Secure Name Configuration and Prefix Registration

Marc Mosko, Glenn Scott, Nacho Solis, Christopher A. Wood
Palo Alto Research Center, Palo Alto, CA 94304
{mmosko, gscott, isolis, cwood}@parc.com

Categories and Subject Descriptors: C.2 [Computer-Communication Networks]: Network Protocols

Categories and Subject Descriptors: C.2.4 [Computer-Communication Networks]: Distributed Systems – *distributed applications*

Keywords: Content-centric networking; name and prefix registration

1. INTRODUCTION

Content Centric Networking (CCN) is a networking architecture that emerged from the pitfalls of today's IP-based Internet design. Contrary to host-based traffic in the modern Internet, CCN traffic is driven by explicit requests for named content. One of the main features of this name-based content retrieval strategy is that it effectively decouples content from its original producer, thereby enabling more natural content distribution. At the same time CCN enables routers to opportunistically cache content *in the network*. Cached content can then be returned in response to future interests for the same content. This avoids the need to forward every request (interest) from consumers all the way to producers, thus lowering network congestion and reducing content retrieval latency.

CCN routers are responsible for forwarding requests (called interests) for content emitted from consumers to an authoritative source (producer) capable of generating or providing the desired content (object). Generally, forwarding interest messages toward a producer is done using only their names. Routers maintain a data structure called a Forwarding Interest Base (FIB) which maps *name prefixes* to a set of interfaces to which interests should be forwarded. Like IP routing tables, FIBs are populated either manually or using a routing algorithm.

In existing routing algorithms (see [1] and references therein), producers advertise prefixes of content they are willing to serve under, and these routes are propagated throughout the network to enable routing. For example, Google might choose to serve content under the `/google/` namespace prefix. Routing algorithms serve to install routes to `/google/` in router FIBs so that interests for any content with this name prefix are forwarded to the Google "producer." Although this arrangement is functionally correct, i.e., it will enable consumer interests to be routed to the Google producer, there is an issue of whether or not said producer is *permitted* to publish or serve content under its desired namespace. Without any form of trusted authentication and authorization, *anyone* is free to advertise content under *any* prefix. Moreover, there is the problem of registering and configuring this name prefix *in the transport stack* so that interests will be routed up towards the producer application.

We address these issues with the CCN dynamic name configuration and local prefix registration service. Together, these elements provide an application with the means to (a) securely register namespaces under which content will be served, (b) obtain the authentication token necessary to install a prefix locally so that interests can be returned to a producer application, and, (c) advertise a certain prefix to the rest of the network so that consumers may retrieve content from the target producer. Our design provides a complete end-to-end workflow that enables producers to begin serving content under a namespace they are authorized to use, thereby addressing one of the larger problem in namespace management.

2. THE CCN TRANSPORT STACK

All applications interface with the CCN transport stack using the Portal API – a lightweight interface that provides simple interest and content exchanges. The CCN transport stack is a set of components, each of which is focused on a specific task. It adheres to the chain-of-command pattern: each component processes a message and then forwards it to the next component in the chain. Each component has an outbound queue to move messages from the application toward the network, as well as an inbound queue to move messages from the network toward the application. Each transport stack requires the following two components: the API adapter and the forwarder adapter. Applications connect to the API adapter to communicate to the stack, and the stack uses the forwarder adapter to connect to the local forwarder. The forwarder is the component that contains, updates, and uses the FIB, PIT, and content store (CS) when processing inbound and outbound messages. Messages are pipelined through the transport stack components from the upper-level API to the **forwarder** by means of these adapters. Messages can be scoped to the local (i.e., endhost) machine with a special `localhost` prefix, thereby enabling IPC among applications running on the same endhost.

ICN'15, Sept.30–Oct. 2, 2015, San Francisco, CA, USA.
ACM 978-1-4503-3855-4/15/09.
DOI: http://dx.doi.org/10.1145/2810156.2812605 .

