Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 19, 2014

M. Behringer
M. Pritikin
S. Bjarnason
Cisco

January 15, 2014

Making The Internet Secure By Default
draft-behringer-default-secure-00

Abstract

Pervasive monitoring on the Internet is enabled by the lack of general, fundamental security. In his presentation at the 88th IETF Bruce Schneier called for ubiquitous use of security technologies to make pervasive monitoring too expensive and thus impractical. However, today security is too operationally expensive, and thus only used where strictly required.

In this position paper we argue that all network transactions can be secure by default, with minimal or no operator involvement. This requires an autonomic approach where all devices in a domain enrol automatically in a trust domain. Once they share a common trust anchor they can secure communications between themselves, following a domain policy which is by default secure.

The focus of this proposal is the network itself, with all protocols between network elements, including control plane protocols (e.g., routing protocols) and management plane protocols (e.g., SSH, netconf, etc). The proposal is evolutionary and allows a smooth migration from today's Internet technology, device by device.

Status of This Memo

Table of Contents

1. Problem Statement

The reason that security mechanisms and protocols are not used today is not that they are not available: There are plenty of secure protocols available. Reasons why they are not used are typically:

- Lack of operational experience with the secure protocols.
- Complexity of the set-up of security mechanisms, including key management.
- Ongoing operational cost for key management, trouble-shooting, etc.

All of these reasons are operational reasons. The net result is that enabling security on the network today is expensive, and therefore only done when the expense can be justified. Many network operators have tight budgets and will therefore avoid unnecessary operational

expenses. Many security measures which are in theory available don't get deployed because of the operational cost involved.

2. Autonomic Security

The fundamental idea in this paper is to make security a default behavior of a given network, and by extension the Internet as a whole. This is enabled by autonomic concepts [I-D.behringer-autonomic-network-framework], [I-D.irtf-nmrg-autonomic-network-definitions]: Network devices negotiate all required information, including security policies and key materials, without the need of administrator intervention. In an autonomic network, the default behavior is to secure every protocol. A policy can define deviations from this rule if required. The approach is backwards compatible, and allows for a smooth, device by device upgrade. If in a negotiation it turns out that a peer element is not autonomic, it simply keeps using existing mechanisms. Policy controls this process to avoid downgrade attacks.

Negotiation of security parameters and keying material is available today in many protocols, for example the IPsec protocol suite, or TLS. However all such protocols act independently, and need to be set up and maintained independently. In a network that runs internally iBGP, OSPF, BFD and PIM for example, each protocol security needs to be set up independently. Here we propose a generic approach, where a node-level domain identity can be used to negotiate parameters for any other protocol. We also demonstrate how this node-level domain identity can be bootstrapped automatically.

2.1. Overview Of The Proposed Solution

The proposed solution requires some generic autonomic behavior on nodes. "Autonomic" is generally described as "self-management", which contains other "self-*" properties [Kephart]. Fundamentally, an autonomic node acts slightly differently from a traditional node:

- It discovers other nodes, and their identities.
- It negotiates capabilities and parameters with other nodes or groups of nodes.
- When another node belongs to the same domain (or a trusted one), and has the required capabilities, keys are negotiated automatically, and the required security protocol or mechanism is enabled automatically.
- A high-level policy, called "intent" can be used to define the default behavior of the network.

As mentioned above, all those properties exist today inside specific protocols, but each protocol has to be configured manually, and the key material has to be managed independently for each. Autonomic Networking makes those properties available on a generic basis, to all processes running on a node. This way, the key management problem is solved a single time for all protocols, in an automatic fashion. Security mechanisms along with their parameters and key materials can be bootstrapped completely automatically.

With an autonomic approach on network elements, security can therefore be made "automatic", with little to no administrative intervention. This allows networks to run in a default secure mode for all protocols, at little or no additional operational expenditure.

2.2. Zero-Touch Bootstrapping Domain Certificates

The underlying security principle for Autonomic Networking is the concept of a domain identity. A network operator manages and controls all devices of a domain. A device belonging to a domain by default trusts only devices belonging to the same domain. Each device in the domain receives a domain certificate, which can be cryptographically validated by other devices.

The distribution of domain certificates can be completely automated, in a secure way. The document "Bootstrapping Key Infrastructures" [I-D.pritikin-bootstrapping-keyinfrastructures] describes how domain certificates can be securely bootstrapped in a zero-touch way, and is the technical foundation behind this position paper. A practical example on how a user in a homenet can use this mechanism is outlined in the document "Bootstrapping Trust on a Homenet" [I-D.behringer-homenet-trust-bootstrap].

This allows the network operator to add devices to the domain based on existing device credentials (e.g. IEEE 802.1AR vendor certificate) or other any other criteria (location, time). In addition, the new device has the possibility to validate the authenticity of the domain that it is being invited to, by requiring the invite to be signed by the manufacturer.

After a device has joined the domain, it can now communicate with any other device in the domain and use the domain certificate as trust anchor for any subsequent operations.

2.3. Bootstrapping Key Material

When an autonomic device wants to communicate or exchange information with another autonomic device, it establishes and validates the identity of the peer. A device belonging to the same domain is trusted by default; a policy can restrict for which operations the trust is valid. When the identity has been confirmed, the devices negotiate key material and parameters for all subsequent protocols.

