A Framework for SIP User Agent Configuration

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This document defines the application of a set of protocols for configuring a SIP user agent. The SIP user agent must discover how and from where to retrieve its initial configuration and be notified of changes and updates which impact its configuration. The objective is to define a means for automatically configuring a user agent such that it can be functional without user or administrative intervention. The framework for discovery, delivery, notification and updates of user agent configuration is defined here. This framework is also intended to ease ongoing administration, configuration and upgrading of large scale deployments of SIP user agents. The contents and format of the configuration data to be defined is outside the scope of this document.

Table of Contents
Overview

This document defines a framework which allows SIP user agents (UA) to automatically:
- discover a configuration server (Discovery)
- enroll with the configuration server (Enrollment)
- retrieve configuration data (Configuration Retrieval)
- receive notification of configuration changes (Change Notification)
- upload configuration data changes back up to the server (Configuration Upload)

The content and format of the data is not defined in this document. It is to be defined in configuration data profile(s) in other document(s). The goal of this framework is to satisfy the requirements defined in [10] and [11] excluding the requirements which pertain to configuration data profile content and format.

Discovery is the process by which a UA SHOULD find the address and port at which it SHOULD enroll with the configuration server. As there is no single discovery mechanism which will work in all network environments, a number of discovery mechanisms are defined with a prescribed order in which the UA SHOULD try them until one succeeds.

Enrollment is the process by which a UA SHOULD make itself known to the configuration server. In enrolling the UA MUST provide identity information, a named list of requested configuration data profiles and supported protocols for configuration retrieval. It SHOULD also SUBSCRIBE to a mechanism for notification of configuration changes. As a result of enrollment the UA receives a URL for each of the configuration data profiles that the configuration server is able to provide.

Configuration Retrieval is the process of retrieving the content for each of the configuration data profiles the UA requested.

Change Notification is the process by which the configuration server notifies the UA that the content of one or more of the configuration data profiles has changed. Subsequently the UA SHOULD retrieve the data profile from the specified URL upon receipt of the change notification.

Configuration Upload is the process by which a UA or other entity pushes a change to a configuration data profile back up to the configuration server.

Today all SIP UA vendors use proprietary means of delivering configuration to the UA. This configuration framework is intended to enable a first phase migration to a standard means of configuring SIP user agents. It is expected that UA vendors should be able to use this configuration framework as a means of delivering their existing proprietary configuration data profiles (i.e. using their existing proprietary binary or text formats). This in itself is a tremendous advantage in that a SIP environment can use a single configuration server to deliver configuration data to UAs from multiple vendors. Follow-on standardization activities can: 1) define a standard format (e.g. XML or name-value pairs [8]) and 2) specify the content (i.e. name the configuration parameters) of the configuration data profiles.
Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1].

The syntax and semantics used here extend those defined in SIP (RFC 2543) [6]. SIP is described in an augmented Backus-Naur form (ABNF). See [6, section C] for an overview of ABNF.

3 Discovery

The first time a UA is plugged in it does not know the address or port at which to enroll with the local configuration server. It must discover this address and port. A UA SHOULD support all of the listed discovery mechanisms. It MUST support at least one of them. Once the UA has discovered the address and port and has successfully enrolled with the configuration server, the UA SHOULD cache the address and port to avoid the need to re-discover the configuration server. However if enrollment, configuration retrieval or configuration upload fails at any time, the UA SHOULD apply the discovery and enrollment process again. This provides a means for configuration server fail over and load balancing.

The UA SHOULD use the following mechanisms to discover the host address and port at which it SHOULD enroll with the configuration server. Each mechanism should be tried in the following order until an address and port is provided which results in successful enrollment (i.e. the server responds with a successful 2xx class response):

- DHCP site-specific option [1]
- DNS SRV
- DNS A record
- Multicast
- Manual provisioning

The rationale for this order follows. Assuming that most UAs are going to use DHCP for IP configuration anyway, using a DHCP option is the least costly in terms of lookup time (i.e. no additional messages are required). Hence DHCP is first. DNS SRV allows the more flexibility than DNS A records. Hence DNS SRV is tried before DNS A records. Multicast is used last of the automated discovery mechanisms as it is the most restricted in terms of network environments that support it. Multicast is included, even though the applicable environments are restricted, as it is the only mechanism that can be used without the support of the local network administrator (The phone administrator and the network administrator are often different people and perhaps in different departments.)

