ANIMA
Internet-Draft
Intended status: Informational
Expires: 4 September 2025

T. Eckert, Ed.
Futurewei Technologies USA
B. Liu
Huawei Technologies

3 March 2025

ACP free "Automation Network Infrastructure" for simple in-network automation (aNI)

draft-eckert-anima-acp-free-ani-00

Abstract

This document describes how to to build the software infrastructure for distributed automation agents using a lightweight variation of the "Autonomic Networking Infrastructure" (ANI), by using the existing ANI domain keying material (certificates and trust anchors) as well as the protocols GRASP and BRSKI protocols, but eliminating the expensive to implement "Autonomic Control Plane" (ACP) and adding proxying "Autonomic Software Agents" (ASA) instead.

The resulting infrastructure is called "automation Network Infrastructure" and can be implemented solely as application level software components on routers, switches or co-located managemenet devices, avoiding the need to change any router or switches forwarding or control-plane protocols.

The aNI achieves mosts of the benefits of the ANI but foregoes the ability to easily make pre-existing, insecure control-plane protocols secure or provide all the same protection against operator or SDN controller misconfigurations that the ACP provides.

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 4 September 2025.

Table of Contents

1. Introduction

1.1. Background

[RFC8993] describes the reference model for Autonomic Networking which consists of a so-called "Autonomic Network Infrastructure" (ANI) and in-network (devices) software agents called "Autonomic Service Agents" (ASA).

ASA can be imagined as simple as scripts in programming languages such as python or javascript developed by bendors, integrators or operators. They are primarily intended to run on network devices such as router or switches, or on control equipment that is also decentrally deployed, such as management servers in network equipment locations often called point of presence (PoP). These agents can operate alone or in support of centralized network management systems including controller or orchestrators.

One of the core components of that ANI is the so-called "Autonomic Control Plane" (ACP, [RFC8994]), a set of functionalities establishing an autonomously (zero-touch) created and maintained VRF across all network that is primarily intended to provide always-on network reachability even in the absence of any operator or management established provisioning of the nodes. The ACP then allows operators, centralized management software or ASA to communicate across it to manage the network and for example provision both the basic network addressing and routing (so-called data-plane) or specific subscriber services or to perform monitoring actions/ automations. See also [RFC8368].

Unfortunately, implementing an ACP in network devices, especially those with legacy operating system software infrastructures can be a challenging exercise, and as of today, no widely adopted production implementations on commercial routers exist.

Without an ANI, it is challenging to build simple automation agents running on (or near to) network devices because they are missing functionalities not ubiquitously found in networks today: credentials for secure communications with mutual trust, connectivity and discovery of other agents, defining new protocols for communication between agents and ability to communicate when the network or specific nodes are not yet configured for correct end-to-end reachability or when that reachability is broken.

This document describes the mechanisms how these support functions can be supported for ASA and via ASA. The resulting design is called by this document the "automation Network Infrastructure" (aNI). Lowercase ’a’ is used to distinguish it from the ANI which does include an ACP.

Like the ANI, the aNI is defined so that it does not introduce dependencies against external, centralized components, such as orchestrators or management controllers. Like the ANI the aNI can therefore support those centralized components.

1.2. Overview

This document introduces the concepts and components of an ACP-free automation Network Infrastructure (aNI), which leverages the components developed for the ANI except for its ACP. The ACP is replaced by the assumed to be pre-established data-plane of the network, and as necessary by proxy ASA.

Unlike the ANI with ACP, this aNI solution can easily be introduced into existing networks simply though the development of control-plane programs, for example developed in scripting languages such as python or javascript (whatever can best run on routers). The aNI does not require any changes to routers forwarding planes and does not expect more than basic IPv4 and/or IPv6 end-to-end connectivity across segments of the network.

In summary, the aNI differs from the ANI as follows:

* The networks existing and assumed to be pre-configured IPv4 or IPv6 connectivitiy (called data-plane) is used to provide end-to-end connectivity for GRASP or other protocols used to build ASA. Unlike the ACP, IPv4 is a fully permitted option, especially because large and complex industrial networks will continue to depend on it because it has some significant simplicity benefits over IPv6 in options such as [RFC1819] addressing. Nevertheless, for any new deployments where there is no clear benefit of IPv4 over IPv6, IPv6 is recommended.

* Discovery between ASA and between ASA and central network management components utilizes GRASP in the same ways as it does with ACP. It requires on every network node a GRASP ASA, which is a lightweight (potentially scripted in python) user-level process that forwards GRASP discovery messages hop-by-hop. This GRASP ASA can automatically be started and requires no configuration. The hop-by-hop connections between these GRASP ASA is TLS.

* Mutual trust for end-to-end GRASP connections between ASA, and between GRASP agents, but also for any other existing protocols that may be used is based on the domain certificate model of [RFC8994]: Each node is identified by a certificate which also identifies the (aNI) domain and also indicates the primary network layer address of the node which is used for aNI communications. Like the ANI, the aNI can therefore operate without any dependencies against DNS.

