Internet Engineering Task Force (IETF)
Request for Comments: 6011
Category: Informational
ISSN: 2070-1721

S. Lawrence, Ed.
Linden Research, Inc.
J. Elwell
Siemens Enterprise Communications

October 2010

Session Initiation Protocol (SIP) User Agent Configuration

Abstract

This document defines procedures for how a SIP User Agent should locate, retrieve, and maintain current configuration information from a Configuration Service.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6011.

Table of Contents

1. Introduction

A user gets a new SIP User Agent (UA); it may be a hardware device or software. Some User Agents have a user interface that can accept a username, password, and domain name. Other devices, like Analog Telephony Adapters (ATAs), have no user interface other than that provided by an attached analog phone. How does a non-technical user minimally configure it so that when it is started, something useful happens?

1.1. Scope

This document specifies a procedure for how a SIP User Agent locates, retrieves, and maintains current configuration information for a given SIP Service Provider. As such, it specifies requirements to be met by both the User Agent, the Configuration Service at the SIP Service Provider, and the network infrastructure services that allow them to communicate.

Nothing in this specification prohibits a User Agent from obtaining configuration information by any means in addition to the mechanisms specified herein.

The intent of this specification is to provide mechanisms sufficient for User Agents to discover an appropriate source of configuration and maintain the currency of that configuration. A User Agent implementation compliant with this specification MAY also implement additional mechanisms necessary in particular environments or when the services specified here are not available.

The form and content of configuration data to be downloaded are outside the scope of this specification, although Section 3.1, "Configuration Data Items" suggests a minimum set of data items likely to be required by all types of UAs.

1.2. Terminology

The following terms are used in this document:

User Agent, UA

> As defined in RFC 3261 [RFC3261]. Note that this includes any implementation of a User Agent. A SIP phone is a User Agent, but the term also encompasses any other entity that uses SIP (for example, for a text chat, for sharing a whiteboard, or for a fax).

Soft User Agent, Soft UA

A User Agent that runs as an application within some larger system that has responsibility for some of the steps described in this specification. In those cases, the Soft UA must be able to obtain the information from the platform. In all cases, the term User Agent also encompasses a Soft User Agent.

SIP Service Provider, Service Provider

An entity that provides services to User Agents using the SIP protocol. This specification requires that a Service Provider make configuration data and certain other information available in order to configure User Agents.

Configuration

The set of information that establishes operational parameters for a particular User Agent.

Configuration Service, CS

The source of Configuration for User Agents.

Configuration Service Domain

The DNS name for the service from which a Configuration is requested.

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 [RFC2119].

1.3. User Agent Installation Examples

This section is non-normative; it is a set of "user stories" -- narrative descriptions of the user experience in different environments. These are "black box" descriptions meant to include the actions to be taken by the human participants (including administrators and system operators as well as the "user" of the UA), but not how the network elements communicate or operate internally. The intent is that these narratives provide context for the subsequent technical specifications.

1.3.1. Hosted IP Service Provider Example

Configuring a new UA to use a hosted IP telephony service will typically proceed as follows: the customer makes a request to their Service Provider to add one or more new users to their service. The customer may supply further details such as a preferred username, type of endpoint and any requests for specific functionality, depending on what information the Service Provider considers useful, but no additional information is required from the customer.

The Service Provider performs any necessary provisioning actions on their equipment, and returns to the customer provisioning information, which may include a domain name or a numeric domain identifier for the provider, a user identifier, and a password. Typically, a Service Provider will supply provisioning information for each device to be provisioned, but may choose to supply information that can be used with multiple devices, or for a limited duration or with other benefits and restrictions.

The customer enters the provisioning information into the UA to be configured, whereupon the UA uses this information to locate the configuration service, securely fetch the configuration information, and configure itself for operation.

1.3.2. IP-PBX Example

Configuring a new UA in a typical business begins by provisioning a user identity in the Private Branch Exchange (PBX) (add user "John Smith"), and assigning a phone number to the user. That number must then be assigned to a line on a specific UA; this is usually done by selecting a UA and provisioning it in the PBX by its serial number (usually a Media Access Control (MAC) address), and then assigning the identity or phone number to a 'line' on that UA in the PBX configuration system.

Once provisioning in the PBX is complete, the new user goes to his or her workplace and connects the UA to the network. When connected and powered up, the UA is provided with the user identity, phone number, and any other configuration data with no local user interaction -- just connecting it to the network loads the configuration from the PBX and the UA is operational.

