Technical Objectives for the Autonomic Network Infrastructure
draft-carpenter-anima-ani-objectives-00

Abstract

This document defines several technical objectives for the Generic Autonomic Signaling Protocol (GRASP) for use by components of the Autonomic Networking Infrastructure outlined in the ANIMA reference model.

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 May 22, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

This document defines several technical objectives for use with the Generic Autonomic Signaling Protocol (GRASP) [I-D.ietf-anima-grasp]. They are intended for use by corresponding Autonomic Service Agents (ASAs) that realise infrastructure components of the Autonomic Network Infrastructure (ANI) outlined in the ANIMA reference model [I-D.ietf-anima-reference-model]. Also other early use cases are in scope.

Note: This draft is posted to allow systematic discussion of the various objectives in a consistent way. It is quite probable that rather than this being published as an RFC, the various objective definitions will be incorporated directly in the relevant specifications.

The reference model identifies several infrastructure components that will fit together to form the ANI, and early use cases for ANIMA are also considered:

Secure Bootstrap.

Autonomic Control Plane (ACP).

Stable Connectivity of Network OAM.
Intent Distribution.

Prefix Management

The following sections define GRASP objectives for each of these cases. They are described in informal object notation and formally using CBOR data definition language (CDDL) [I-D.greevenbosch-appsawg-cbor-cddl]. Undefined CDDL terms are defined in [I-D.ietf-anima-grasp].

2. Objectives for Secure Bootstrap

Three components are involved in the Bootstrapping Remote Secure Key Infrastructures (BRSKI) process described in [I-D.ietf-anima-bootstrapping-keyinfra]: the Registrar, the Join Assistant (or Proxy), and the Joining Node (or Pledge). The proxy and the pledge are attached to the same link-layer and use link-local communications only, to minimize exposure to eavesdroppers and malicious nodes. There may be multiple hops between the proxy and the registrar. Therefore, two different GRASP objectives are required: one that is used over an existing secure network between the registrar and the proxy, and another that is used over an insecure link-local hop between the proxy and the pledge. The security aspects and the corresponding limited instances of GRASP are discussed in [I-D.ietf-anima-bootstrapping-keyinfra] and [I-D.ietf-anima-grasp].

Note that since secure bootstrap takes place, by definition, on an incompletely secure network, the use of any protocol needs to be kept as simple and limited as possible. Therefore, only one GRASP message type is used - flooding - to avoid giving away any unnecessary information by any node involved.

A registrar announces itself to potential proxies by use of the "AN_registrar" objective. This is a synchronization objective primarily intended to be flooded throughout the network using the GRASP Flood Synchronization (M_FLOOD) message. In accordance with the design of the Flood message, a locator consisting of a specific IP address, IP protocol number and port number will be distributed with the flooded objective. An example of the objective is informally:

["AN_registrar", F_SYNCH, loop_count, [7, "BRSKI-TLS"]]

The formal CDDL definition is

registrar-objective = ["AN_registrar", F_SYNCH, loop-count, [radius, method]]
radius = uint ; loop-count at the source node
method = text ; name of the BRSKI method supported

The ‘radius’ parameter allows a proxy that receives this message to determine its distance in hops from the registrar, by subtracting the received ‘loop-count’ from ‘radius’.

The ‘method’ parameter indicates the specific BRSKI method available at the given locator. The initial possible values are "BRSKI-TLS" and "BRSKI-COAP". A registrar that supports more than one method will flood multiple versions of the "AN_registrar" objective.

2.1. Flooding Alternative for Proxy

A proxy announces itself to potential pledges by use of the "AN_proxy" objective. This is a synchronization objective primarily intended to be flooded on a single link using the GRASP Flood Synchronization (M_FLOOD) message. In accordance with the design of the Flood message, a locator consisting of a specific link-local IP address, IP protocol number and port number will be distributed with the flooded objective. An example of the objective is informally:

["AN_proxy", F_SYNCH, 1, "BRSKI-TLS"]

The formal CDDL definition is

proxy-objective = ["AN_proxy", F_SYNCH, 1, method]
method = text ; name of the BRSKI method supported

The loop-count is fixed at 1 since this is a link-local operation.