The UA implementer MAY provide the user or administrator with the means to change the order in which these mechanisms are tried. However by default without user interaction it SHOULD use the order listed above.

3.1 DHCP Option

It is likely that most UAs in an environment of any significant
number will use DHCP for IP configuration. DHCP becomes a convenient means to discover the configuration server. In the same DHCP request for basic IP configuration, the UA can add the site-specific option [TBD] [1] to the options field. This indicates a request for the configuration server address and port. If the configuration server address and port is not returned in the DHCP response or the server does not respond with a successful 2xx class response, the next discovery mechanism is attempted.

[site-specific DHCP option (i.e. > 128)?]
or
[DHCP Option for SIP Servers?]

3.2 DNS SRV

Using the service identifier ÂsipuaconfigÂ the DNS SRV records [5] are requested for the local domain for the protocols (i.e. UDP and/or TCP) that the UA supports. The UA tries to enroll using the search order as prescribed in RFC 2782 [5]. If none of the servers respond with a successful 2xx class response (or none are returned in the SRV records) the next discovery mechanism is attempted.

3.3 DNS

The UA SHALL try a DNS A record lookup on the host name ÂsipuaconfigÂ. If the server does not respond with a successful 2xx class response, the next discovery mechanism is attempted.

3.4 Multicast

The enrollment request is sent to the multicast address for SIP registration [6] "sip.mcast.net" (224.0.1.75). If a server does not respond with a successful 2xx class response to the enrollment request, the next discovery mechanism is attempted.

3.5 Manually Provisioned

The UA SHOULD let the user (or administrator) know if the automatic discovery has failed and allow the user or administrator to manually (or perhaps using some other out of band means e.g. beam, smart card, etc.) enter the configuration server address and port to be used for enrollment.

Petrie       Informational - Expires September 2001               5
A Framework for SIP               March 2001
User Agent Configuration

4 Enrollment and Change Notification

The enrollment and configuration change notification are paired together and provided via the SIP SUBSCRIBE/NOTIFY framework [7]. This document defines the profile on top of the SUBSCRIBE/NOTIFY framework [7] for this purpose.

UA enrollment with the configuration server is accomplished via the SUBSCRIBE request. A UA MUST enroll with the configuration server prior to retrieving configuration data profiles. As part of the enrollment the UA MUST identify itself, its configuration retrieval protocol capabilities and configuration data profile requirements.
The configuration server may use this information to decide how to allocate resources (e.g. load balancing) to support the UA for its specific configuration retrieval needs. The configuration server may also use the UA enrollment event as the trigger to generate a new set of configuration data for the specific UA (e.g. based upon provisioned defaults and configuration profile context knowledge for the environment). This would allow the configuration server to provide configuration data for a new UA without previously provisioning the specific UA on the server.

Configuration Change Notification is communicated to the UA via a NOTIFY request from the configuration server. The NOTIFY request is used by the configuration server to convey the URLs that the UA MUST use to retrieve its requested configuration data profiles. The NOTIFY is used immediately after enrollment. It MAY be subsequently used by the configuration server to identify the list of configuration data profile URLs which have changed (i.e. change notification).

The SUBSCRIBE request for enrollment is sent to the address(es) identified in the discovery process until the first successful 2xx class response is received. As part of the binding of the SUBSCRIBE/NOTIFY framework this document defines a new Event token: Config-Event. The Event header field is mandatory in SUBSCRIBE and NOTIFY requests and MUST contain this token value when used for the purpose of enrollment and configuration change notification.

[Does Config-Event need to be registered with IANA?]

If enrollment fails (i.e. no 2xx response to SUBSCRIBE), the UA SHOULD re-discover the configuration server address and port as described in section 3.

