Content Delivery Network Interconnection (CDNI) Request Routing: CDNI Footprint and Capabilities Advertisement using ALTO

draft-alto-cdni-request-routing-alto-06

J. Seedorf, Y. Richard Yang, Kevin Ma, J. Peterson, X. Lin

IETF 104
March 26, 2019
Prague

Outline

- Document is quite stable, but depends on
  - SSE (3.7.3, 4.2.4, 5.7.3, 6.2.4); all are examples
  - Unified Properties (6, 8)
- Changes addressing reviews
  - https://www.ietf.org/rfcdiff?url1=draft-ietf-alto-cdni-request-routing-alto-03&url2=draft-ietf-alto-cdni-request-routing-alto-06
- Issues raised by reviews but not addressed yet

Types of Changes: Clarification

- Throughout the document
- A simple example

```
3.7.3.  Incremental Updates Example

   A benefit of using ALTO to provide CDNI FCI maps is that such maps
   can be updated using ALTO incremental updates.  Below is an example
   that also shows a benefit of using a JSON merge patch to encode a big
   update and using a JSON patch to encode a small update.

     POST /updates/cdnifcimaps HTTP/1.1
     Host: alto.example.com
     Accept: text/event-stream,application/alto-error+json
     Content-Type: application/alto-updatestreamparams+json
     Content-Length: ###

     { "add": {
         "my-cdnifci-stream": {
             "resource-id": "my-default-cdnifci-map"
         }
     }

     HTTP/1.1 200 OK
     Connection: keep-alive
     Content-Type: text/event-stream
```

```
3.7.3.  Incremental Updates Example

   A benefit of using ALTO to provide CDNI FCI maps is that such maps
   can be updated using ALTO incremental updates.  Below is an example
   that also shows the benefit of having both JSON merge patch and JSON
   patch to encode updates.

   At first, an ALTO client requests updates for "my-default-cdnifci-
   map", and the ALTO server returns the "control-uri" followed by the
   full CDNI FCI map.  Then when there is a change in the delivery-
   protocols in that `http/2` is removed (from http/1.1 and http/2 to
   only http/1.1) due to maintenance of the http/2 clusters, the ALTO
   server uses JSON merge patch to encode the change and pushes the
   change to the ALTO client.  Later on, the ALTO server notifies the
   ALTO client that "ipv4:192.0.2.0/24" is added into the footprint for
   delivery-protocol http/1.1 by sending the change encoded by JSON
   patch to the ALTO client.

     POST /updates/cdnifcimaps HTTP/1.1
     Host: alto.example.com
     Accept: text/event-stream,application/alto-error+json
     Content-Type: application/alto-updatestreamparams+json
     Content-Length: ###

     { "add": {
         "my-cdnifci-stream": {
             "resource-id": "my-default-cdnifci-map"
         }
     }

     HTTP/1.1 200 OK
     Connection: keep-alive
     Content-Type: text/event-stream
```

Update: New Format of Error Handling Definition

draft-ietf-alto-cdni-request-routing-alto.txt

```
5.6.  Response

   The format is the same as an unfiltered CDNI FCI map.  See
   Section 3.6 for the format.

   The returned CDNI FCI map MUST contain only BaseAdvertisementObject
   objects whose CDNI capability object is the superset of one of CDNI
   capability object in "cdni-fci-capabilities".  Specifically, that a
   CDNI capability object A is the superset of another CDNI capability
   object B means that these two CDNI capability objects have the same
   capability type and mandatory properties in capability value of A
   MUST include mandatory properties in capability value of B
   semantically.  For example, if a CDNI FCI capability in "cdni-fci-
   capabilities" is Delivery Protocol capability object with "http/1.1"
   in its field "delivery-protocols" and the original full CDNI FCI map
   has two CDNI FCI objects whose capabilities are Delivery Protocol
   capability objects with ["http/1.1"] and ["http/1.1", "https/1.1"] in
   their field "delivery-protocols" respectively, both of these two CDNI
   FCI objects MUST be returned.  If the input parameters contain a CDNI
   capability object that is not currently defined, the ALTO server MUST
   behave as if the CDNI capability object did not appear in the input
   parameters.
```

```
5.6.  Response

   The response MUST indicate an error, using ALTO protocol error
   handling specified in Section 8.5 of the ALTO protocol [RFC7285], if
   the request is invalid.

   Specifically, a filtered CDNI FCI map request is invalid if:

   o  the value of "capability-type" is null;

   o  the value of "capability-value" is null;
   o  the value of "capability-value" is inconsistent with "capability-
      type".

   When a request is invalid, the ALTO server MUST return an
   E_INVALID_FIELD_VALUE" error defined in Section 8.5.2 of [RFC7285],
   and the "value" field of the error message SHOULD indicate this CDNI
   FCI capability.

   The ALTO server returns a filtered CDNI FCI map for a valid request.
   The format of a filtered CDNI FCI map is the same as an unfiltered
   CDNI FCI map.  See Section 3.6 for the format.

   The returned CDNI FCI map MUST contain only BaseAdvertisementObject
   objects whose CDNI capability object is the superset of one of CDNI
   capability object in "cdni-fci-capabilities".  Specifically, that a
   CDNI capability object A is the superset of another CDNI capability
   object B means that these two CDNI capability objects have the same
   capability type and mandatory properties in capability value of A
   MUST include mandatory properties in capability value of B
   semantically.  See Section 5.7.2 for a concrete example.
```

