D. Petrie
Pingtel Corp.

Internet Draft
draft-petrie-sip-config-framework-01.txt
Expires: May 2002

November 2001

A Framework for SIP User Agent Configuration

Status of this Memo

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

Status of this Memo....1
Abstract....1
1 Overview....3
2 Conventions used in this document....4
3 Changes from Previous Draft....4
4 Discovery....4
4.1 DHCP Option....6
4.2 DNS....6
4.3 Multicast....6
4.4 Manually Provisioned....6
5 Enrollment and Change Notification....6
5.1 Header Field Definitions....7
5.1.1 Config-Allow....7
5.2 SUBSCRIBE....8
5.2.1 Additional From Field Parameters....8
5.3 NOTIFY....9
5.3.1 NOTIFY Body Content Format....10
6 Configuration Retrieval....11
7 Configuration Upload....11
8 Examples....12
8.1 Example Message Flows....12
8.2 Example Messages....14
9 Security Considerations....17
10 Open Issues....18
11 References....19
12 Author's Addresses....20

1 Overview

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.

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, name requested configuration data profile 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. Each profile requires a separate enrollment or SUBSCRIBE session.

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.

2 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 Changes from Previous Draft

Many thanks to those who contributed and commented on the previous draft. Detailed comments were provided by Henning Schulzrinne from Columbia U., Cullen Jennings from Cisco, Rohan Mahy from Cisco.

Split the enrollment into a single SUBSCRIBE dialog for each profile. The 00 draft sent a single SUBSCRIBE for listing all of the desired. These have been split so that each enrollment can be routed differently. As there is a concept of device specific and user specific profiles, these may also be managed on separate servers. For instance in a roaming situation the device might get it s configuration from a local server which knows the LAN specific configuration. At the same time the user specific profiles might come from the user s home environment configuration server.'

Removed the Config-Expires header as it is largely superfluous with the SUBSCRIBE Expires header.

Eliminated some of the discovery nonsense.

Suggest caching of discovered configuration server to avoid avalanche problem when a whole building full of devices powers up.

Added the User-Profile From header field parameter so that the device can indicate the request for a user specific profile different than the device s default user.

4 Discovery

The purpose of discovery is to figure out how to address the configuration server so that the device can enroll. The enrollment process involves sending a SIP SUBSCRIBE. Prior to this the discovery process must find the address to use in the URL for the URI and To header field. The URL SHOULD use the user id:

sipuaconfig. From a SIP perspective the configuration server is simply a user agent. By using a well known user id, this makes it easy for proxy servers to be provisioned to route the enrollment requests from devices to the appropriate configuration server for the domain.

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 option for SIP [1]
- 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. 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. This includes the ability to manually override the discovery process. However by default without user interaction it SHOULD use the order listed above.

Once discovery is successful the device SHOULD persistently cache the address to avoid avalanche problems when a whole building full of devices powers up at once. The characteristic of the profile may dictate this behavior. For example device specific profiles may need to change when the device is moved to a different location.

User specific profiles may be independent of the LAN, network or device location.

4.1 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 address. In the same DHCP request for basic IP configuration, the UA can add the option for SIP[3] [1] to the options field. This indicates a request for the default SIP proxy server address and port. For example if the DHCP option for SIP returns an address of sip.acme.com and a port of 5080, the following URL is constructed: sip:sipuaconfig@sip.acme.com:5080. If the proxy server address and port is not returned in the DHCP response or the server does not respond to the enrollment request with a successful 2xx class response, the next discovery mechanism is attempted.

4.2 DNS

The UA SHOULD construct a fully qualified host name using

sipuaconfig as the host and the local domain if defined. It SHOULD try a DNS A record lookup on the fully qualified host name. If the name resolves in DNS it should then attempt enrollment. For example the URL constructed in the local domain of acme.com would look like: sip:sipuaconfig@sipuaconfig.acme.com. If the server does not respond to enrollment with a successful 2xx class response, the next discovery mechanism is attempted.

4.3 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.

4.4 Manually Provisioned

The UA MAY indicate to the user (or administrator) that automatic discovery has failed. The UA SHOULD allow the user or administrator to manually (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.

5 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.

Each profile that the device requires is obtained via a separate

enrollment or SUBSCRIBE request and SIP dialog. That is for each different profile a device enrolls for, a different Call-Id is used. The device names the profile in the SUBSCRIBE Event header field. The configuration server then delivers a URL at which the device and retrieve the profile in a subsequent NOTIFY request. Changes to the profile are indicated in additional NOTIFY requests sent from the configuration server.

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 a new Event token must be named for each type of profile.

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 field is defined for use in SUBSCRIBE and NOTIFY requests for the purpose of enrollment and configuration change notification:

The keys used the following table:
R request
r response
m mandatory
o optional
- - not applicable

Header	Where	SUBSCRIBE	NOTIFY
Config-Allow	R	m	-

5.1 Header Field Definitions

5.1.1 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 URL for the configuration profile in the NOTIFY request.