1.3.3. Special Considerations for High Security Deployments

To deploy a new UA in a high security scenario requires some special consideration. A security-conscious deployment will most likely require that the SIP and other management interfaces, including the interface to the configuration service, be secured before the device is put into service.

In order to achieve any level of security, the device will need to be pre-configured with some security-related information in the form of certificates. This may be achieved in a number of ways. Some examples include:

1. An administrator who configures the device in a secure environment before making the device available to the user.

2. Some certificates may be built into the device during the manufacturing process enabling the configuration service to certify information such as the manufacturer, UA type, and MAC address. The configuration service may then be used to provision the device with other certificates as required.

3. The device may have a facility for the user to provide the security information in the form of a security card or dongle.

All these mechanism are likely to restrict the user to a limited set of devices approved for use in a particular deployment.

2. Obtaining User Agent Configuration

This section specifies how a User Agent connects to the network, determines for which domain to request configuration, obtains configuration from that domain, and is notified by that domain when the configuration changes.

The User Agent MAY obtain configuration information by any means in addition to those specified here, and MAY use such information in preference to any of the steps specified below, but MUST be capable of using these procedures alone in order to be compliant with this specification.

2.1. Network Discovery

A UA needs a minimum set of parameters to allow it to communicate on the network. Some networks allow the UA to automatically discover these parameters, while other networks require some or all of these parameters to be manually provisioned on the UA.

2.1.1. Link Layer Provisioning

The UA SHOULD attempt to use Link Layer Discovery Protocol - Media Endpoint Discovery (LLDP-MED; see [ANSI.TIA-1057-2006]) for automatic provisioning of link layer parameters.

In some deployments, failure to properly provision the link layer may result in the UA having incorrect Layer 2 priority, degrading the quality of service, or being on the wrong virtual LAN (VLAN), possibly resulting in complete loss of service.

2.1.2. Network Layer Provisioning

In order to communicate using IP, the UA needs the following minimal IP configuration parameters:

IP Network Parameters

- UA IP Address
- Subnet Mask
- Gateway IP address
- DNS Server IP address(es)

With the exception of a Soft UA that relies on its platform to obtain the IP Network Parameters:

- If the User Agent is using IP version 4 on a network technology for which the Dynamic Host Configuration Protocol (DHCP) [RFC2131] is defined, the UA MUST attempt to obtain the IP Network Parameters using DHCP and MUST request DHCP options 141 (see Section 4.1) and 15 [RFC2132]. If the DHCP service provides a value for option 141, the domain name(s) it provides MUST be saved as candidates for use as the Local Network Domain (see Section 2.2, "Obtaining the Configuration Service Domain"). If and only if no values are returned for option 141, the UA MUST save any values returned for option 15 for use as the Local Network Domain.
- If the User Agent is using IP version 6 on a network technology for which the Dynamic Host Configuration Protocol version 6 (DHCPv6) [RFC3315] is defined, the UA MAY use any standard IPv6 mechanism to determine the IP Network Parameters, but MUST request DHCPv6 options 58 (see Section 4.2) and 21 [RFC3319]. If the DHCPv6 service provides a value for option 58, those domain names MUST be saved as candidates for use as the Local Network Domain

(see Section 2.2, "Obtaining the Configuration Service Domain"). If and only if no values are returned for option 58, the UA MUST save any values returned for option 21 for use as the Local Network Domain.

2.2. Obtaining the Configuration Service Domain

To obtain a configuration, the UA needs to know what domain to request it from. This domain is the Configuration Service Domain; its value is a DNS name (see [RFC1034]).

User control or prior configuration MAY establish a value for the Configuration Service Domain that takes precedence over the discovery procedure defined below. In the absence of user control or prior configuration, candidate values for the Configuration Service Domain are obtained as specified in Section 2.2.1, "The Local Network Domain", or if that is unsuccessful, by the manual mechanism specified in Section 2.2.2, "Manual Domain Name Entry".

2.2.1. The Local Network Domain

The UA MUST attempt to use each value obtained for the Local Network Domain name (see Section 2.1.2, "Network Layer Provisioning") as the Configuration Service Domain. If multiple names are provided by DHCP and/or DHCPv6 (multiple names may be returned by these services if both are in use, if the UA has multiple network interfaces, or if the option responses have multiple values), the UA MUST attempt to use each of the names provided until a configuration is successfully obtained. The order in which values obtained in different responses are used is not defined by this specification -- the UA MAY use any order; multiple values returned within a single response MUST be tried in the order they were provided in that response.