The following new header fields are defined for use in SUBSCRIBE and NOTIFY requests for the purpose of enrollment and configuration change notification:

The keys used the following table:

<table>
<thead>
<tr>
<th>Header</th>
<th>Where</th>
<th>SUBSCRIBE</th>
<th>NOTIFY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config-Allow</td>
<td>R</td>
<td>m</td>
<td>-</td>
</tr>
<tr>
<td>Config-Require</td>
<td>R</td>
<td>m</td>
<td>-</td>
</tr>
<tr>
<td>Config-Expires</td>
<td>R</td>
<td>-</td>
<td>o</td>
</tr>
</tbody>
</table>

4.1 Header Field Definitions

4.1.1 Config-Allow

The Config-Allow header field is used by the UA in the enrollment
request (SUBSCRIBE) to list the protocols that it is capable of using to retrieve configuration data. The configuration server MUST adhere to the protocol capabilities of the UA when providing the list of URLs for the configuration profiles in the NOTIFY request.

Syntax:
```
Config-Allow = "Config-Allow" :: 1#config-protocol
config-protocol = @tftp | @http | @https | token
```

4.1.2 Config-Require

The Config-Require header field contains the names of all of the configuration data profiles that the UA requires. The name(s) of the configuration profiles are to be defined in a future document(s) specifying the content and format of the specific profile.

Syntax:
```
Config-Require = @Config-Require :: 1#config-profile-name
config-profile-name = token
```

4.1.3 Config-Expires

The optional Config-Expires header field defines the lease length of the configuration data. If Config-Expires was present in the last NOTIFY received from the configuration server and the UA has not received a notification from the configuration server within this period of time, the UA SHOULD re-enroll by sending a new SUBSCRIBE message to the configuration server. If the enrollment fails, the UA SHOULD re-discover the configuration server using the mechanisms described in section 3. The configuration server SHOULD send a NOTIFY before the lease expires with the event Config-Event, a renewed lease length in the Config-Expires header field and the complete list of configuration data profile URLs in the request body for the UA's configuration data. The configuration data profile URLs SHOULD have the same sequence numbers if the content has not changed. The sequence numbers MUST be different for profiles whose content has changed. The absence of the Config-Expires header field in the lease renewal indicates an indefinite expiration.

Note: the Config-Expires header field sets a lease that the UA observes to determine when its configuration is stale. This lease is renewed with every NOTIFY message from the configuration server. The Expires header field in the SUBSCRIBE request describes the duration that the configuration server will continue to send change notifications to the UA. This is renewed with every SUBSCRIBE request from the UA.

Syntax:
```
Config-Expires = @Config-Expires :: delta-seconds
```

4.2 SUBSCRIBE
The SUBSCRIBE request is used by the UA to enroll in the configuration domain of the configuration server. It uniquely identifies the UA with vendor, model and serial number information. The UA also MUST specify its capabilities for configuration retrieval as well as the configuration data profiles that it requires. That is the UA MUST include the Config-Allow and Config-Require header fields and each MUST contain at least one token. The configuration server MUST not send an error if it is not able to provide all of the configuration data profiles listed in the SUBSCRIBE request Config-Require header field. The configuration server SHOULD provide the configuration data profile that it is able to or desires (see example at the end of section 4.3) to deliver to the UA. If the configuration server sends a 301 Moved Permanently response to the enrollment SUBSCRIBE, the UA SHOULD cache the URL contained in the response Contact header field in place of the address and port found during discovery for future enrollment.

The configuration server MAY use the enrollment (SUBSCRIBE request) as the stimulus to generate a new instance of a configuration data profile unique to the UA. Alternately the configuration server MAY be provisioned ahead of time to know about new UAs and their specific configuration data content (for example based upon serial number, MAC address).

[Request URI should not contain a user id? The user may not be known yet.]

[What happens when the config server receives multiple SUBSCRIBE requests from the same UA but for different list of profiles. Does the last request supercede all previous ones?]

[What happens when the config server receives multiple SUBSCRIBE requests from the same IP address but for different devices? This might happen if some entity is acting as a proxy for a bunch of other devices. It might also happen if the IP address for a UA gets reused for some other UA (the DHCP lease timeout may be much shorter than the SUBSCRIBE/NOTIFY lease timeouts). The SUBSCRIBE lease SHOULD not exceed the DHCP lease? The UA SHOULD reenroll if its IP address changes?]

