Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 22, 2014

M. Behringer
S. Bjarnason
Balaji. BL
T. Eckert
Cisco

June 20, 2014

An Autonomic Control Plane
draft-behringer-autonomic-control-plane-00

Abstract

In certain scenarios, for example when bootstrapping a network, it is desirable to automatically bring up a secure, routed control plane, which is independent of device configurations and global routing table. This document describes an approach for an "Autonomic Control Plane", which can be used as a "virtual out of band channel" - a self-managing overlay network, which is independent of configuration, addressing and routing on the data plane.

Status of This Memo

This Internet-Draft will expire on December 22, 2014.

Copyright Notice

Table of Contents

1. Introduction

Today, the management and control plane of networks typically runs in the global routing table, which is dependent on correct configuration and routing. Misconfigurations or routing problems can therefore disrupt management and control channels. Traditionally, an out of band network has been used to recover from such problems, or personnel is sent on site to access devices through console ports. However, both options are operationally expensive.

In increasingly automated networks either controllers or autonomic service agents in the network require a control plane which is independent of the network they manage, to avoid impacting their own operations.

This document describes a self-forming, self-managing and self-protecting "Autonomic Control Plane" (ACP) which is inband on the network, yet independent of configuration, addressing and routing problems. It therefore remains operational even in the presence of

configuration errors, addressing or routing issues, or where policy could inadvertedly affect control plane connectivity. It serves as a "virtual out of band channel": An operator can use it to log into remote devices in such cases. And an SDN controller can use it to securely bootstrap network devices in remote locations, even if the network in between is not yet configured; no data-plane dependent bootstrap configuration is required. An example of such a secure bootstrap process is described in [I-D.pritikin-bootstrapping-keyinfrastructures]

2. Problem Statement

An "Autonomic Control Plane" (ACP) provides a solution to some of today's operational challenges. These fall into three broad categories:

- Bootstrapping a network while devices are not yet configured. Bootstrapping a new device today requires all devices between the controller and the new device to be completely and correctly addressed, configured and secured. Therefore, bootstrapping a network happens in layers around the controller. Without console access it is not possible today to make devices securely reachable before having configured the entire network between.

- Maintaining reachability of network devices even in the case of certain forms of misconfiguration and routing issues. For example: certain AAA misconfigurations can lock an administrator out of a device; routing or addressing issues can make a device unreachable; shutting down interfaces over which a current management session is running can lock an admin irreversibly out of the device. Traditionally only console access can help recover from such issues.

- Data plane dependencies for NOC/SDN controller applications: Certain network changes are today hard to operate, because the change itself may affect reachability of the devices. Examples are address or mask changes, routing changes, or security policies. Today such changes require precise hop-by-hop planning; an ACP would simplify them.

3. Self-Creation of an Autonomic Control Plane

This section describes the steps to set up an Autonomic Control Plane, and highlights the key properties which make it "indestructible" against many inadvert changes to the data plane, for example caused by misconfigurations.

3.1. Preconditions

Each autonomic device has a globally unique domain certificate, with which it can cryptographically assert its membership of the domain. The document [I-D.pritikin-bootstrapping-keyinfrastructures] describes how a domain certificate can be automatically and securely derived from a vendor specific Unique Device Identifier (UDI) or IDevID certificate.

3.2. Adjacency Discovery

Adjacency discovery exchanges identity information about neighbors, either the UDI or, if present, the domain certificate (see Section 3.1. This document assumes the existance of a domain certificate.

Adjacency discovery provides a table of information of adjacent neighbours. Each neighbour is identified by an globally unique device identifier (UDI).

The adjacency table contains the following information about the adjacent neighbours.

- Globally valid Unique devide identifier (UDI)

- Link Local IPv6 address

- Trust information

- Validity of the trust

Adjacency discovery can populate this table by several means. One such mechanism is to discover using link local multicast probes, which has no dependency on configured addressing and is preferable in an autonomic network.

3.3. Authenticating Neighbors

Each neighbour in the adjacency table is authenticated. The result of the authentication of the neighbour information is stored in the adjacency table. We distinguish the following cases:

- Inside the domain: If the domain certificate presented is validated to be in the same domain as that of the autonomic entity then the neighbour is deemed to be inside the autonomic domain. Only entities inside the autonomic domain will by default be able to establish the autonomic control plane. An ACP channel will be established.