* The aNI encourages re-use of any existing protocols such as HTTPS or others, that can help to avoid re-coding any already working ASA functionality. GRASP is recommended whenever new designs could be easier than with potentially more complex frameworks that exist for HTTPS or CoAP(s). Combinations of existing protocols with GRASP is also a recommended option, for example to use GRASP to automate existing protocols by amending announcement and discovery via GRASP and therefore eliminating manual or SDN controller based provisioning steps.

* All signaling protocols that are considered to be part of the aNI MUST use transport layer encryption of at least the security provided by TLS1.3. If there are any existing or new protocols that do not meet this expectation, they are simply not considered to be part of the aNI. For example existing routing protocols do typically not confirm to this level of security. They can continue to operate unaffected from aNI. They just do not conform to the security level of the aNI.

* Enrollment of security credentials for new network rollouts is recommended to use BRSKI ([RFC8995]) or any specified variation thereof, depending on the operational requirements of the network enrollment process. In existing and pre-configured networks, alternatives such as netconf zero-touch with certificate enrollment may be viable alternatives, but are not equally comprehensively document as a solution.

* To support automation in the presence of missing end-to-end connectivity between all necessary nodes including new to-be-provisioned "pledges", proxies are used. These proxies likely are a combination of generic transport proxies (e.g.: HTTP proxies) or service specific proxies (ASA with proxy functionality) depending on requirements.

1.3. Architecture Example

The following Figure 1 summaries the archticture components. These are all introduced in more details in the following architecture section.

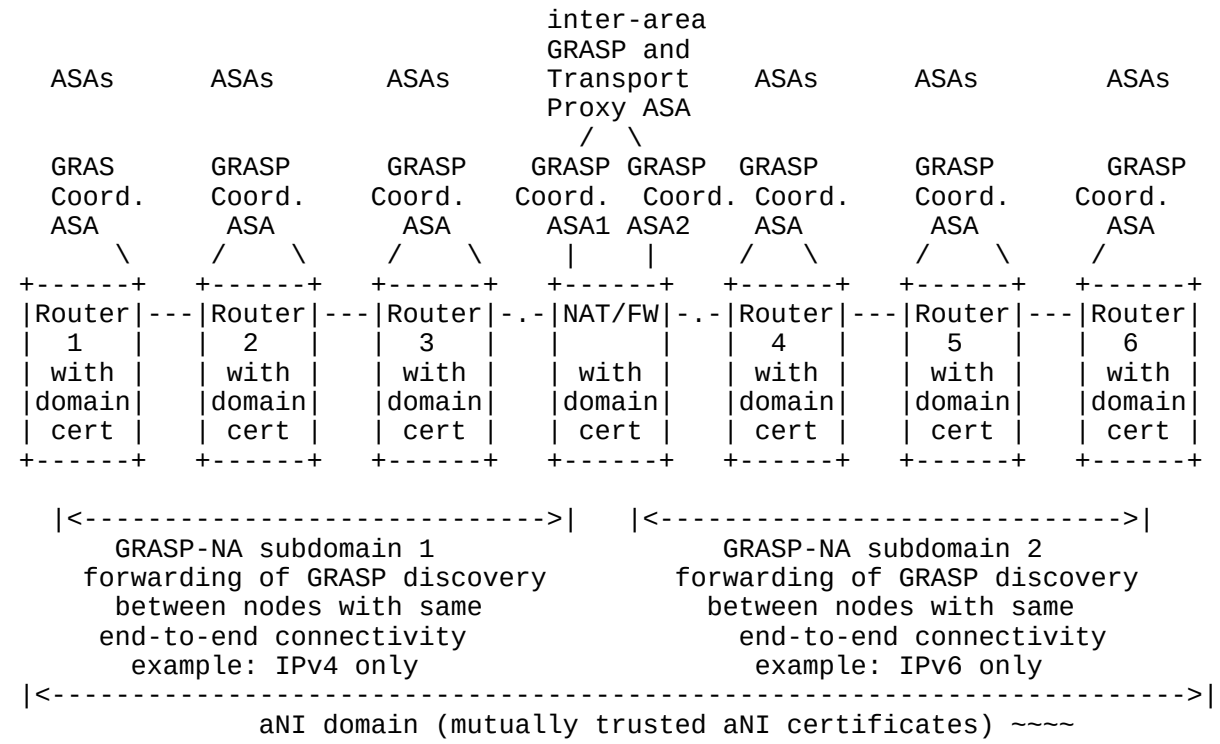

Figure 1: aNI Architecture example

The example architecture picture outlines the most relevant aspects of an aNI that where introduced in the overview section above and where they are the same or differ from the ANI with ACP.

* An aNI domain is the set of all nodes using the same aNI domain credentials (certificates and trust anchors). These are using the same concept as ACP certificates/trust anchors.

* Enrollment of keying materials can use any protocol but prefers BRSKI (as in ACP).