4.2.1 Additional From Field Parameters

In the enrollment and configuration change notification messages (i.e. SUBSCRIBE and NOTIFY requests and responses) the SIP-URL [6] MUST not contain userinfo if the default UA user is not known (e.g. first time startup of new UA out of the box).

The following additional From field parameters are defined for the purpose of identifying the UA device:

Vendor Âa token used to identify the UA vendor name

Model Âa token used to identify the UA hardware/software model

Version Âa token used to identify the firmware/software version currently installed on the UA

Serial Âthe token used to identify the serial number for the UA
Mac Âthe token used to identify the MAC address in hex for the UA

From RFC 2543 [6] the From header field syntax is extended to include:

- **from-param** = tag-param | generic-param | device-param
- **device-param** = vendor-param | model-param | version-param | serial-param | mac-param

- **vendor-param** = ÅVendorÅ Å=Å token
- **model-param** = ÅModelÅ Å=Å token
- **version-param** = ÅVersionÅ Å=Å token
- **serial-param** = ÅSerialÅ Å=Å token
- **mac-param** = ÅMacÅ Å=Å token

### 4.3 NOTIFY

The NOTIFY message is sent by the configuration server to convey the URLs at which the UA can retrieve the requested configuration data profiles. This occurs in two contexts:

Immediately following the enrollment SUBSCRIBE the configuration server MUST send a NOTIFY providing URLs for the configuration data profiles requested by the UA in the Config-Request header field of the SUBSCRIBE request. If the configuration server is not able to provide some specific configuration data profiles or it does not want the UA to retrieve some specific configuration profiles at that point in time, it MAY exclude those URL(s) from the NOTIFY. At a later time when the configuration server is able to provide the data profile(s) or it wishes the UA to retrieve the data profiles at that point in time, the configuration server MAY send a NOTIFY request containing the URL(s) for the configuration data profile(s) which the UA SHOULD retrieve immediately.

If the configuration server becomes aware of a configuration change that it wishes to be effective immediately on the UA, the configuration server SHOULD send a NOTIFY message containing the complete list of URLs for the configuration data profiles that the UA requested when it enrolled. The configuration data profiles with changed content SHOULD have sequence number larger than that of the last NOTIFY request. The UA SHOULD retrieve and make effective the changed configuration URLs immediately upon receipt of the NOTIFY request. The UA MAY choose to wait to make the changes effective (e.g. to prevent the change from disrupting active calls on the UA).

[Do we need an option for the configuration server to tell the UA that it MUST make the change immediately regardless of state? Should this be the default?]

The UA SHOULD send a 200 response to the NOTIFY immediately upon receipt and validation of the solicited request. The configuration server SHOULD include, in the change notification NOTIFY request, the complete list of the configuration data profile URLs. The sequence numbers associated with the configuration data profiles with changed content should be larger than those in the previous NOTIFY. These configuration data profile URLs MUST be among those
the UA named in the Config-Requires header field in the most recent enrollment (SUBSCRIBE request). The URLs listed in the NOTIFY request MUST use one of the protocols the UA listed in the Config-Allow header field provided during enrollment in the most recent SUBSCRIBE request. The sequence numbers for the configuration data profile URLs are positive integers chosen by the configuration server. The sequence number value MUST increase monotonically as modifications are made to a data profile.

This mechanism may be used by the configuration server to provide firmware updates. For example on a UA that caches or has a persistent firmware image: if the server realizes (e.g. from the enrollment information) the UA is running the most currently available firmware version, it would not provide the URL for the firmware. However at a later point in time when a new firmware version was available the configuration server could send a NOTIFY with the URL for the new firmware version, indicating the UA SHOULD upgrade now.

4.3.1 NOTIFY Body Content Format

The NOTIFY request contains a body of Content-Type: text/plain. The content is formatted according to RFC 822 [8]. For each of the named configuration data profiles which the configuration server is able to provide, the body contains a header field with the same name as the configuration data profile. The value of the header field MUST contain a URL and a sequence number as described in the syntax below. The protocol of the URL MUST be one of those listed in the Config-Allow header field provided by the UA in the enrollment request. The sequence number associated with the URL is intended to allow the UA to decide if it has the latest content of the configuration data profile without having to download and compare the contents.