- o Outside the domain: If there is no domain certificate presented by the neighbour, or if the domain certificate presented is invalid or expired, then the neighbour is deemed to be outside the autonomic domain. No ACP channel will be established.

3.4. Capability Negotiation

Autonomic devices have different capabilities based on the type of device and where it is deployed. To establish a trusted secure communication channel, devices must be able to negotiate with each neighbour a set of parameters for establishing the communication channel, most notably channel type and security type. The channel type could be any tunnel mechanism that is feasible between two adjacent neighbours, for example a GRE tunnel. The security type could be any of the channel protection mechanism that is available between two adjacent neighbours on a given channel type, for example IPSEC. The establishment of the autonomic control plane can happen after the channel type and security type is negotiated.

3.5. Channel Establishment

After authentication and capability negotiation autonomic nodes establish a secure channel towards their direct AN neighbours with the above negotiated parameters. In order to be independent of configured link addresses, these channels can be implemented in several ways:

- o As a secure IP tunnel (e.g., IPsec), using IPv6 link local addresses between two adjacent neighbours. This way, the ACP tunnels are independent of correct network wide routing. They also do not require larger than link local scope addresses, which would normally need to be configured or maintained. Each AN node MUST support this function.

- o L2 separation, for example via a separate 802.1q tag for ACP traffic. This even further reduces dependency against the data plane (not even IPv6 link-local there required), but may be harder to implement.

3.6. Context Separation

The ACP is in a separate context from the normal data plane of the device. This context includes the ACP channels IPv6 forwarding and routing as well as any required higher layer ACP functions.

In classical network device platforms, a dedicated so called "Virtual routing and forwarding instance" (VRF) is one logical implementation option for the ACP. If possible by the platform SW architecture,

separation options that minimize shared components are preferred. The context for the ACP needs to be established automatically during bootstrap of a device and - as necessitated by the implementation option be protected from being modified unintential from data plane configuration.

In addition this provides for security, because the ACP is not reachable from the global routing table. Also, configuration errors from the data plane setup do not affect the ACP.

3.7. Addressing in the ACP

The channels explained above only establish communication between two adjacent neighbours. In order for the communication to happen across multiple hops, the autonomic control plane requires internal network wide valid addresses and routing. Each autonomic node must create a loop back interface with a network wide unique address inside the ACP context mentioned in Section 3.6.

We suggest to create network wide Unique Local Addresses (ULA) in accordance with [RFC4193] with the following algorithm:

- Prefix FC01::/8

- Global ID: a hash of the domain ID; this way all devices in the same domain have the same /48 prefix.

- Subnet ID and interface ID: These can be either derived deterministically from the name of the device, or assigned at registration time of the device.

3.8. Routing in the ACP

Once ULA address are set up all autonomic entities should run a routing protocol within the autonomic control plane context. This routing protocol distributes the ULA created in the previous section for reachability. The use of the autonomic control plane specific context eliminates the probable clash with the global routing table and also secures the ACP from interference from the configuration mismatch or incorrect routing updates.

The establishment of the routing plane and its parameters are automatic and strictly within the confines of the autonomic control plane. Therefore, no manual configuration is required.

All routing updates are automatically secured in transit as the channels of the autonomic control plane are by default secured.

The routing protocol inside the ACP should be light weight and highly scalable to ensure that the ACP does not become a limiting factor in network scalability. We suggest the use of RPL as one such protocol which is light weight and scales well for the control plane traffic.

4. Self-Healing Properties

The ACP is self-healing:

- New neighbors will automatically join the ACP after successful validation and will become reachable using their unique ULA address across the ACP.

- When any changes happen in the topology, the routing protocol used in the ACP will automatically adapt to the changes and will continue to provide reachability to all devices.

- If an existing device gets revoked, it will automatically be denied access to the ACP as its domain certificate will be validated against a Certificate Revocation List during authentication.

5. Self-Protection Properties

As explained in Section 3, the ACP is based on channels being built between devices which have been previously been authenticated based on their domain certificates. The channels themselves are protected using standard encryption technologies like IPsec which provide additional authentication during channel establishment, data integrity and data confidentiality protection of data inside the ACP and in addition, provide replay protection.

An attacker will therefore not be able to join the ACP unless having a valid domain certificate, also packet injection and sniffing traffic will not be possible due to the security provided by the encryption protocol.