Syntax:

```
Config-Allow  =  "Config-Allow" ":" config-protocol
                 *( ,  config-protocol)
```

```
config-protocol  =  tftp  |  http  |  https  | token
```

5.2 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. The UA MUST include the Config-Allow header field which MUST contain at least one token. The configuration server SHOULD not send an error if it is temporarily not able to provide the configuration data profile listed in the SUBSCRIBE request Event header field. In the first time out of the box case, the SUBSCIRBE dialog may be the only means of communicating with the device as it does not yet have configuration. The configuration server SHOULD send a 403 response to the SUBSCRIBE if is not willing to provide the requested configuration profile to the device. 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 device may request many configuration data profiles by sending multiple SUBSCRIBE requests each in a different SIP dialog. This may be useful if the device requires user specific profiles for multiple users. In this case the UserProfile parameter would vary for each SUBSCRIBE. Alternatively the device may require multiple types of profiles where each SUBSCRIBE would have a different Event header field token.

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).

5.2.1 Additional From Field Parameters

When the device first starts up out of the box, it has no user or local configuration. The device MUST to provide a unique identity

such that it is possible for the configuration server to generate configuration profiled for the device. The following additional From field parameters are defined for the purpose of identifying the UA device:

Vendor a token used to identify the UA vendor name

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

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

Serial the token used to identify the serial number for the UA

Mac the token used to identify the MAC address in hex for the UA

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

```
from-param  =  tag-param | generic-param | device-param
device-param  =  vendor-parm | model-parm | version-parm |
                    serial-parm | mac-parm | user-profile-param
vendor-parm  =   Vendor   =  token
model-parm  =   Model   =  token
version-parm  =   Version   =  token
serial-parm  =   Serial   =  token
mac-parm  =   Mac   =  token
user-profile-param =  UserProfile   =  SIP-URL
```

The Vendor, Model, Version, Serial and Mac parameters MUST be provided in the From URL for the enrollment SUBSCRIBE request. Most profiles will either be device or user specific. If the profile is user specific and the device knows the user for which it is to retrieve, the profile UserProfile MUST be provided. If the profile is device specific or the device does not know whose user profile it should retrieve, the device SHOULD not provide the UserProfile parameter. The profile is user specific and UserProfile is not present the configuration server assumes the default user for the device.

5.3 NOTIFY

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

Immediately following the enrollment SUBSCRIBE the configuration server MUST send a NOTIFY providing the URL for the configuration data profile requested by the UA in the Event header field of the SUBSCRIBE request. If the configuration server is not able to provide the specific configuration data profile or it does not want the UA to retrieve the specific configuration profile at that

point in time, it MAY defer sending NOTIFY. At a later time when the configuration server is able to provide the data profile or it wishes the UA to retrieve the data profile at that point in time, the configuration server MAY send a NOTIFY request containing the URL for the configuration data profile which the UA SHOULD retrieve as soon as it is safe to do so.

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 URL for the configuration data profile that the UA requested when it enrolled. The configuration data profile 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 URL 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 MUST include, in the change notification NOTIFY request, the configuration data profile URL. The sequence numbers associated with the configuration data profile with changed content should be larger than those in the previous NOTIFY. The URL 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 number for the configuration data profile URL MUST be 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 could defer sending the NOTIFY with 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 as soon as it is safe to do so.

5.3.1 NOTIFY Body Content Format

The NOTIFY request contains a body of Content-Type: text/plain. The content is formatted according to RFC 822 [8]. It contains a header field with the same name as the configuration data profile as indicated in the Event header. 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 SUBSCRIBE 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

6 Configuration Retrieval

The UA MUST retrieve its configuration data profile using the URL specified by the configuration server in the NOTIFY request. If the retrieval fails, the UA SHOULD not re-enroll until the SUBSCRIBE session expires to avoid a cascade effect if the server goes down temporarily. The device MAY re-try the profile retrieve of the profile from the URL before the SUBSCRIBE expires. Should the re-enrollment fail, the UA SHOULD re-discover the configuration server as described in section 4.

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

8 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.

8.1 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 lookup for the configuration service within the local domain. It gets a response from the DNS server for the configuration server fully qualified host name. The UA then enrolls with the configuration by sending a SUBSCRIBE request for the profile type indicated in the Event header. The configuration server sends back a successful response. The configuration server then sends a NOTIFY request with the URL for the configuration data profile that the UA named in the

enrollment SUBSCRIBE request. The UA sends a 200 response to the NOTIFY. The UA then downloads the configuration data profile via the URL from the NOTIFY request. This process may be repeated in parallel for each of the required profiles. 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 with an incremented sequence number for the profile. As the sequence number has changed, the UA downloads the configuration data profile from the given URL.