Update: Security Considerations

draft-ietf-alto-cdni-request-routing-alto.txt

9. Security Considerations

...ion is the proper authentication of advertisement information provided by a downstream CDN. The ALTO protocol provides a specification for a signature of ALTO information (see Section 15 of [RFC7285]. ALTO thus provides a proper mechanism for protecting the integrity of FCI information.

More Security Considerations will be discussed in a future version of this document.

9. Security Considerations

As an extension of the base ALTO protocol [RFC7285], this document fits into the architecture of the base protocol, and hence the Security Considerations (Section 15) of the base protocol fully apply when this extension is provided by an ALTO server.

In the context of CDNI FCI, additional security considerations should be included as follows.

For authenticity and integrity of ALTO information, an attacker may disguise itself as an ALTO server for a dCDN, and provide false capabilities and footprints to a uCDN using the CDNI FCI map. Such false information may lead a uCDN to (1) select an incorrect dCDN to serve user requests or (2) skip uCDNs in good conditions.

For potential undesirable guidance from authenticated ALTO information, dCDNs can provide a uCDN with limited capabilities and smaller footprint coverage so that dCDNs can avoid transferring traffic for a uCDN which they should have to transfer.

For confidentiality and privacy of ALTO information, footprint properties integrated with ALTO unified property may expose network location identifiers (e.g., IP addresses or fine-grained PIDs).

For availability of ALTO services, an attacker may get the potential huge full CDNI FCI maps from an ALTO server in a dCDN continuously to run out of bandwidth resources of that ALTO server or may query filtered CDNI FCI services with complex capabilities to run out of computation resources of an ALTO server.

Protection strategies described in RFC 7285 can solve problems mentioned above well. However, the isolation of full/filtered CDNI FCI maps should also be considered.

If a dCDN signs agreements with multiple uCDNs, it must isolate full/filtered CDNI FCI maps for different uCDNs in that uCDNs will not redirect requests which should not have to served by this dCDN to this dCDN and it may not disclose extra information to uCDNs.

To avoid this risk, a dCDN may consider generating URIs of different full/filtered CDNI FCI maps by hashing its company ID, a uCDN's company ID as well as their agreements. And it needs to avoid expoing all full/filtered CDNI FCI maps resources in one of its IRDs.

Remaining Issue: Map -> Table

- Issue
  - From RFC7285: “This document also defines dictionary maps (or maps for short) from strings to JSON values. For example, the definition below defines a Type3 object as a map. Type1 must be defined as string, and Type2 can be defined as any type.
    object-map { Type1 -> Type2; } Type3;”
  - CDNI: FCI is not a key-value store format
- Proposed solution
  - Rename Map to Table or unnecessary?

Remaining Issue: More Generic Query

- Current design allows query only by footprint or only by capability
- Suggested query: by both footprint and capability
- Possible next step:
  - Do not address
  - Revise this document
  - Extend Unified Property to allow joint entity (footprint) and property (capability) query

Summary: Plan

- Take a pass to address any additional WG comments, if raised: in 2 weeks after IETF

- Wait for SSE and UP updates