The remaining attack vector would be to attack the underlying AN protocols themselves, either via directed attacks or by denial-of-service attacks. However, as the ACP is built using link-local IPv6 address, remote attacks are impossible. The ULA addresses are only reachable inside the ACP context, therefore unreachable from the data plane. Also, the ACP protocols should be implemented to be attack resistant and not consume unnecessary resources even while under attack.

6. Use Cases for the ACP

The ACP automatically enables a number of use cases which provide immediate benefits:

o Secure bootstrap of new devices without requiring any configuration. As explained in Section 3, a new device will automatically be bootstrapped in a secure fashion and be deployed with a domain certificate. This will happen without any configuration, allowing a new device to be shipped directly to the end-user location without the need for any pre-provisioning.

o Virtual-out-of-band (VooB) control plane which provides connectivity to all devices regardless of their configuration or global routing table. This makes it possible to manage devices without having to configure data plane services or to deploy a separate management network. It also simplifies management applications, because changes done by the applications cannot affect reachability of the devices.

7. The Administrator View

An ACP is self-forming, self-managing and self-protecting, therefore has minimal dependencies on the administrator of the network. Specifically, it cannot be configured, there is therefore no scope for configuration errors on the ACP itself. The administrator may have the option to enable or disable the entire approach, but detailed configuration is not possible. This means that the ACP must not be reflected in the running configuration of devices, except a possible on/off switch.

While configuration is not possible, an administrator must have full visibility of the ACP and all its parameters, to be able to do trouble-shooting. Therefore, an ACP must support all show and debug options, as for any other network function. Specifically, a network management system or controller must be able to discover the ACP, and monitor its health. This visibility of ACP operations must clearly be separated from visibility of data plane so automated systems will never have to deal with ACP aspect unless they explicitly desire to do so.

Since an ACP is self-protecting, a device not supporting the ACP, or without a valid domain certificate cannot connect to it. This means that by default a traditional controller or network management system cannot connect to an ACP. To make this possible for systems not supporting the ACP natively, the connection to the ACP must be manually established, through configuration. [EDNOTE: More details to be provided in a later version of this document.] Long term NMS

systems might become autonomic devices with domain certificates, and then automatically join the ACP.

8. Security Considerations

An ACP is self-protecting and there is no need to apply configuration to make it secure. Its security therefore does not depend on configuration.

However, the security of the ACP depends on a number of other factors:

- The usage of domain certificates depends on a valid supporting PKI infrastructure. If the chain of trust of this PKI infrastructure is compromised, the security of the ACP is also compromised. This is typically under the control of the network administrator.

- Security can be compromised by implementation errors (bugs), as in all products.

Fundamentally, security depends on correct operation, implementation and architecture. Autonomic approaches such as the ACP largely eliminate the dependency on correct operation; implementation and architectural mistakes are still possible, as in all networking technologies.

9. IANA Considerations

This document requests no action by IANA.

10. Acknowledgements

This work originated from an Autonomic Networking project at Cisco Systems, which started in early 2010. Many people contributed to this project and the idea of the Autonomic Control Plane, amongst which (in alphabetical order): Ignas Bagdonas, Parag Bhide, Alex Clemm, Toerless Eckert, Yves Hertoghs, Bruno Klauser, Max Pritikin, Ravi Kumar Vadapalli.

11. Change log [RFC Editor: Please remove]

00: Initial version.

12. References

[I-D.irtf-nmrg-an-gap-analysis]
Behringer, M., Carpenter, B., and S. Jiang, "Gap Analysis for Autonomic Networking", draft-irtf-nmrg-an-gap-analysis-00 (work in progress), April 2014.

[I-D.irtf-nmrg-autonomic-network-definitions]
Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking - Definitions and Design Goals", draft-irtf-nmrg-autonomic-network-definitions-00 (work in progress), December 2013.

[I-D.pritikin-bootstrapping-keyinfrastructures]
Pritikin, M., Behringer, M., and S. Bjarnason, "Bootstrapping Key Infrastructures", draft-pritikin-bootstrapping-keyinfrastructures-00 (work in progress), January 2014.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

Authors' Addresses

Michael H. Behringer
Cisco

Email: mbehring@cisco.com

Steinthor Bjarnason
Cisco

Email: sbjarnas@cisco.com

Balaji BL
Cisco

Email: blbalaji@cisco.com

Toerless Eckert
Cisco

Email: eckert@cisco.com