3. NAME CONFIGURATION AND PREFIX REGISTRATION

To address the problem of producer name registraticfxon and advertisement, we present the design of a Kerberos-like [2] name configuration and prefix registration service. Our design presupposes the existence of a Dynamic Name Configuration Service (DNCS), which is a centralized authority responsible for managing namespaces. Part of its managerial responsibilities include issuing *permission tokens* that are used by endhost transport stacks to install and advertise namespace prefixes. Each endhost has a single Dynamic Name Configuration Agent (DNCA) daemon service which is responsible for all communication with the DNCS.[1] This interaction is done on behalf of the producer to obtain the aforementioned permission tokens necessary to "use" prefixes. Specifically, when a producer application wishes to "use" a prefix N, it issues a request (in the form of a localhost-scoped interest) to the local DNCA that contains N, the producer's identity (e.g., a certificate), and a signature computed on both. The DNCA then provides this request to the DNCS in the form of an interest, which forces the DNCS to validate the request and authorize the producer to use N – returning a permission token as an acknowledgment – or deny the request and return a simple NACK. A permission token only needs to bind the namespace N to the producer's identity, e.g., it can consist of a simple DNCS-generated signature over both.

In addition to the local DNCA, endhosts also have a Prefix Registration Service (PRS) and Routing Protocol service (RP). The former is responsible for instructing the single local forwarder (FWD) to install, remove, or update FIB entries to suit the needs of a producer application. After a producer application receives a permission token from the DNCS, through the DNCA, the producer will ask the PRS for permission to perform a *specific action* with the local FIB. For example, the PRS might permit the producer to only install the namespace N in the FIB. The producer provides the DNCS permission token in the PRS request, and the PRS only authorizes the request if a valid permission token is provided. The short-term and single-use PRS token generated in response to a producer's request binds their identity to the namespace and desired forwarder action. Finally, the producer can issue this single use token to the forwarder to complete the action. Security of this procedure is rooted in the validity of each signature computed and verified, as well as the trustworthiness of the DNCS (the trust anchor). Note that installing an entry in the FIB *does not* actually advertise N to the rest of the network. To do this, the producer must request the local RP to advertise N, providing the DNCS permission token in the process. A complete overview of the end-to-end workflow is shown in Figure 1. We use the notation LT and ET to refer to the DNCS long-term permission token and PRS short-term (ephemeral) permission token, respectively.

4. ELEMENTS OF TRUST

Trust for secure prefix registration is anchored in the DNCS. The endhost PRS agent and routing process trust the DNCS (through the DNCA), and the local forwarder (FWD) trusts

[1]Note that the DNCA is a standard CCN application that runs on top of the transport stack.

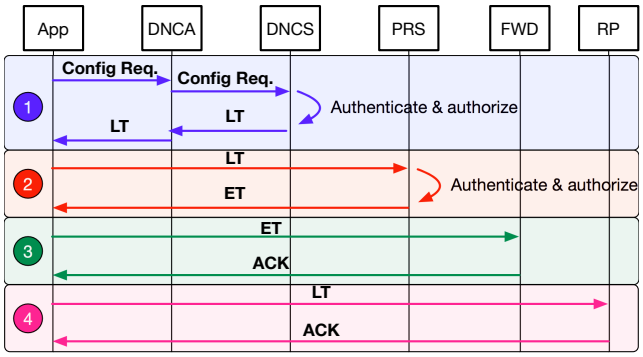

Figure 1: Kerberos-like sequence diagram showing the messages that need to be sent to install and advertise a prefix.

the PRS. Namespaces will not be advertised by the routing protocol unless given explicit permission by the DNCS (see Step 4 of Figure 1). Thus, malicious producers may only advertise namespaces under which they are not authorized to serve if they can subvert the routing process verification check or forge a long-term permission token from the DNCS.

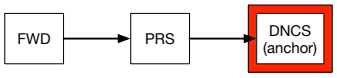

Figure 2: PRS chain of trust anchored at the DNCS.

5. FUTURE WORK

We presented the preliminary design and a complete end-to-end workflow that enables CCN producers to begin serving content under a namespace they are authorized to use. Our solution provides an application with the means to (a) securely register namespaces under which content will be served, (b) obtain the authentication token necessary to install a prefix locally, and (c) advertise a certain prefix to the rest of the network.

Future work will entail addressing issues the design of the DNCS which, as presented, is a centralized service. A more federated and hierarchical approach akin to today's DNS is needed to prevent it from becoming a single point of trust and failure. Moreover, trust delegation between parent and children DNCS instances needs to be handled securely. We will also explore issues of trust management in the routing processes and protocols, as well as ways to avoid subversion and collusion by producers.

6. REFERENCES

[1] JJ Garcia-Luna-Aceves. Name-based content routing in information centric networks using distance information. In *Proceedings of the 1st international conference on Information-centric networking*, pages 7–16. ACM, 2014.

[2] B Clifford Neuman and Theodore Ts' O. Kerberos: An authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, 1994.