If the DHCP service does not provide any local domain name values, the UA SHOULD use the manual mechanism defined in Section 2.2.2, "Manual Domain Name Entry".

2.2.2. Manual Domain Name Entry

A UA MAY provide an interface by which a DNS name is supplied directly by the user for the Configuration Service Name.

2.3. Constructing the Configuration Request URL

Using the Configuration Service Domain name obtained in Section 2.2, "Obtaining the Configuration Service Domain", the UA MUST construct an HTTPS URL [RFC2818] with which to request configuration. Constructing this URL consists of two parts:

- Section 2.3.1, "Obtaining a Configuration Service Base URL"

- Section 2.3.2, "Adding Configuration Request Parameters"

2.3.1. Obtaining a Configuration Service Base URL

The Configuration Service Domain is resolved to one or more URLs using the URI-enabled Naming Authority Pointer (U-NAPTR) DDDS application defined in "Domain-Based Application Service Location Using URIs and the Dynamic Delegation Discovery Service (DDDS)" [RFC4848].

The lookup key for the U-NAPTR request is the Configuration Service Domain name determined in Section 2.2, "Obtaining the Configuration Service Domain". The UA MUST make a DNS request for NAPTR records for that domain name. From the returned records, the UA MUST select those whose Service field value is "SFUA.CFG"; from those records, the UA MUST extract the HTTPS URL of the Configuration Service from the Regular Expression field (see next paragraph for the construction of that field value).

The NAPTR records for the Configuration Service Domain name whose Service field value is "SFUA.CFG" MUST be configured with the Flag field set to "U", an empty Substitution field, and a Regular Expression field value of the following syntax (i.e., a regular expression to replace the domain name with an https URI):

```
u-naptr-regexp = "!.*!" <URI> "!"
```

where <URI> is as defined in STD 66 [RFC3986], the URI syntax specification, and where the scheme of the URI is "https".

Note that the UA does not need to implement a general regular expression evaluator in order to process the record above correctly. The URI value can be extracted by stripping the fixed value "!^.*!" from the beginning of the value, and "!" from the end of the value to obtain the base URL. See Section 2.3.3, "Configuration Request URI Example".

2.3.1.1. Configuration Service Redundancy

Multiple Configuration Servers can be used to provide redundancy and additional capacity for provisioning User Agents. If the DNS NAPTR request for the Configuration Service Domain name returns multiple records with the 'SFUA.CFG' service tag, then the UA should treat the resulting URLs as alternatives, ordered according to the rules for the priority and weight as specified for NAPTR records.

In addition to redundancy provided by multiple NAPTR records, resolution of the host part of the HTTPS URL can produce multiple results.

2.3.1.2. Configuration Service Name to Base URL Resolution Failure

If the DNS request to resolve the Configuration Service Domain name to a request URL does not receive any response, the UA should follow standard DNS retry procedures.

If the DNS request to resolve the Configuration Service Domain name to a host name returns a response that indicates that no matching result is available (NXDOMAIN), the UA SHOULD attempt to obtain another Configuration Service Domain name using the procedures in Section 2.2, "Obtaining the Configuration Service Domain".

2.3.2. Adding Configuration Request Parameters

To construct the full configuration request URL, the UA adds one or more parameters to the base URLs to specify what configuration the UA is requesting.

1. The UA MUST add all parameters from those defined in the Configuration Request Parameters list below for which the UA has a value. Any parameter from that set for which the UA does not have a value MUST be omitted.

2. The query parameter names defined by this specification all begin with the prefix 'sfua-'. All names beginning with the prefix 'sfua-' are reserved for this specification and future revisions. The UA MUST NOT include any request parameter whose name begins with the prefix 'sfua-' that is not defined by this specification (including any future revisions).

3. Any parameter not defined by the specification is allowed, but MUST be ignored by any Configuration Service that does not recognize it.

2.3.2.1. Configuration Request Parameters

The following parameters are defined for the configuration request. Section 4.4 creates an IANA registry for these and any parameters defined in the future.

sfua-id

The URN identifying the User Agent, constructed as specified in Section 4.1 of [RFC5626] "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)".

Since the procedure defined by [RFC5626] allows any UA to construct a value for this parameter, the sfua-id parameter MUST always be included.