* Addressing can use IPv4 and/or IPv6 and is solely the data-plane of the existing network.

* aNI domains may be subdivided into areas connected by inter-area nodes, as shown in the picture a NAT/FW. Inside each area unrestricted connectivity between ASA is expected, across areas it is not.

* Inside each area, GRASP coordination agents enable distribution of information and service discovery. This is the equivalent of GRASP inside the ACP ([RFC8994]).

* Between areas, forwarding of GRASP messages is managed by inter-area GRASP proxy ASA as well as existing specialized forwarding such as NAT and Firewall forwarding, or additional user-level transport proxy ASA. The latter are generalizing the concept of BRSKI or HTTP proxies.

2. Architecture

2.1. Credentuals and mutual trust

aNI relies on the same domain trust model as the ANI. All nodes in an aNI domain are expected to have an aNI certificate and trust anchor(s) that allow to verify and authenticate the aNI certificates of other domain members.

aNI certificates carry an aNI node name attributes, which differs from acp-node-names ([RFC8994], section 6.2.2) as follows:

* The address part can either be an IPv6 address or an IPv4 address. (ABNF TBD). The aNI address of a node is a data-plane address that is known to be permanently assigned to the node, routable and reachable across a segment of the aNI domain. There is no new address assignment with ULA addresses as in the ACP.

* This document does not discuss the more complex additional options such as extensions or routing subdomains, but syntactically they are possible as they are for ACP certificates.

* Encoding into the X.509 aNI domain certificate is via a new "OTHER-NAME" attribute to allow distinguishing it from ACP addresses. A node can therefore have a single certificate with both an aNI name element and (potentially later) an ACP name element without conflicts between them.

Enrollment of these certificates is, as in [RFC8994] by arbitrary methods, preferrably BRSKI. BRSKI details for aNI are discussed further domain in this document.

2.2. Area segmented connectivity

aNI domains may not have transparent end-to-end connectivity across all nodes. Instead, thay may be partitioned by nodes implementing functionality such as NAT or Firewall which provide only partial connectivity. Or they may be partitioned without any connectivity because of different VRFs. Or they may the partitioned by different network layers. One part of the network may only use IPv4, another only IPv6.

An aNI connectivity area is a contiguous set of nodes between which the data-plane provides in the absence of errors transparent end-to-end connectivity for all the traffic desired for the aNI. This primarily involves traffic between ASA, but also traffic considered to be used in conjunction with ASA or aNI, such as traffic between central management nodes (controller/orchestrators), servers (NTP, DHCP, ...) and other required sockets on nodes. This means that there may be filtering and limitations of traffic at the edge or inside such aNI areas, for example to prohibit traffic between these nodes and nodes outside the area (such as subscriber nodes).

Note that these requirements should allow to add the aNI without change of any existing data-plane setup in most existing networks (private or service provider), and that many of them do not even need to consider multiple aNI connectivity areas.

Note that that incorrect configuration of filtering of filtering will cause errors in the aNI in the same way as it does cause errors for any other management traffic. The aNI does not - unlike the ACP - protect against such misconfiguration.

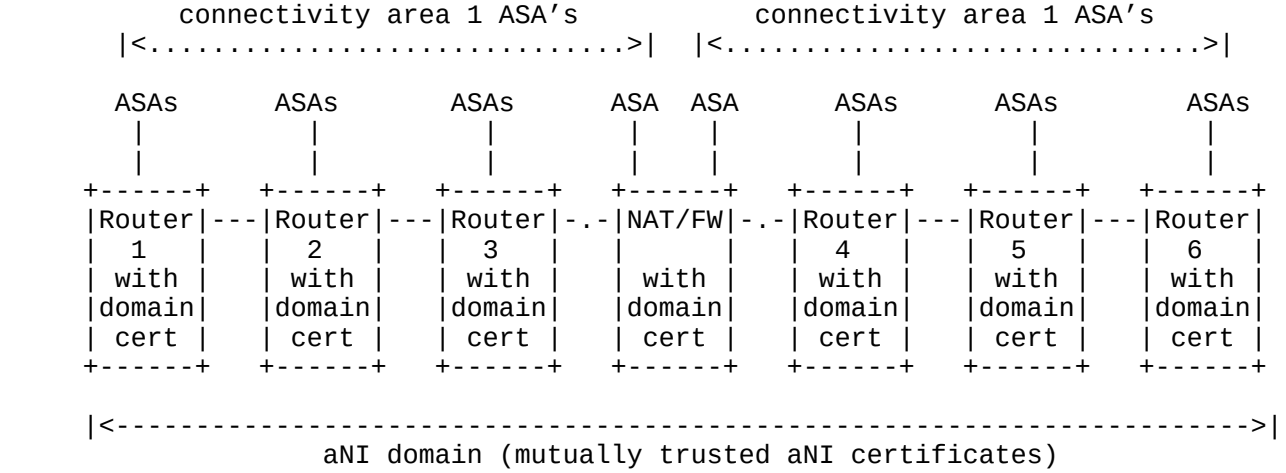

Figure 2: aNI domain and areas

Figure 1 shows a simple example of a single aNI domain with two connectivity areas, for example because of a NAT/FW node separating the two areas. In the most simple case, an ASA connects only into a single area and sends/receives traffic only within a single area. There is no connectivity between ASA across the two areas in this example.