Syntax:
config-profile = token : Seq-Param ; Url-Param
Seq-Param = Sequence = 1*digit
Url-Param = Url = tftp-url | Http-url | Https-url
Tftp-url [need reference]
Http-Url as defined in [12, section 3.3]
Https-url [need reference]

Example:
X-Acme-Special: Sequence=1234567;Url=http://www.acme.com/config.txt

5 Configuration Retrieval

The UA MUST retrieve its configuration data profiles using the URLs specified by the configuration server in the NOTIFY request. If any of the retrievals fail, the UA SHOULD re-enroll as described in section 4. Should the enrollment fail, the UA SHOULD re-discover the configuration server as described in section 3.

[Is this a good idea? It might cause a nasty cascade effect if the server for a bunch of URLs goes down. In all likelihood, the configuration server won’t check whether the server is available for
each of the URLs it hands out that would probably be too expensive. The effect will be to have a bunch of UA spinning back and forth between hitting the configuration server and hitting the failed server for the URL. What are the alternatives? Leave the precise procedure up to the UA since some may need more sophisticated cutover mechanisms than others. Retry fetching the URL using an exponential backoff timer between attempts up to some maximum interval. When that limit is reached, the UA can try to re-enroll. However, if the configuration server gives it the same failing URL, it should continue to retry after waiting the maximum interval timeout.)

6 Configuration Upload

If the UA or another entity wishes to modify a configuration data profile it MAY make the change persistent on the configuration server if it is authorized to do so. The configuration server SHOULD support the ability to upload via the same URL the UA used to retrieve the configuration data profile. For TFTP the UA does a put [9]. For HTTP and HTTPS the UA does a POST with a multipart MIME attachment containing any URL parameters in one part and the changed configuration data profile [whole or changes only ?? define in profiles ??] in another part as defined in [?]. If the UA or user is not permitted to make the changes on the configuration server the configuration server returns an HTTP error response code of 403 Forbidden. If the configuration server returns a 403 the UA SHOULD disallow the changes from being effective on the UA. The UA SHOULD not make the changes effective until it receives a successful response (e.g. for HTTP 2xx).

If the URL is for HTTP/HTTPS the server MUST return the changed configuration data profile in the response (assuming it was allowed). The configuration server SHOULD include an incremented sequence number in the HTTP/HTTPS response if the configuration data profile contents changed [Sip-Ua-Config-Seq header field?]. The UA SHOULD use the configuration data profile contents from the HTTP response as opposed to the data that was pushed in the request as changes may occur from other sources. The configuration server SHOULD send out a NOTIFY for this change, using the same sequence number in the configuration data profile URL parameter. This allows the UA to know that it already has the current contents of the configuration data profile and SHOULD not download that configuration data profile.

[TBD Â in 403 case restrict and provide feedback as to what specifically is not allowed to be modified by the UA or user]

7 Examples

Below is an example high level message flow for a new UA discovering and using configuration data from a configuration server. Following the high level message flows are some specific SIP messages illustrating SUBSCRIBE and NOTIFY messages from enrollment and configuration change notification.

7.1 Example Message Flows
The following high level message flows illustrate the configuration process of discovery, enrollment, configuration retrieval and change notification with associated configuration retrieval. The UA uses DHCP with the local option requesting the configuration server address and port. The DHCP server does not provide the configuration server address or port. The UA then does a DNS SRV lookup for the configuration service within the local domain. It gets a response with one configuration server address and port. The UA then enrolls with the configuration by sending a SUBSCRIBE request for the Config-Event. The configuration server sends back a successful response. The configuration server then sends a NOTIFY request with the list of URLs for all the configuration data profiles that the UA named in the enrollment SUBSCRIBE request. The UA sends a 200 response to the NOTIFY. The UA then downloads all of the configuration data profiles via the URLs from the NOTIFY request. The UA is now configured as prescribed.

Later ... an administrator makes a change to the configuration for the UA on the configuration server. The configuration server on behalf of the administrator, sends a NOTIFY (change notification) request to the UA listing the configuration data profiles that changed (the minimum subset of the list of configuration data profiles the UA requested during enrollment). The UA downloads the configuration data profiles that changed.