For example, the domain certificate introduced here helps in case of routing protocols to negotiate a common key to be used for routing protocol authentication as explained in [I-D.polk-saag-rtg-auth-keytable] and draft-ietf-karp-ops-model [I-D.ietf-karp-ops-model]. The actual key negotiation is outside scope of this document.

2.4. Backward Compatibility

The approach described in this paper is evolutionary and allows for a smooth migration. The enabling feature for this is the intrinsic negotiation capability of an autonomic network. As part of the BGP example above, a node will initially negotiate capabilities with the peer(s). If the peer does not respond to the negotiation, we assume it does not understand autonomic behavior, and fall back to traditional, configured methods. If it does respond to negotiation, the required parameters are negotiated and enabled automatically. This way, a network can be migrated node by node. To detect and possibly avoid downgrade attacks autonomic nodes log and notify such events; once a negotiation with a node has happened, no downgrade is allowed any more.

2.5. Policy

With the approach described here, nodes become intelligent enough to negotiate the optimum security between themselves. This way, security can be bootstrapped without any involvement of an admin or configuration tool. However, different networks have different requirements. For example, link encryption could be established automatically using this approach. However, what should a node do when for some reason the negotiation of crypto parameters fails? Should it allow the connection unsecured, or should it drop it? Also, in some environments, integrity protection for network protocols is desired, in others all protocols should be encrypted.

These are policy questions. One network may want to only do integrity protection, another one also encrypt; on might pursue a "fail-open", another one a "fail-close" policy: When the security operation fails, the communication is allowed in clear, or not

allowed. In an autonomic network there is a high level policy called "intent" which defines such default behavior. This policy may also define how to handle connections to other domains.

2.6. Inter-Domain Security

The security in an autonomic network is primarily based on the domain certificates. This way, devices within a domain can authenticate each other, using the same domain trust anchor. Sub-domains can be used to segment a larger network and introduce different policies.

The same concept can be applied toward other domains, but using a common trust anchor, or by cross-signing. A service provider for example could validate the trust anchors of his peer providers once, so that his nodes can also authenticate the nodes of the peers. While such SP-to-SP approaches have failed in the past to reach any significant deployment, here this interaction only has to happen on the root certificate level once for the duration of the certificate lifetime. This may be an acceptable burden. Common trust anchors between all SPs are technically possible, but such approaches have failed in the past, and are not further considered here.

Once nodes of another domain are authenticated, the policy in the domain can define their authorization levels. This allows us to define for example a policy like "trust all nodes from SPx for secure eBGP connections". At first, the main BGP configuration can be manual, with only the security being negotiated; at later stages additional aspects of BGP and other protocols can be automated as well.

3. Future Work

This paper illustrates the first step only in a long journey. The main focus at this moment are network nodes, however, we believe that the concepts of autonomic networking can be applied to end systems and applications as well. This is for future study.

At the moment, trust within the domain is coarse grained, and only allows for sub-domains. More fine-grained control inside a domain, as well as inter-domain trust management requires more work.

To enable self-management, many components are required in a standardized way, such as negotiation capabilities, a message bus, and discovery mechanisms. Discussions have started in the IRTF on definitions and goals of autonomic networking. Many details still need to be worked out, although it is expected that many available components can be re-used in the framework of Autonomic Networking.

4. Summary

Today, networks depend on detailed configuration, either from humans, network management systems, or controllers. Configuration of security is always explicit, requiring serious efforts primarily in the operational management. The fundamental weakness today is that there is no universal, secure node identity, so that all components today have to bootstrap their own security model.

We believe that a node-level secure domain identity in the form of a certificate, with a zero-touch yet secure bootstrap mechanism is the fundamental cornerstone to making security a default on networks. This approach is incremental and allows network elements to automatically negotiate security with their peers. The model can be extended inter-domain, allowing to also secure connections over multiple domains. Following this model, networks could be secure by default.

5. Informative References

[I-D.behringer-autonomic-network-framework]
Behringer, M., Pritikin, M., Bjarnason, S., and A. Clemm, "A Framework for Autonomic Networking", draft-behringer-autonomic-network-framework-01 (work in progress), October 2013.

[I-D.behringer-homenet-trust-bootstrap]
Behringer, M., Pritikin, M., and S. Bjarnason, "Bootstrapping Trust on a Homenet", draft-behringer-homenet-trust-bootstrap-01 (work in progress), October 2013.

[I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", draft-ietf-homenet-arch-11 (work in progress), October 2013.

[I-D.ietf-karp-ops-model]
Hartman, S. and D. Zhang, "Operations Model for Router Keying", draft-ietf-karp-ops-model-10 (work in progress), January 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.polk-saag-rtg-auth-keytable]
Polk, T. and R. Housley, "Routing Authentication Using A Database of Long-Lived Cryptographic Keys", draft-polk-saag-rtg-auth-keytable-05 (work in progress), November 2010.

[I-D.pritikin-bootstrapping-keyinfrastructures]
Behringer, M., Pritikin, M., and S. Bjarnason, "Bootstrapping Key Infrastructures", January 2014.

[Kephart] Kephart, J. and D. Chess, "The Vision of Autonomic Computing", IEEE Computer vol. 36, no. 1, pp. 41-50, January 2003.

Authors' Addresses

Michael H. Behringer
Cisco

Email: mbehring@cisco.com

Max Pritikin
Cisco

Email: pritikin@cisco.com

Steinthor Bjarnason
Cisco

Email: sbjarnas@cisco.com