2.3. End-to-end transport protocols

End-to-end connections considered to be part of the aNI MUST use a transport layer protocol with at least the level of encryption/ security as TLS1.3. Compared to the ACP this means that it is not possible to simply use existing, non-end-to-end transport layer encrypted protocols such as (non-TLS version of) DHCP, DNS, NTP, SNMP or other common (and older) management protocols. If and where TLS/ DTLS (or QUIC) variations to the protocols exist, they of course are applicable to the aNI.

Connunication SHOULD use existing protocols and extend them as necessary instead of re-inventing new protocols unnecessary. New protocol development for purposes for which no suitable existing protocols are available SHOULD use GRASP. This applies to peer-to-peer and client-server connectivity between ASA or any other traffic considered to be part of the aNI.

2.4. GRASP security and transport substrate

[RFC8990] requires that GRASP relies on a "security and transport" substrate, which in [RFC8994] is the ACP, TLS >= 1.2 and ACP domain certificates for mutual authentication.

In the aNI, the "security and transport" substrate is the data-plane for connectivity, meaning the pre-existing IPv4 and/or IPv6 connectivity, TLS >= 1.3 for any GRASP connection (except for link-local DULL GRASP for link-local discovery), and aNI domain certificates for mutual authentication. The security considerations effectively ae the same as for GRASP across an ACP, but end-to-end (unicast) GRASP connections depend on proper functioning of data-plane routing. Workarounds for this are descussed later in this document.

2.5. Distributed ASA coordination via GRASP

ASA need to be able to self coordinate and orchestrate. Responders need to be able to announce themselves to be discovered and selected by responders. ASA may have other information that needs to be disseminated to multiple interested ASA across the domain.

GRASP supports these operations through Discovery and Flood messages which are flooded hop-by-hop through the network by GRASP coordination agents. Loops are prevented by sequence number duplicate elimination on reception, so that these agents donot require any routing information. These age that can easily be implemented, for example as lightweight python scripts.

The procedures for GRASP coordination agents are the same as described in [RFC8994]. Nodes discover their neighbors on interfaces via DULL GRASP, and they build TLS1.3 connections to their neighbors, mutually authenticating each others by their aNI certificates. DULL GRASP uses a new port number (IANA assignment TBD) to distinguish it from DULL GRASP for the ACP. Therefore, GRASP in the aNI can co-exist with GRASP in an ACP, hence a migration from an aNI to a full ANI with ACP is easily possible.

aNI nodes MUST support a GRASP coordination ASA. They MAY support other methods for coordination and orchestration including service announcement discovery and selection, such as DNS, but this specification does not specify any way how to automatically establish a sufficient DNS infrastructure and hence also no functionality that depends on DNS. Instead, DNS-SD compatible service announcement, discovery and selecction is suggested to rely on GRASP as described in [I-D.eckert-anima-grasp-dnssd].