The ‘method’ parameter indicates the specific BRSKI method available at the given locator. The initial possible values are "BRSKI-TLS" and "BRSKI-COAP". A proxy that supports more than one method will flood multiple versions of the "AN_proxy" objective.

2.2. Negotiation Alternative for Proxy

This alternative to "AN_proxy" uses additional features of GRASP. It requires additional complexity in the pledge, and requires the pledge to announce its existence to any on-link eavesdroppers via a Discovery message. It is therefore not recommended on security grounds, but is defined here for completeness.

A pledge discovers a local proxy and negotiates a BRSKI method with it by use of the "AN_join" objective. First, the pledge performs GRASP discovery, with the loop-count set to 1 and limited to link-local addresses. If multiple responses occur, it chooses one by an
implementation-defined method. Then the pledge initiates GRASP negotiation to choose a mutually acceptable BRSKI method.

An example of the objective is informally:

["AN_join", F_NEG, 6, ["BRSKI-COAP","BRSKI-TLS"]]

The formal CDDL definition is

join-objective = ["AN_join", F_NEG, loop-count, [*method]]
method = text ; name of the BRSKI method supported

The parties will negotiate until one side proposes a single BRSKI method and the other side accepts. In the simplest case of immediate acceptance, there will only be two messages (Request Negotiate and End Negotiate). The locator (IP address, IP protocol number, port number) used for the negotiation will be available for the subsequent BRSKI operations, if required.

Note that in the Discovery message, the loop count will be set to 1 to limit discovery to the local link. In the negotiation stage, the loop count will serve its normal purpose (limiting the negotiation to 6 steps in the above example).

3. Objective for Autonomic Control Plane

The Autonomic Control Plane (ACP) [I-D.ietf-anima-autonomic-control-plane] constructs itself without outside intervention. To achieve this, each node needs to identify its link-local neighbors on all interfaces, and agree on a secure connection method with each of them. There are at least two possible approaches for this: a flooding mechanism, in which each node announces itself and its security methods to its neighbors, or a discovery and negotiation mechanism, in which each node actively discovers its neighbors. For the moment this draft describes both methods.

For either method, each node runs an ASA that supports the corresponding objective. This ASA runs permanently, in order to discover or detect new ACP neighbors or to remove failed neighbors.

3.1. Flooding Alternative

A node announces itself to potential ACP peers by use of the "AN_ACP" objective. This is a synchronization objective primarily intended to be flooded on a single link using the GRASP Flood Synchronization (M_FLOOD) message. In accordance with the design of the Flood message, a locator consisting of a specific link-local IP address, IP
protocol number and port number will be distributed with the flooded objective. An example of the objective is informally:

["AN_ACP", F_SYNCH, 1, "IKEv2"]

The formal CDDL definition is

acp-objective = ["AN_ACP", F_SYNCH, 1, method]
method = text ; name of the connection method supported

The loop-count is fixed at 1 since this is a link-local operation.

The ‘method’ parameter indicates the specific connection method available at the given locator. The initial possible values are "IKEv2" and "TLS". A node that supports more than one method will flood multiple versions of the "AN_ACP" objective.

Note that a node serving both as an ACP node and BRSKI proxy may choose to distribute the "AN_ACP" objectives and "AN_proxy" objectives in the same message, since GRASP allows multiple objectives in one Flood message.

3.2. Negotiation Alternative

Each node discovers its neighbours and negotiates a connection method with each one by use of the "AN_ACP" objective. First, the node performs GRASP discovery, with the loop-count set to 1 and limited to link-local addresses. It records each response that it receives within the chosen discovery timeout. Then the pledge initiates GRASP negotiation with each newly discovered peer in turn to choose a mutually acceptable connection method.

An example of the objective is informally:

["AN_ACP", F_NEG, 6, ["IKEv2","TLS"]]

The formal CDDL definition is

acp-objective = ["AN_ACP", F_NEG, loop-count, [*method]]
method = text ; name of the connection method supported

The parties will negotiate until one side proposes a single connection method and the other side accepts. In the simplest case of immediate acceptance, there will only be two messages (Request Negotiate and End Negotiate). The locator (IP address, IP protocol number, port number) used for the negotiation will be available for the subsequent operations, if required.
Note that in the Discovery message, the loop count will be set to 1 to limit discovery to the local link. In the negotiation stage, the loop count will serve its normal purpose (limiting the negotiation to 6 steps in the above example).

