Network Working Group B. Carpenter
Internet-Draft Univ. of Auckland
Intended status: Standards Track B. Liu, Ed.
Expires: October 6, 2016 Huawei Technologies
W. Wang
X. Gong
BUPT University
April 04, 2016

Generic Autonomic Signaling Protocol Application Program Interface (GRASP API)
draft-liu-anima-grasp-api-00

Abstract

This document specifies the application program interface of the Generic Autonomic Signaling Protocol (GRASP). The API is used for Autonomic Service Agent (ASA) calling the GRASP protocol module to communicate the autonomic network signalings with other ASAs.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

As defined in [I-D.ietf-anima-reference-model] , ASA is the atomic entities of an autonomic function; and it is instantiated on autonomic nodes. When ASAs communicate with each other, they should use the GRASP [I-D.ietf-anima-grasp].

As the following figure shows, the GRASP could contain two sub-layers. The bottom is the GRASP base protocol module, which is only responsible for sending and recieving GRASP messages. The upper layer is some extended funcitons based upon GRASP basic protocol. For example, [I-D.liu-anima-grasp-distribution] is one of the extended functions.

```
             +-----+                             +-----+
             |ASAs|                              |ASAs|
             +-----+                             +-----+
                |                                   |
                | GRASP Function API                |
                |                                   |
             +------------------+                   |GRASP API
             | GRASP Extended   |                   |
             | Function Modules |                   |
             +------------------+                   |
             +------------------------------------------+
             |              GRASP Module                |
             +------------------------------------------+
```

Both the GRASP base module and the extended function modules should be available to the ASAs. Thus, there needs to be two sub-set of API. However, since the extended functions could be added in a incremental manner, it is not very proper to define the function APIs in a single document. This document only defines the base GRASP API.

2. GRASP API for ASA

2.1. Design Principles

The assumption of this document is that any Autonomic Service Agent (ASA) needs to call a GRASP module that handles protocol details (security, sending and listening for GRASP messages, waiting, caching discovery results, negotiation looping, sending and receiving sychronization data, etc.) but understands nothing about individual objectives. So this is a high level abstract API for use by ASAs.

This is very preliminary. Two particular gaps exist:

- Bootstrapping issues are hidden behind grasp_init.

- Rapid mode is not supported.

2.2. API definition

- grasp_init(objectives asa_negotiate_result [asa_geq_fn]) -> OK/ fail

  - This initialises state in the GRASP module for the calling entity (the ASA). The objectives parameter is a list of all GRASP objective options that the ASA supports (see below for ABNF and more details). It also provides references to one or two functions within the ASA that GRASP may call asynchronously.

  - OK: means that the ASA has been authenticated and has credentials to proceed.

  - fail: means that the ASA has not been authenticated and cannot operate.

- grasp_discover(objective) -> OK/fail

  - This causes GRASP discovery for the listed objective.

  - OK: means that discovery has succeeded and the locator has been cached.

  - fail: means that discovery has failed. The ASA must wait before retrying.

- o grasp_negotiate(objective) -> objective/fail

  - This causes GRASP to start negotiation for the given objective, and to negotiate within the limits set (see below).

  - objective: Negotiation succeeded, this contains the negotiated value.

  - fail: Negotiation failed. The ASA must wait before retrying.

- o grasp_synchronize(objective) -> objective/fail

  - This sends a GRASP synchronization request for the given objective.

  - objective: Synchronization succeeded, this contains the received value.

  - fail: Synchronization failed. The ASA must wait before retrying.

- o asa_geq_fn(objective_number value value) -> true/false

  - This function is optionally provided by the ASA for use by the GRASP module. It returns true iff the first value is considered greater than or equal to the second value. The semantics of this comparison are known only to the ASA; the GRASP module uses this function to conduct negotiation to find an acceptable value between the set limits. If no function is provided, GRASP uses normal arithmetic comparison of the values, considered as unsigned binary numbers.

- o grasp_listen_negotiate(objective) -> OK/fail

  - This causes GRASP to listen for negotiation requests for the given objective, and to negotiate within the limits set (see below). This call may be repeated whenever the limits change.

  - OK: GRASP is listening.

  - fail: GRASP is not listening.

- o asa_negotiate_result(objective) -> objective/fail

  - This is a function in the ASA called asynchronously by GRASP, if the ASA has previously called grasp_listen_negotiate and an incoming negotiation has terminated.

  - objective: Negotiation succeeded, this contains the negotiated value.

  - fail: Negotiation failed.

- grasp_stop_negotiate(objective) -> OK/fail

  - This causes GRASP to stop listening for negotiation requests for the given objective.

  - OK: GRASP is not listening.

  - fail: unspecified failure.

- grasp_listen_synchronize(objective) -> OK/fail

  - This causes GRASP to listen for synchronization requests for the given objective, and to reply with the given value. This call may be repeated whenever the value changes.

  - OK: GRASP is listening.

  - fail: GRASP is not listening.

- grasp_stop_synchronize(objective) -> OK/fail

  - This causes GRASP to stop listening for synchronization requests for the given objective.

  - OK: GRASP is not listening.

  - fail: unspecified failure.

3. Security Considerations

TBD.

4. IANA Considerations

This does not need IANA assignment.

5. Acknowledgements

This document was produced using the xml2rfc tool [RFC2629].

6. References

6.1. Normative References

[I-D.ietf-anima-grasp]
Bormann, C., Carpenter, B., and B. Liu, "A Generic Autonomic Signaling Protocol (GRASP)", draft-ietf-anima-grasp-04 (work in progress), March 2016.

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

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <http://www.rfc-editor.org/info/rfc2629>.

6.2. Informative References

[I-D.ietf-anima-reference-model]
Behringer, M., Carpenter, B., Eckert, T., Ciavaglia, L., Liu, B., Jeff, J., and J. Strassner, "A Reference Model for Autonomic Networking", draft-ietf-anima-reference-model-00 (work in progress), January 2016.

[I-D.liu-anima-grasp-distribution]
Liu, B. and S. Jiang, "Information Distribution over GRASP", draft-liu-anima-grasp-distribution-00 (work in progress), October 2015.

Authors' Addresses

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Bing Liu (editor)
Huawei Technologies
Q14, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing 100095
P.R. China

Email: leo.liubing@huawei.com

Wendong Wang
BUPT University
Beijing University of Posts & Telecom.
No.10 Xitucheng Road
Hai-Dian District, Beijing 100876
P.R. China

Email: wdwang@bupt.edu.cn

Xiangyang Gong
BUPT University
Beijing University of Posts & Telecom.
No.10 Xitucheng Road
Hai-Dian District, Beijing 100876
P.R. China

Email: xygong@bupt.edu.cn