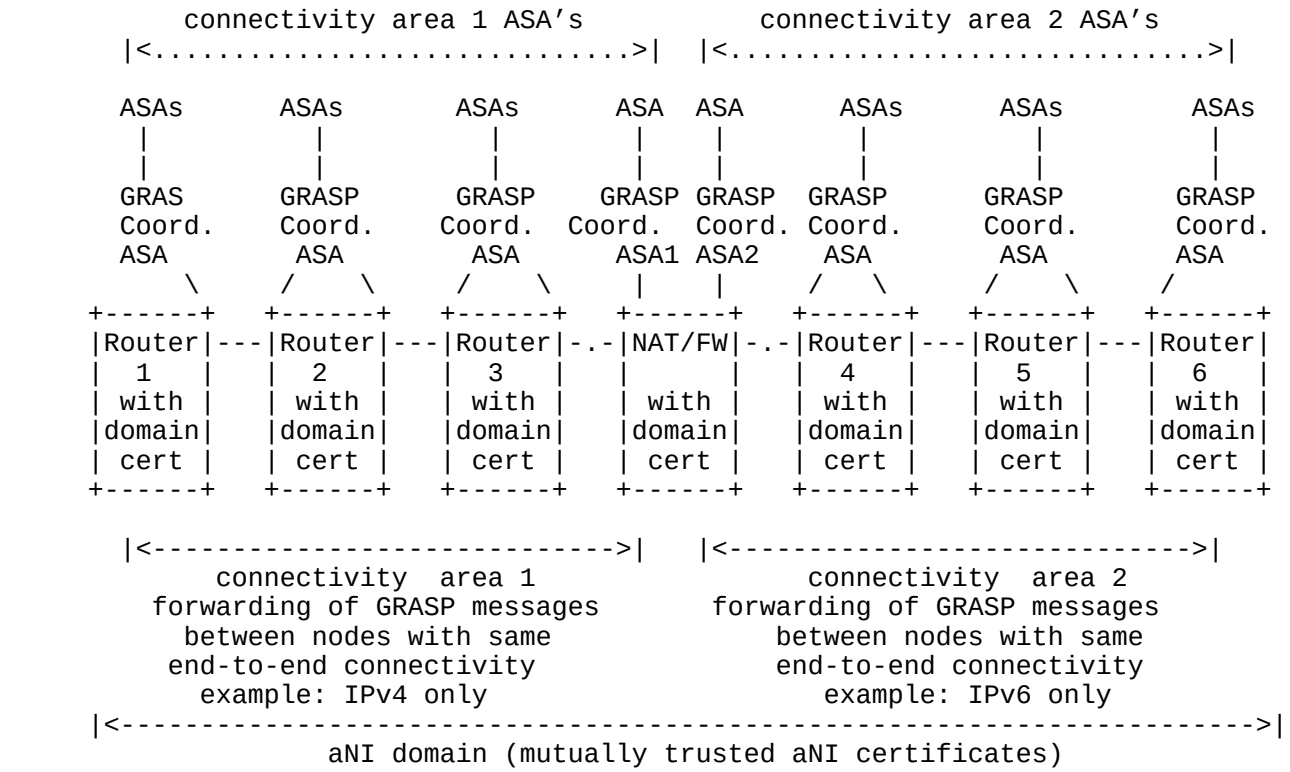

Figure 3: GRASP connectivity agents

Figure 2 above shows the GRASP coordination ASA added to the prior example. GRASP agents need to be able to determine all interfaces that belong to the same area and forward messages only between those interfaces. If the node has interfaces in multiple areas, a separate instance of forwarding of GRASP messages needs to be run for each area. In the example, this is shown for the NAT/FW node, which has interfaces in two areas.

Note that ASA only need to rely on the GRASP coordination agent for sending and receiving of GRASP coordination messages. GRASP "unicast" traffic to other nodes can simply use direct TLS connections to the nodes ASA.

2.6. Inter-area communications

This section explains how to implement inter-area ASA communications using the model shown in Figure 4.

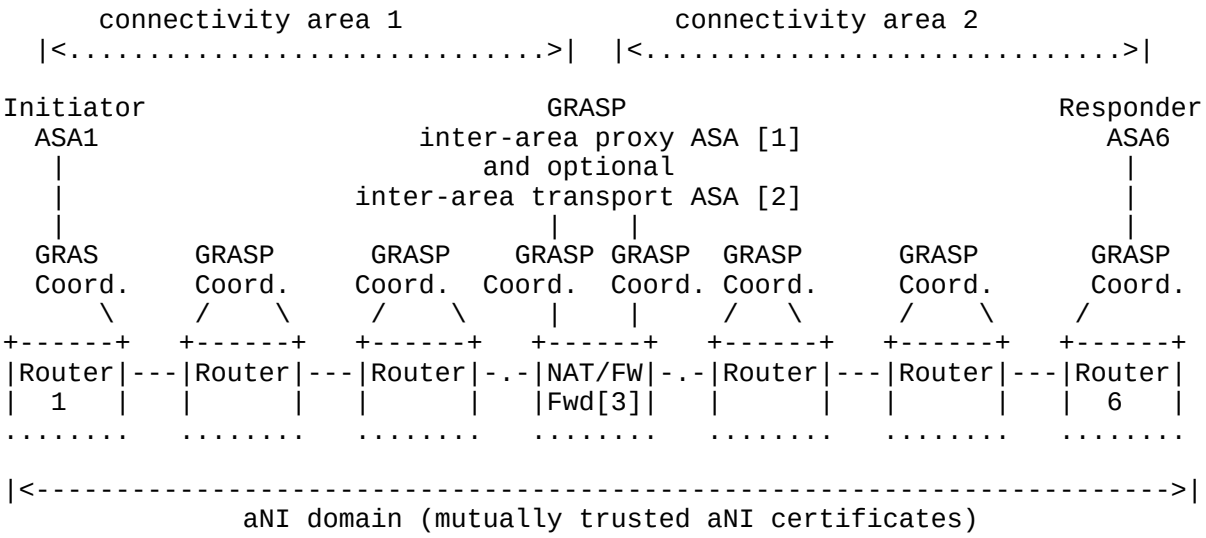

Figure 4: Inter Area communications

A responder ASA6 on Router 6 in area 2 intends to provide services also to be consumable by an Initiator ASA1 on Router 1 in area 1.