.
.
.

```
Administrative change on configuration server via user interface
.
.
.

Change Notification

SIP NOTIFY Event: Sip-Device w/ changed profile URL
<==================================================
200 OK
==================================================>
HTTP GET
==================================================>
200 OK (profile data in body)
<==================================================
.
.
.

User changes data in a profile on the user agent
.
.
.

Configuration Upload
```



```
HTTP POST (changed profile attached as multipart MIME)
==================================================>
200 OK (profile data in body, as change confirmation)
<==================================================
.
.
.
```

8.2 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 profile named: sip-device in the Event 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 with the URL for the configuration data profile which changed. The UA has identified the specifics about itself in the From field parameters: Vendor, Model, Version, Serial, Mac. In this example the UserProfile parameter is not included in the From field as the Sip-Device profile is device specific not user specific.

UA => Config. Server

```
SUBSCRIBE sip: sipuaconfig@config.localdomain.com SIP/2.0
To: sip:sipuaconfig@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: 1 SUBSCRIBE
Event: Sip-Device
Config-Allow: tftp, http, https
Expires: 86400
Content-Length: 0
```

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: 1 SUBSCRIBE
Content-Length: 0
```

In the following example the device is requesting a user specific profile Sip-User. The device specifies that it want the profile for the user: sip:fredsmith@localdomain.com.

UA => Config. Server

```
SUBSCRIBE sip: sipuaconfig@config.localdomain.com SIP/2.0
To: sip:sipuaconfig@config.localdomain.com
From: sip:10.1.1.123;Vendor=acme;Model=model-a
  ;Version=1.5.0.1;Serial=1234567890;Mac=000aaa1234cd
  ;UserProfile=sip%3Afredsmith%40localdomain.com
Call-Id: 11111111@10.1.1.123
Cseq: 1 SUBSCRIBE
Event: Sip-Device
Config-Allow: tftp, http, https
Expires: 86400
Content-Length: 0
```

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
  ;UserProfile=sip%3Afredsmith%40localdomain.com
Call-Id: 11111111@10.1.1.123
Cseq: 1 SUBSCRIBE
Content-Length: 0
```

The following example is the immediate NOTITY request the configuration server sent to the UA following enrollment. The URL in the request body is for the configuration data profile the UA named in the Event header field in the above SUBSCRIBE request from the UA.

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: Sip-Device
```

```
Content-Type: text/plain
Content-Length: 79

Sip-Device: Sequence=1
  ;Url=http://config.localdomain.com/device/1234567890
```

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 or content in the request body with a higher sequence number.

```
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: Sip-Device
Cseq: 23 NOTIFY
Content-Type: text/plain
Content-Length: 79

Sip-Device: Sequence=2
  ;Url=http://config.localdomain.com/device/1234567890
```


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
```

9 Security Considerations

[This section needs to be greatly expanded and elaborated]

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.

10 Open Issues

[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]

11 References

[1] R. Droms, "Dynamic host configuration protocol," Request for Comments (Draft Standard) 2131, Internet Engineering Task Force, Mar. 1997.

[2] S. Alexander and R. Droms, "DHCP options and BOOTP vendor extensions," Request for Comments (Draft Standard) 2132, Internet Engineering Task Force, Mar. 1997.

[3] G.Nair, H.Schulzrinne , DHCP Option for SIP Servers , <draft-ietf-sip-dhcp-04.txt>, IETF; Mar. 2001, Work in progress.

[4] S. Bradner, "Key words for use in RFCs to indicate requirement levels," Request for Comments (Best Current Practice) 2119, Internet Engineering Task Force, Mar. 1997.

[5] A. Gulbrandsen, P. Vixie, and L. Esibov, A DNS RR for specifying the location of services (DNS SRV), Request for Comments 2782, Internet Engineering Task Force, Feb. 2000.

[6] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, SIP: session initiation protocol, <draft-ietf-sip-rfc2543bis-05.txt>, IETF; Oct. 2001, Work in progress.

[7] A. Roach, Event Notification in SIP , <draft-ietf-sip-events-01.txt>, IETF; Nov. 2001, Work in progress.

[8] D. Crocker, STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES , Request for Comments 822, Internet Engineering Task Force, Aug. 1982

[9] K. Sollins, THE TFTP PROTOCOL (REVISION 2) , Request for Comments 1350, Internet Engineering Task Force, Jul. 1992

[10] H Schulzrinne, Configuring IP Telephony End Systems , <schulzrinne-sip-config-00.txt>, IETF; Dec. 2000, Work in progress

[11] D. Petrie, Requirements for a SIP User Agent Configuration Framework , <draft-petrie-sip-config-framewk-reqs-00.txt>, IETF; Feb. 2001, Work in progress

[12] T. Berners-Lee et al, Uniform Resource Locators (URL) ,

Request for Comments 1738, Internet Engineering Task Force, Dec. 1994

[13] E. Rescorla, HTTP Over TLS , Request for Comments 2818, Internet Engineering Task Force, May 2000

[14] T. Dierks, C. Allen, The TLS Protocol Version 1.0 , Request for Comments 2246, Internet Engineering Task Force, Jan. 1999

12 Author's Addresses

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