If the UA implements [RFC5626], and includes the '+sip.instance' Contact header field parameter in any request, when requesting configuration it MUST use the same value for the sfua-id parameter.

sfua-user

An identifier for a user associated with the configuration. Note that this might be different than any SIP 'user' in the UA configuration: it could, for example, be the login name of an account on the service provider web site. The syntax of this parameter is that of the 'userid' [RFC2617].

See Section 2.4.1, "Configuration Data Request Authentication" for how this parameter relates to authentication of the configuration data request.

sfua-vendor

An identifier that specifies the vendor of the User Agent. The syntax of the value of this parameter is that of a DNS domain. The domain value MUST be that of a domain owned by the vendor.

sfua-model

An identifier that further specifies the User Agent from among those produced by the vendor. The syntax of the value of this parameter is the same as the 'token' [RFC3261]. Values for this parameter are selected by the vendor.

sfua-revision

An identifier that further specifies the User Agent from among those produced by the vendor. The syntax of the value of this parameter is the same as the 'token' [RFC3261]. Values for this parameter are selected by the vendor.

2.3.3. Configuration Request URI Example

Using the rules in Section 2.2, "Obtaining the Configuration Service Domain", the UA has determined that the Configuration Service Domain value is "example.net". To obtain the base URL, the UA constructs the DNS NAPTR request for "example.net.", which returns the DNS records:

```
NAPTR  10 10 "u" "SFUA.CFG" "!^.*$!https://p1.example.net/cfg!" ""
NAPTR 100 10 "u" "SFUA.CFG" "!^.*$!https://p2.example.net/cfg!" ""
NAPTR  90 50 "s" "SIP+D2T"  ""  _sip._tcp.example.net.
NAPTR 100 50 "s" "SIP+D2U"  ""  _sip._udp.example.net.
```

Figure 1: Example Configuration Service NAPTR Query Results

The records with the service-field "SFUA.CFG" each provide a base URL value for SIP UA configuration requests.

Our hypothetical example communications device is a 'HypoComm' version 2.1, made by ExampleCorp, and has the link layer MAC address of 00:11:22:33:44:55. It does not have any prior knowledge of a user identity for which to request configuration, so it constructs query parameters using the values it does have, combining each with the base URL to create these request URLs (lines wrapped for readability):

```
https://p1.example.net/cfg
   ?sfua-id=urn:uuid:00000000-0000-1000-8000-001122334455
   &sfua-vendor=examplecorp.com
   &sfua-model=HypoComm
   &sfua-revision=2.1
https://p2.example.net/cfg
   ?sfua-id=urn:uuid:00000000-0000-1000-8000-001122334455
   &sfua-vendor=examplecorp.com
   &sfua-model=HypoComm
   &sfua-revision=2.1
```

Figure 2: Example Configuration Request URLs

2.4. Obtaining Configuration from the Configuration Service

To request configuration using a URL constructed as specified in Section 2.3, "Constructing the Configuration Request URL" the User Agent MUST do an HTTPS GET request to each of the URLs until a configuration that the UA can use is returned in response to one of the requests.

A successful final response from the Configuration Service to a GET request for configuration data MUST contain configuration data for the UA in the HTTP response body. Note that the full capabilities of HTTP as specified in [RFC2616] are available to the CS, so responses such as redirection can be used by the CS as a part of the process of providing configuration data.

Configuration data returned in a successful response is subject to change by the CS. The HTTP cache control metadata (the max-age directive value from any Cache-Control header, and the Etag and Last-Modified header values) returned in the response that provides configuration data is used to determine when a configuration change has occurred (Section 2.5.1.3, "Configuration Change Notices") and to validate any stored configuration data (Section 2.6, "Validity of Stored Configuration Data").

- An HTTP response from the CS that provides configuration data MUST include cache control metadata sufficient to ensure that when a new configuration is available, the cache control information for that new data is different.

- The UA MUST retain all of the HTTP cache control metadata from any response that provides configuration data.

2.4.1. Configuration Data Request Authentication

Since the Configuration Request URL scheme is HTTPS, the UA MUST always use Transport Layer Security (TLS) [RFC5246] to establish a connection with the Configuration Service.

The UA MUST provide a server_name extension in the TLS Client Hello message as defined in [RFC4366] "Transport Layer Security (TLS) Extensions", whose value is the Configuration Service Domain name (note that this might not be the same as the host part of the CS base URL). This allows the CS to identify and provide a server certificate containing the desired identity (allowing for a single server to serve multiple domain names).