Direct inter-area network layer connectivity from ASA1 to ASA6 may be possible in some cases, such as when area 1 is on the inside of a NAT and area 2 is on its outside. However, ASA1 would have no way to even discover the availability of ASA6 without the introduction of the elements described in this section, because no GRASP messages are forwarded by the GRASP coordination ASA across area boundaries. Likewise in the opposite case, when ASA6 is on the inside, and ASA1 is on the outside, not only does a connection require likely special setups, but it is also a matter of additional policy if that type of communication is even desired. Likewise are the conditions different, when the impeding element is a Firewall or the areas are different VRF.

The three type of components of the solution are called the GRASP inter-area proxy ASA [1], optional inter-area transport ASA [2] and Forwarding functionality [3] in the forwarding plane of the inter-area node.

2.6.1. Inter-area information flooding

The most simple example of inter-area forwarding is that of GRASP Flood messages used for information distribution. All information is contained in the GRASP Flood message and thus no dependend inter-area communications is expected. Examples of such information distribution could be simple router configurations for common functions. Nevertheless, forwarding of such information betwween areas does very likely need to be policy filtered if not policy modified. This is a job of specific GRASP inter-area proxy ASA.

2.6.2. Loop prevention in inter-area GRASP flooding

In any case where GRASP inter-area proxy ASA flood GRASP messages across area boundaries, care must be taken to avoid looping messages. This specifically means that the GRASP signaling elemtn that is used to detect looping messages must not be changed when passing GRASP messages to another area, and the identifier in it needs to be unique across areas.

2.6.3. Inter-area GRASP policy filtering

Inter-area flooding of GRASP messages should support policy filtering independent of the below described mechanisms to ensure the necessary connectivity. This policy filtering may be derived automatically from specific subset of the Objective namespace or other novel GRASP signaling elements.

2.6.4. Inter-area service announcements

As described in this sections introduction, in many cases a GRASP Flood Message may be a service announcement for one or more responder sockets of the announcer, or a third-party node (in case the announcement is not directly from the responder ASA).

In this case, there is the expectation that nodes receiving this announcement may want to initiate connections to those responder sockets. The inter-area forwarding ASA does thus need to understand if or how those responder sockets can be connected to from the area into which it forwards the GRASP Flood message.

In the aforementioned outside-to-inside flooding across a NAT inter-area node, the GRASP inter-area proxy ASA does not need to change anything in the GRASP Flood message if inside and outside use the same IP address family.

If instead the desired reachability is from outside to inside, or if the address families differ, then it is typically necessary to set up specific NAT/PAT between inside and outside to enable the communication, and to also change the announced service address in the announced GRASP Flood to the other area.

The use of GRASP in all these cases does cleanly resolve the problem that such communication setups are facing when the service announcement and discovery are not readily available in-band on the inter-area nodes, which today is almost always the case.

In one example, ASA6 is on the inside, uses an RFC1918 addres, but its service should be available on the outside. The GRASP inter-area proxy ASA learns about the service and establishes the necessary PAT, mapping a port available on the outside to that inside service, and accordingly adjusts the GRASP announcement before it passes it to the outside area towards ASA1. When ASA1 then connects, it effectively connects to the outside address of the inter-area node, where the existing NAT forwarding plane connects it to the inside RFC1918 address and server port for ASA6.

2.6.5. inter-area transport proxy ASA

Instead of relying on such pre-existing/configurable forwarding plajne connectivity, such inter-area connectivity can and depending on use case must be implemented at user-level by so-called inter-arrea transport proxy ASA. These are typically what in BRSKI (and HTTP) terms are called connection or stateful proxies, but they could equally be stateless proxies if this is ensured to be supported by the responders.

(stateful) BRSKI proxies simply act on one side as TCP responders, recreating a new TCP connection on the other side as initiators and then forwarding traffic bidirectionally across that proxies connection. HTTP proxies operate as HTTP message proxies, but can also be turned into TCP connection proxies through the HTTP CONNECT method.

2.7. Pledge Bootstrap

With ANI, new nodes (pledges) for a domain are bootstrapped by provisioning them first with domain credentials via BRSKI and then provisioning them with any desirable protocols via the ACP.

With aNI, these pledges do not have end-to-end data-plane connectivity after BRSKI enrollment. They still will likely only have link-local connectivity. While automatic DHCP or IPv4 SLAAC connectivity may be something a data-plane provides for end-user nodes, this is typicaly not the case for router in the domain that are newly added (or replaced).

For this reason, the aNI requires some form of proxy-connectivity ASA setup for such newly enrolled network nodes on a link-local connected aNI domain node, for example the same node that was providing the BRSKI proxy ASA. This of course is only necessary if those new nodes are assumed to be remotely configured/provisioned instead of being able to fully self-provision through appropriate autonomous service agents.