4. Objective for Stable Connectivity of Network OAM

For OAM purposes [I-D.ietf-anima-stable-connectivity], a special-purpose ASA, which we will call the NOC ASA, mediates connectivity between NOC systems performing OAM operations and autonomic nodes that can be reached securely via the ACP. This is requires a discovery operation, which could be handled in two ways: the NOC ASA discovers all nodes, or each node discovers the NOC ASA. The latter seems much more practical. However, the NOC will need to know something about each target node, so the corresponding objective is defined as a negotiation objective to allow for this.

An example of the objective is informally:

["AN_NOC", F_NEG, 6, [TBD]]

The formal CDDL definition is

\[
\text{nec-objective} = \text{["AN_NOC", F_NEG, loop-count, [TBD]]}
\]

TBD = any ; node information to be defined

When a node joins the ACP, one of its initial actions must be to perform GRASP discovery for "AN_NOC" and then to send a Request Negotiate message to the NOC ASA supplying TBD. If successfully received, the NOC ASA must rely with an End Negotiate message. From then on, any OAM communication between the NOC and the node in question will proceed over the ACP using the information TBD.

5. Objectives for Intent Distribution

The format and semantics of Intent are not yet defined, although some aspects are discussed in [I-D.du-anima-an-intent]. Here we assume that Intent is supplied to the whole network as a single file and that the file is obtained by each node that needs it via a specific Uniform Resource Identifier, typically a URL. We also assume that Intent, within a given autonomic domain, is qualified by a monotonically increasing version number, so that nodes can determine if their current copy of Intent is out of date. (A timestamp is not used for this purpose, since it would depend on all nodes having consistent clocks.)

Thus, a source of Intent announces itself to all nodes by use of the "AN_intent" objective. This is a synchronization objective primarily
intended to be flooded using the GRASP Flood Synchronization (M_FLOOD) message. An example of the objective is informally:

["AN_intent", F_SYNCH, loop_count, [12345, "https://noc.example.com/Intent/"]]

The formal CDDL definition is

```
intent-objective = ["AN_Intent", F_SYNCH, loop-count, [version-number,uri]]
version-number = uint
uri = text ; URI conforming to RFC 3986
```

A node that needs to obtain or update Intent will use the latest received version of this objective to check if the version number has increased, and will use the given URI to obtain the current Intent if necessary.

6. Objective for Prefix Management

TBD

7. Flood Frequency

Any ASA that floods one of the above objectives should do so at a carefully chosen frequency. Recipient nodes may be starting up, reconnecting, or waking up from sleep, so floods need to be refreshed periodically. On the other hand, excessive flooding will consume bandwidth, CPU and battery capacity throughout the network, and might even resemble a DoS attack. A general guideline is to flood an objective once immediately after its value is initialised or changed, and then repeat the flood at intervals of at least 30 seconds. Additionally, the flooding interval should be slightly jittered to avoid synchronicity with other floods. Finally, the value of a flooded objective should change as rarely as possible (on a timescale of at least minutes, not seconds).

8. Security Considerations

General security issues for GRASP are covered in [I-D.ietf-anima-grasp]. Specific issues that are not mentioned above are discussed in the referenced drafts for BRSKI, ACP, stable connectivity and Intent.
9. IANA Considerations

IANA is requested to add the following entries to the GRASP Objective Names Table registry created by [I-D.ietf-anima-grasp]:
AN_registrar
AN_proxy
AN_ACP
AN_NOC
AN_intent

10. Acknowledgements

TBD.

11. References

11.1. Normative References

[I-D.greevenbosch-appsawg-cbor-cddl]

[I-D.ietf-anima-grasp]

11.2. Informative References

[I-D.du-anima-an-intent]

[I-D.ietf-anima-autonomic-control-plane]

[I-D.ietf-anima-bootstrapping-keyinfra]


Appendix A. Change log [RFC Editor: Please remove]

draft-carpenter-anima-ani-objectives-00, 2018-11-15:

Initial version

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
Huawei Technologies Co., Ltd
Q14, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing 100095
P.R. China

Email: leo.liubing@huawei.com