<table>
<thead>
<tr>
<th>UA</th>
<th>DHCP Server</th>
<th>DNS Server</th>
<th>Config. Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP config. req.</td>
<td>============&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP config. wo/ local option</td>
<td>============&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNS SRV req. for sipuaconfig service in local domain</td>
<td>==============&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Host and port for config. server returned</td>
<td>==============&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enrollment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIP SUBSCRIBE Config-Event w/ requested profile names</td>
<td>==============&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200 OK</td>
<td>===============&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIP NOTIFY Config-Event w/ requested profile URLs</td>
<td>===============&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200 OK</td>
<td>===============&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration retrieval</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP GET (For each profile URL)</td>
<td>===============&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200 OK (specific profile data in body)</td>
<td>===============&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Administrative change on configuration server via user interface

Change Notification

SIP NOTIFY Config-Event w/ changed profile URLs

HTTP GET (for each changed profile)

Configuration Upload

HTTP POST (changed profile attached as multipart MIME)

7.2 Example Messages

The following SUBSCRIBE request example is from a UA enrolling with a configuration server. As this SUBSCRIBE request is for configuration enrollment the Event header field contains the token Config-Event. The UA tells the configuration server that it supports the TFTP, HTTP, HTTPS protocols for retrieving configuration data profiles in the Config-Allow header field. The UA tells the configuration server that it would like the configuration data profiles named: sip-device, sip-user, x-acme-special in the Config-Require header field. The UA tells the configuration server that it is enrolling for 86400 seconds via the Expires header field. During this period of time the configuration server MUST send a change notification listing the configuration data profiles which have changed. The UA has identified the specifics about itself in the From field parameters: Vendor, Model, Version, Serial, Mac.
The following is an example response to the above enrollment request.

Config. Server => UA

SIP/2.0 202 Accepted
To: sip:config.localdomain.com
From: sip:10.1.1.123;Vendor=acme;Model=model-a
    ;Version=1.5.0.1;Serial=1234567890;Mac=000aaa1234cd
Call-Id: 987654321@10.1.1.123
Cseq: 22 NOTIFY
Event: Config-Event
Config-Requires: Sip-Device, Sip-User, X-Acme-Special, X-Acme-Kernel
Expires: 43200
Content-Length: 0

The following example is the immediate NOTIFY request the configuration server sent to the UA following enrollment. The Config-Expires header field indicates the lease length of the configuration data profiles. After that period of time if the UA had not received an additional NOTIFY request from the configuration server it should re-retrieve the configuration data profiles from the provided URLs. The URLs are listed in the request body for each of the named configuration data profiles the UA listed in the Config-Require header field in the above SUBSCRIBE request from the UA, that the configuration server is able or wishes to provide. Note that the configuration server did not provide a URL for X-Acme-Kernel (perhaps it decided that the kernel image on the UA was already current).

Config. Server => UA

NOTIFY sip:10.1.1.123 SIP/2.0
To: sip:10.1.1.123;Vendor=acme;Model=model-a
    ;Version=1.5.0.1;Serial=1234567890;Mac=000aaa1234cd
From: sip:config.localdomain.com
Call-Id: 987654321@10.1.1.123
Cseq: 22 NOTIFY
Event: Config-Event
Config-Expires: 43200
Content-Type: text/plain
Content-Length: 175
Sip-Device: Sequence=1
The following is an example response from the UA for the above request.

```
UA => Config. Server

SIP/2.0 200 Ok
To: sip:10.1.1.123;Vendor=acme;Model=model-a
   ;Version=1.5.0.1;Serial=1234567890;Mac=000aaa1234cd
From: sip:config.localdomain.com
Call-Id: 987654321@10.1.1.123
Cseq: 22 NOTIFY
Content-Length: 0
```