2.7.1. Unsecured pledges bringup

If new devices need to be brought into a domain that does not have BRSKI or another stand-alone mechanism to provision domain credentials, then a bootstrap proxy agent on a neighboring domain node can provide connectivity to such a pledge. This ASA would involve the following aspects:

* The ASA is capable to discover the presence of such pledges through any pre-existing protocol on the pledge. This can include LLDP, or simply the signaling elements that may be provided by by the pledge in DHCP requests and help to identify it. Use of such pre-existing mechanisms would allow to avoid expecting any changes to pledges software in factory-fresh condition. This is especially valuable on pledges that allow to later install additional softeware easily, such as ASA, e.g.: after enrolment.

* The bootstrap proxy agent on this neighboring node indicates the presence of such a new pledge through an appropriate new-pledge-announcement protocol to an enrollment service that may run on a central management node. For example that server announces via GRASP its availaility and the bootstrap proxy agent signals presence of new plege(s) to that GRASP discovered server.

* The bootstrap proxy agent implements or orchestrates a transport proxy that allows the enrollment servers to connect to sockets on the pledge via it. known or assumed to allow remote configuration. For example, the transport proxy could be a simple HTTPS server implementing only the CONNECT method to such ports on directly connected pledges. The prior step GRASP signaling would inform the management node of the discovered (link-local) addresses and port(s) on those pledges through appropriate URLs. Once the management station connects to the bootstrap proxy agent and issues the HTTP CONNECT, it has a transparent TCP connection to the plege.

2.7.2. BRSKI pledges

If the pledge can already be enrolled with BRSKI or an equivalently secure alternative protocol into the aNI domain, there are a range of more options, but it may be a good start to simply use the same mechanisms as for an unsecured pledge except for the following.

Any connections into the pledge after being enrolled via BRSKI SHOULD only be via TLS 1.3 or better and use the aNI domain credentials for authentication. Ideally, no insecure ports should need to be open on such a to-be-provisioned node.

2.7.3. aNI bootstrap

Whether a new, to-be-provisioned node uses BRSKI or not, any aNI functionality such as the GRASP coordination agent or any other ASA SHOULD be enabled through the provisioning system only after the required data-plane connectivity has been provisioned.

This is beneficial, so that the GRASP coordination agent can determine the necessary information about which interface belongs to which area from the data-plane configuration. In the most easy provisioning option, the data-plane configuration explicitly labels interfaces or addresses with aNI area numbers, so that the GRASP coordination agent has this information readily available. Otherwise, areas have to be determined by examining VRF membership and NAT inside/outside of interfaces/addresses, to automatically determine which area they should be assigned to.

2.8. Inter-domain aNI communications

Many use cases require communications between ASA that are not in the same automation domain. This section discusses options how to support this.

2.9. Using aNI domain credentials

Secure communication via TLS 1.3 or any other equal or better strength transport protocol with mutual certificate authentication with a peer in a different aNI domain requires mutual trust in the other domains aNI certificates. If for example the domain of a peer1 is peer1domain.example.com, then secure transport connections to peer1 needs a prior provisioning of the trust anchors for peer1domain.example.com on the local node.

aNI certificates and trust-anchors, just like ACP certificates and trust anchors are not required to be WebPKI certificates, so no trust or enrolment into WebPKI systems is required. This allows aNI to contually operate without any Internet connectivity. It also eliminates any challenges that would otherwise easily be introduced with the desire for the certificate to contain private information such as the aNI domain information field. Equally, enrollment and renewal of certificates is easily and automated with BRSKI or other protocol alternatives.

In return to these benefits of private aNI certificates, it is necessary to provision mapping information between a trusted remote aNI domain trust anchors and that domains domain name.

2.10. Using federated aNI credentials

To avoid configuring the above described aNI credential to domain mapping, typical deployhment cases such as service provider to customer interdomain connections, or collaborating service provider connections could rely on a common trust anchor and shared private (e.g.: BRSKI based certificate enrollment) system. With such a setup, the single root trust anchor of that federation would allow to authenticate all federation member certificates, whereas the trust anchor of each domains aNI is an intermediate trust anchor, allowing all intradomain security to be managed in the same way as without a federation.

2.11. Using WebPKI certificates

If using WebPKI certificates, including enrollment and renewal, is feasible, including the typical necessity of Internet connectivity, then aNI nodes requiring interdomain connections could also use such WebPKI to communicate with each other, reducing the operational complexity to simply configuring which web domain a remote peer is expected/permitted to be from.

3. Summary

In-network automation through simply programmed software agents called ASA requires a set of underlying infrastructure functions. In the ANI, [RFC8993], these are provided through the combination of ACP, BRSKI and GRASP.

This document outlines how the mayority of this ANI functionality can be provided without the need for the expensive to implement ACP, by simply making ASA rely on the the already existing connectivity in the network, the so-called data-plane.

The resulting infrastructure module introduced in this document is called aNI - automation Network Infrastructure and can be implement solely as application-level software running on switches, routers or co-located management nodes, not requiring any changes to existing IPv6 and/or IPv6 routing and forwarding planes, but instead relying solely on transport-layer security (authentication and encryption of signaling protocols).