A UA MUST attempt to validate the server certificate provided by the CS, in accordance with the rules defined in Section 3.1 of [RFC2818]. Unfortunately, the validation attempt might fail (e.g., because the UA might not have in firmware a trusted root CA cert to which the CS certificate chain can be connected or because the root CA cert has expired since the UA firmware was last updated). If the UA is unable to validate the server certificate provided by the CS, the UA SHOULD store the server certificate and alert the user if that CS host provides a different certificate in the future. Although this 'trust on first use' model is not as secure as certificate validation, it does give some protection against man-in-the-middle (MITM) attacks in the future.

If it has one, the UA MUST provide a client certificate. The CS MUST validate the UA client's certificate, if one is provided. If the CS is unable to authenticate the certificate provided by the UA (for example, the UA is using a self-signed certificate), then the CS MAY choose to cache the certificate, provided that the UA successfully authenticates using HTTP authentication (see next paragraph). This allows a CS to treat the digest authentication credentials as a single-use password to authenticate the client certificate. This 'trust on first use' model provides protection against future MITM attacks, provided that the initial communication is not compromised.

If the CS requires HTTP authentication of the configuration data request, the HTTP 'username' parameter used MUST be the same value as the sfua-user value provided in the configuration data request parameters. The UA MUST implement both Basic and Digest authentication as specified by [RFC2617].

2.4.2. Configuration Data Request Failure

The HTTP configuration data request can fail in a number of ways; the error handling for each is defined below:

- If a DNS request to resolve the host name in the request URL returns a response that indicates that no matching result is available (NXDOMAIN), the UA MUST remove that request URL from the list of alternatives for the Configuration Service Domain.

- If the attempt to open a TCP connection to the host in the request URL fails, the UA MAY attempt requests to any alternative URLs for the same configuration service without waiting between alternatives, but any requests to the same host MUST wait between requests according to the procedure defined in Section 2.7, "Retry Backoff Procedure".

- If the TCP connection succeeds but the TLS handshake fails, including failure of the UA to validate the certificate provided by the Configuration Service host (if the UA is capable of validation), the UA MUST remove the failed URL from the list of alternative URLs for this Configuration Service Domain.

- If the request returns a permanent HTTP failure response (response code >= 400, and does not contain a Retry-After header field), the UA MUST remove the failed URL from the list of alternatives for this Configuration Service Domain.

- If the list of alternatives for this Configuration Service Domain becomes empty, the UA MUST attempt to obtain another Configuration Service Domain name using the procedures in Section 2.2, "Obtaining the Configuration Service Domain".

- If the UA has reached its chosen maximum number of retries (this specification does not specify a maximum number of retries, but any retries to the same host MUST follow the procedure defined in Section 2.7, "Retry Backoff Procedure"), the UA MAY attempt to obtain another Configuration Domain name using the procedures in Section 2.2, "Obtaining the Configuration Service Domain".

2.5. Configuration Changes

The configuration data provided by the CS is subject to change. This specification provides for two mechanisms by which the UA discovers that a configuration change is available:

- SIP subscription by the UA to the CS for notification of changes to the configuration data.

- HTTP polling by the UA of the configuration data URL at the CS.

The choice of mechanism is made by the Configuration Service and signaled to the UA in each HTTP response that provides configuration data. In such a response, the CS MUST either:

- Indicate that the UA is to subscribe for change notifications by including a Link header in the response with the link relation 'monitor' and SIP URI. This choice is specified in Section 2.5.1, "Configuration Change Subscriptions".

- Indicate that the UA is to poll for updates using HTTP by not including a Link header with the link relation 'monitor'. This choice is specified in Section 2.5.2, "Configuration Change Polling".

A User Agent MUST support both mechanisms, and use the mechanism indicated by the Configuration Service.

2.5.1. Configuration Change Subscriptions

If the CS chooses to use the SIP subscription mechanism, it MUST include a Link header in the HTTP configuration data response as specified by [RFC5989]; the URI value in the Link header MUST be a SIP URI, and the link relation ('rel' attribute) value MUST be 'monitor'. The 'monitor-group' relation MUST NOT be used -- see below for rules regarding monitoring of multiple configuration data resources. The SIP URI returned in the Link header is the 'configuration change subscription URI'.

