "id": 1
  , "ma_report_task_obj_id": 1
  , "ma_report_result_time": 1404317298
  , "ma_report_result_cross_traffic": null
  , "ma_report_result_values": "...,...,"
}
```

7. Security considerations

Large Measurement Platforms may result in a security hazard if they are not properly secured. This is so because they encompass a large number of MAs that can be managed and coordinated easily to generate traffic and they can potentially be used for generating DDoS attacks or other forms of security threats.

From the perspective of the protocols described in this documents, we can identify the following threats:

- Hijacking: Probably the worst threat is that an attacker takes over the control of one or more MAs. In this case the attacker would be able to instruct the MAs to generate traffic or to eavesdrop traffic in their location. It is then critical that the MA is able to strongly authenticate the Controller. An alternative way to achieve this attack is to alter the

communication between the Controller and the MAs. In order to prevent this form of attack, integrity protection of the communication between the Controller and the MAs is required.

- Polluting: Another type of attack is that an attacker is able to pollute the Collectors database by providing false results. In this case, the attacker would attempt to impersonate one or more MAs and upload fake results in the Collector. In order to prevent this, the authentication of the MAs with the Collector is needed. An alternative way to achieve this is for an attacker to alter the communication between the MA and the Collector. In order to prevent this form of attack, integrity protection of the communication between the MA and the Collector is needed.
- Disclosure: Another threat is that an attacker may gather information about the MAs and their configuration and the Measurement schedules. In order to do that, it would connect to the Controller and download the information about one or more MAs. This can be prevented by using MA authentication with the Controller. An alternative mean to achieve this would be for the attacker to eavesdrop the communication between the MA and the Controller. In order to prevent this, confidentiality in the communication between the MA and the Controller is required. Similarly, an attacker may wish to obtain measurement result

information by eavesdropping the communication between the MA and the Collector. In order to prevent this, confidentiality in the communication between the MA and the Collector is needed.

In order to address all the identified threats, the HTTPS protocol must be used for LMAP (i.e. using HTTP over TLS). HTTPS provides confidentiality, integrity protection and authentication, satisfying all the aforementioned needs. Ideally, mutual authentication should be used. In any case, server side authentication MUST be used. In order to achieve that, both the Controller and the Collector MUST have certificates. The certificate of the CA used to issue the certificates for the Controller and the Collector MUST be pre configured in the MAs, so they can properly authenticate them. As mentioned earlier, ideally, mutual authentication should be used. However, this implies that certificates for the MAs are needed. Certificate management for a large number of MAs may be expensive and cumbersome. Moreover, the major threats identified are the ones related to hijacking of the MAs, which are prevented by authenticating the Controller. MAs authentication is needed to

prevent Polluting and Disclosure threats, which are less severe. So, in this case, alternative (cheaper) methods for authenticating MAs can be considered. The simplest method would be to simply use the MA UUID as a token to retrieve information. Since the MA UUID is 128 bit long, it is hard to guess. It would be also possible to use a password and use the HTTP method for authentication. It is not obvious that managing passwords for a large number of MAs is easier than managing certificates though.

An additional security consideration is posed by the mechanism to push information from the Controller to the MAs. If this method is used, it would be possible its abuse by an attacker to control the MAs. This threat is prevented by the use of HTTPS. If HTTPS is used in the established connection between the MA and the Controller, the only effect that a packet generated by an external attacker to the MA or the Controller would be to reset the HTTPS connection, requiring the connection to be re-established.

It is required in this document that both the Controller and that the Collector are authenticated using digital certificates. The current specification allows for the MA to have information about the certificate of the Certification authority used for generating the Controller and Collector certificates while the actual certificates are exchanged in band using TLS. Another (more secure) option is to perform certificate pinning i.e. to configure in the MAs the actual certificates rather than the certification authority certificate. Another measure to increase the security would be to limit the domains that the FQDNs of the Controller and/or the Collector (e.g. only names in the exmaple.org domain).

Large scale measurements can have privacy implications, especially in some scenarios like mobile devices performing measurements. In this memo we have considered using Group IDs to the MA in order to avoid the possibility for the platform to track each individual MA that is feeding results.

8. IANA Considerations

Registration of the well-known URL

9. Acknowledgments

10. References

10.1. Normative References

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.

[RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.

[RFC7230] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014.

[RFC7231] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, June 2014.

[RFC7232] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", RFC 7232, June 2014.

[RFC7233] Fielding, R., Lafon, Y., and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Range Requests", RFC 7233, June 2014.

[RFC7234] Fielding, R., Nottingham, M., and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, June 2014.

[RFC7235] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Authentication", RFC 7235, June 2014.

[I-D.ietf-lmap-information-model] Burbridge, T., Eardley, P., Bagnulo, M., and J. Schoenwaelder, "Information Model for Large-Scale Measurement Platforms (LMAP)", draft-ietf-lmap-information-model-01 (work in progress), June 2014.

10.2. Informative References

[I-D.bagnulo-ippm-new-registry-independent]
Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and A. Morton, "A registry for commonly used metrics. Independent registries", draft-bagnulo-ippm-new-registry-independent-01 (work in progress), July 2013.

[I-D.ietf-lmap-framework]
Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A framework for large-scale measurement platforms (LMAP)", draft-ietf-lmap-framework-07 (work in progress), June 2014.

Authors' Addresses

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6249500
Email: marcelo@it.uc3m.es
URI: http://www.it.uc3m.es

Trevor Burbridge
British Telecom
Adastral Park, Martlesham Heath
IPswitch
ENGLAND

Email: trevor.burbridge@bt.com

Sam Crawford
SamKnows

Email: sam@samknows.com

Juergen Schoenwaelder
Jacobs University Bremen
Campus Ring 1
28759 Bremen
Germany

Email: j.schoenwaelder@jacobs-university.de

Vaibhav Bajpai
Jacobs University Bremen
Campus Ring 1
28759 Bremen
Germany

Email: v.bajpai@jacobs-university.de