Most importantly, the aNI can also work in the presence of the wide range of connectivity impairing functions in networks such as firewalls, NATs, traffic filtering and VRFs. In the ANI, the ACP provided a parallel, unfethered IPv6 connectivity to overcome such impairments. In contrast, in the aNI proxy ASA are required to establish connectivity, ranging from simple GRASP service announcement proxies that provide the missing connectivity when the impairment is because of NAT/PAT, to actual GRASP signaling and transport connection proxies, when there is no connectivity possible otherwise. This approach generalizes the approach already used by BRSKI proxies.

The fundamental security of the aNI is the same as in the ANI: domain certificates with automatic enrolment via BRSKI or other protocols. Like proposed in the ANI, role-based extensions can help to provide more fine-grained authentication.

The main "shortcoming" of the aNI compared to the ANI is that it does not provide (in its current version) transparent end-to-end security for pre-existing unsecured signaling protocols such as NTP, SNMP, DHCP, DNS, TFTP or other commonly used protocols for the networks control and management plane. This is because since the definition of the ACP, secure versions or variations of these protocols have become more commonplace, and it is thus more appropriate today to expect for those secure evolutions to be used instead of investing large amounts of efforts to layer security underneath old protocol options. aNI domain certificates enable the seamless security for any protocol with TLS 1.3 or better transport layer security options - including of course also datagram protocols with DTLS 1.3.

4. Security considerations

4.1. Proxying and Security

Providing remote access to not configured or incorrectly configured nodes constitutes a significant security challenge because those nodes are most often vulerable to attacks, with typical security issues such as open ports with default passwords.

All proxying of connections towards such nodes described in this document is designed such that it can only be initiated from trusted nodes with aNI domain certificates - with the assumnption that ithis is a sufficient level of security to ensure that the initator is not malicious. If this level of security is deemed insufficient, then more fine-grained role based authentication can be added to aNI domain certificates as already outline for ANI domain certificates in [RFC8994], limiting use of such connection proxies to more trustworthy nodes such as management stations.

4.2. Infrastructure security

By not building an ACP, the aNI does not have the same level of infrastructure security as an ANI. Specifically attackers that gain access to physical links between nodes can inject packets and attempt to attack any weak (responder) sockets of network equipment. The ACP prohibits this because it carries all control plane traffic across encrypted point-to-point tunnels. Nevertheless, the aNI does expect that all control plane considered to be part of the aNI is protected by certifcate based transport layer security, so it does conform to todays best established standards for end-to-end security and extends it into the hop-by-hop infrastructure.

To compensate for this lack of infrastructure security, it is recommended to not only deploy the "clamshell security" model common in service provider networks, but to also design ASA that automate

its establishment: Network connectivity to and from IPv4/IPv6 addresses used between nodes of the aNI SHOULD be filtered on the edge of an aNI domain in the forwarding plane (with destination IPv4/IPv6 address ACL) so that attackers without access to a physcial link between aNI node can not inject the above described attacks.

5. References

5.1. Normative References

[RFC1819] Delgrossi, L., Ed. and L. Berger, Ed., "Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+", RFC 1819, DOI 10.17487/RFC1819, August 1995, <https://www.rfc-editor.org/rfc/rfc1819>.

[RFC8368] Eckert, T., Ed. and M. Behringer, "Using an Autonomic Control Plane for Stable Connectivity of Network Operations, Administration, and Maintenance (OAM)", RFC 8368, DOI 10.17487/RFC8368, May 2018, <https://www.rfc-editor.org/rfc/rfc8368>.

[RFC8990] Bormann, C., Carpenter, B., Ed., and B. Liu, Ed., "GeneRic Autonomic Signaling Protocol (GRASP)", RFC 8990, DOI 10.17487/RFC8990, May 2021, <https://www.rfc-editor.org/rfc/rfc8990>.

[RFC8993] Behringer, M., Ed., Carpenter, B., Eckert, T., Ciavaglia, L., and J. Nobre, "A Reference Model for Autonomic Networking", RFC 8993, DOI 10.17487/RFC8993, May 2021, <https://www.rfc-editor.org/rfc/rfc8993>.

[RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <https://www.rfc-editor.org/rfc/rfc8994>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

5.2. Informative References

[I-D.eckert-anima-grasp-dnssd]
Eckert, T. T., Boucadair, M., Jacquenet, C., and M. H. Behringer, "DNS-SD Compatible Service Discovery in GeneRic Autonomic Signaling Protocol (GRASP)", Work in Progress, Internet-Draft, draft-eckert-anima-grasp-dnssd-07, 7 July 2024, <https://datatracker.ietf.org/doc/html/draft-eckert-anima-grasp-dnssd-07>.

Appendix A. Changelog

A.1. draft-eckert-anima-acp-free-ani-00

Initial version

Authors' Addresses

Toerless Eckert (editor)
Futurewei Technologies USA
United States of America
Email: tte@cs.fau.de

Bing Liu
Huawei Technologies
P.R. China
Email: leo.liubing@huawei.com