A UA that receives a successful configuration data response with a Link header that specifies a 'monitor' relation MUST attempt to maintain a subscription to the SIP URI from the Link header in that response for the http-monitor event package. This subscription is referred to herein as a "configuration change subscription".

The CS MUST accept properly authenticated SUBSCRIBE requests from the UA for the http-monitor event package at the URI it provided in the Link header of a configuration data response. Authentication of the SUBSCRIBE request uses any standard SIP authentication mechanism with credentials supplied to the UA in the configuration data.

Configuration data MAY include references in the form of additional URLs at the CS that the UA MUST use to obtain additional data. Any response to requests for these additional URLs that provide configuration data MUST provide cache control data and a configuration change subscription URI. The CS MAY return a unique configuration change subscription URI for each configuration data request, or MAY return the same SIP URI for different requests, so long as a change to the configuration data returned in any of these request results in notification on all subscriptions to the associated subscription URI.

If the CS returns a unique configuration change subscription URI in the Link header of different configuration data requests:

- The UA MUST maintain multiple subscriptions; one to each URI associated with configuration data the UA is using.

If the CS returns the same configuration change subscription URI in the Link header of different configuration data requests:

- The UA is not required to create multiple subscriptions to the same URI.

- The UA MUST associate the URI with each of the configuration data requests for which it was returned, and any NOTIFY or other change in the status of that subscription affects the validity of all of the associated configuration data.

- The CS MUST send a NOTIFY message on the configuration change subscription when there is a change to any of the different configuration data resources for which the subscription URI was returned.

2.5.1.1. Change Subscription Failure

If a configuration change SUBSCRIBE request (either the initial request or any attempt to refresh the subscription) is permanently rejected by the Configuration Service (the CS returns a failure response that is not an authentication challenge or redirection and does not specify a Retry-After header), the UA MUST consider the associated configuration data to be not valid and attempt to revalidate it as specified in Section 2.6.1, "Re-Validating Configuration Data". Since the CS is not allowed to reject a properly authenticated request, this indicates a problem either with the configuration data or the CS.

If a configuration change SUBSCRIBE request (either the initial request or any attempt to refresh the subscription) fails other than by being permanently rejected, the UA MUST consider the associated configuration data to be of unknown validity, and MUST retry the SUBSCRIBE request as specified in Section 2.7, "Retry Backoff Procedure"; the maximum time between retries MUST NOT be more than 30 minutes, and the retries MUST continue as long as the configuration is used. The UA MAY at any time return to any earlier step in the process of obtaining configuration data.

2.5.1.2. Change Subscription Termination

If the CS explicitly terminates the configuration change (http-monitor) subscription by sending a NOTIFY message with a Subscription-State header value of 'terminated', the UA MUST consider the configuration data to be of unknown validity. If the rules for interpreting and acting on the 'reason' code parameter as specified in Section 3.2.4 of [RFC3265] allow, the UA MUST attempt to re-establish the subscription. If those rules do not allow the UA to re-subscribe, then the UA MUST consider the data to be not valid and attempt to revalidate it as specified in Section 2.6.1, "Re-Validating Configuration Data". The UA MAY at any time return to any earlier step in the process of obtaining configuration data.

2.5.1.3. Configuration Change Notices

To inform the UA of a configuration data change, the CS MUST send a NOTIFY message to the UA in the configuration change subscription established by the UA as detailed in Section 2.5.1, "Configuration Change Subscriptions".

The CS MUST NOT send unsolicited (out-of-dialog) NOTIFY messages.

As specified in [RFC5989], the body of a NOTIFY message in the http-monitor event package is the HTTP headers that would have been returned in response to an HTTP HEAD request (a HEAD request returns the headers that would have been returned for a GET request to the same URI, but with no body).

When a NOTIFY message is received by the UA in the configuration change subscription, the UA MUST compare the cache control data it retained when the configuration data was received with the HTTP header values in the NOTIFY message body. If any of the cache control data in the HTTP header values differs from those in the original configuration data response, the UA MUST consider the stored configuration data to be no longer valid. As soon as reasonably possible after the UA discovers that configuration data is no longer valid, the UA MUST attempt a GET request to the HTTPS configuration request URL which provided the configuration data to obtain the changed configuration data.

If this HTTPS request to the URL that previously provided the configuration data fails, the UA MUST attempt to obtain a new URL as specified in Section 2.3, "Constructing the Configuration Request URL".