Assuming at some later point in time, an administrator makes a change to the content of the Sip-Device configuration data profile for the UA. The configuration server sends a NOTIFY request to the UA for the configuration change notification. This example request below indicates the changed URL(s) in the request body with a higher sequence number. In this case only one URL has changed, that for the configuration data profile named: Sip-Device. The configuration server extends the configuration lease to 43200 seconds from when this request is received. This lease applies to all of the configuration data profiles that the UA requested when it last enrolled. If the UA does not receive another NOTIFY request from the configuration server before the lease expires, the UA SHOULD download all of the configuration data profiles from the most recent URLs provided for each of the configuration data profiles listed in the enrollment SUBSCRIBE Config-Require header field.

```
Config. Server => UA

NOTIFY sip:10.1.1.123 SIP/2.0
To: sip:10.1.1.123;Vendor=acme;Model=model-a
   ;Version=1.5.0.1;Serial=1234567890;Mac=000aaa1234cd
From: sip:config.localdomain.com
Call-Id: 987654321@10.1.1.123
Event: Config-Event
Cseq: 23 NOTIFY
Config-Expires: 43200
Content-Type: text/plain
Content-Length: 64

Sip-Device: Sequence=2
```
The following is an example response to the above request.

UA => Config. Server

SIP/2.0 200 Ok
To: sip:10.1.1.123;Vendor=acme;Model=model-a
;Version=1.5.0.1;Serial=1234567890;Mac=000aaa1234cd
From: sip:config.localdomain.com
Call-Id: 987654321@10.1.1.123
Cseq: 23 NOTIFY
Content-Length: 0

Security Considerations

[SIP basic and digest authentication [6] MAY be used for SUBSCRIBE/NOTIFY messages used for enrollment and configuration change notification. As there is a chicken and egg problem as well and the content of SUBSCRIBE/NOTIFY messages are transported in the clear, the credentials that the UA uses in the SUBSCRIBE 401 challenge, or that the configuration server uses in the NOTIFY 401 challenge must be provisioned out of band (i.e. user or administrator manual input, beamed via PDA, smart card, etc.) via a secure means.

Configuration data profile URLs are communicated in the clear in the NOTIFY requests from the configuration server. The security risk of unauthorized access of the URL content can be mitigated if the configuration server and UA both support basic authentication and HTTP or HTTPS. There is a chicken and egg problem here as well since the content of SUBSCRIBE/NOTIFY messages are transported in the clear. Accordingly, the credentials that the UA uses for the HTTP/HTTPS GET/POST 401 challenge must be provisioned out of band (i.e. user or administrator manual input, beamed via PDA, smart card, etc.) via a secure means.

Using HTTPS over TLS[13] the configuration server MAY request the certificate of the UA [14]. If this level of authentication is desired, the UA vendor SHOULD ship the UA with a digital certificate or provide a means by which this can be installed out of band. The configuration server MUST be provisioned with the certificates of authority allowed for each model of UA to be supported.

Using HTTPS the UA MAY request the certificate of the configuration server. If this level of authentication is desired the UA must be provisioned with the allowed certificate(s) of authority and
identities for the configuration server out of band (i.e. user or administrator manual input, beamed via PDA, smart card, etc.) via a secure means.

Open Issues

Local DHCP option (i.e. > 128)?
or
DHCP Option for SIP Servers?

How does the configuration server give feedback to a new UA that it SHOULD/MAY prompt the user for or provide configuration data specifics?

[Request URI should not contain a user id? The user may not be known yet.]

[What happens when the config server receives multiple SUBSCRIBE requests from the same UA but for different list of profiles. Does the last request supercede all previous ones?]

[What happens when the config server receives multiple SUBSCRIBE requests from the same IP address but for different devices? This might happen if some entity is acting as a proxy for a bunch of other devices. It might also happen if the IP address for a UA gets reused for some other UA (the DHCP lease timeout may be much shorter than the SUBSCRIBE/NOTIFY lease timeouts). The SUBSCRIBE lease SHOULD not exceed the DHCP lease? The UA SHOULD re-enroll if its IP address changes?]

[Do we need an option for the configuration server to tell the UA that it MUST make the change immediately regardless of state? Should this be the default?]

[Upload to configuration server configuration data profiles whole or changes only ?? define in profiles ??]

[Security considerations section needs much elaboration]
10 References


User Agent Configuration


Author's Addresses

Dan Petrie
Pingtel Corp.
400 W. Cummings Park Phone: +1 781 938 5306
Woburn, MA USA Email: dpetrie@pingtel.com