2.5.2. Configuration Change Polling

If the CS chooses to use the HTTP polling mechanism, it MUST NOT include a Link header with the relation 'monitor' in the HTTP configuration data response, and MUST include a Cache-Control header that specifies the max-age directive. The max-age cache control directive in HTTP specifies the maximum number of seconds for which the returned data may be cached; this specification defines this time as being the maximum time the configuration data is considered valid.

A short time before the validity time has passed, the UA SHOULD poll to revalidate the configuration data as described in Section 2.6.1, "Re-Validating Configuration Data".

2.6. Validity of Stored Configuration Data

Configuration data stored by a UA is considered valid:

- If the CS chose to use the subscription mechanism to deliver change notices, and the UA has a subscription to the CS as described in Section 2.5.1, "Configuration Change Subscriptions" on which no NOTIFY message from the CS indicating that the configuration data has changed has been received.

- If the CS chose to use the HTTP polling method, and the number of seconds since the configuration data response was received is less than the time specified by the Cache-Control max-age directive in that response.

When a UA initializes itself at any time other than immediately after receiving new configuration data, it MUST consider any stored configuration data to be of unknown validity.

The UA MAY use configuration data that is of unknown validity, or configuration data that is known to be no longer valid, while attempting to revalidate that data or obtain new data. There is no assurance that such configuration data is still useful, but the UA MAY choose to retain the data and to continue to use it.

2.6.1. Re-Validating Configuration Data

To revalidate stored configuration data of unknown validity, the UA MUST repeat the HTTPS GET request it used to obtain the stored configuration data, with the appropriate HTTP headers to make the request a conditional request using the cache control data returned in the response that provided the configuration data. This allows the CS to respond either with a new configuration data response or a 304 (Not Modified) response to indicate that the configuration data has not changed.

If the CS responds with a 304 response, and the original response included a Link header with the 'monitor' relation, the SIP UA MUST assume that the value of that Link header is also still correct (in effect, the HTTP cache control values and the subscription URL are a part of the configuration data), and so the UA MUST attempt to create and maintain a subscription to that URL as when the configuration data was first obtained (Section 2.5.1, "Configuration Change Subscriptions").

If the CS chooses to use the HTTP polling method, then any 304 response MUST include a Cache-Control header containing a max-age directive, and the UA MUST use this new value as the maximum validity time for the associated configuration data.

If the HTTP request to revalidate the configuration fails, the UA MUST follow the procedures defined for a failure of the initial HTTP configuration data request as specified in Section 2.4.2, "Configuration Data Request Failure".

2.7. Retry Backoff Procedure

In case of certain possible failures as described above, the appropriate response is to retry the failed operation. In all of these retry cases, the following rules apply:

- The UA SHOULD retry at least 5 times before abandoning the failed step (except as allowed for in specific error handling rules above).
- Following the first instance of a given failure, the UA MUST select an initial backoff timer value randomly between 2 and 8, inclusive, and wait this number of seconds before retrying the failed request.
- Following any subsequent instance of a given failure, the UA MUST increase the backoff timer value by 2 raised to the power of the number of preceding failures (2^N where N is the number of previous failures), and wait this increased number of seconds or the maximum interval specified by specific error handling procedures, whichever is less, before retrying the failed request.

For example, after an initial failure, the UA randomly chooses an initial backoff timer value of 4 seconds, followed by retries at the following times: 6 seconds (4 + 2^1), 10 seconds (6 + 2^2), 18 seconds (10 + 2^3), 34 seconds (18 + 2^4), and 66 seconds (34 + 2^5).

3. Configuration Data

This document does not specify the form or content of configuration data. As such, the contents of this section are non-normative.

3.1. Configuration Data Items

The configuration data for a SIP UA should, at minimum, include items with the following semantics.

3.1.1. Address-of-Record

The Address-of-Record (AOR) is a SIP or SIPS URI that identifies the user of the device as specified in [RFC3261].

3.1.2. Realm

The realm is used to populate the realm parameter in the SIP Proxy-Authorization header as specified in [RFC3261] when the UA receives an authentication challenge.

3.1.3. Username

The username is used to populate the username parameter in the SIP Proxy-Authorization header as specified in [RFC3261] when the UA receives an authentication challenge.

3.1.4. Digest

The digest is a string containing the digest of the username, realm, and password as specified in [RFC2617] and is used to generate a response to an authentication challenge as specified in [RFC3261].

3.1.5. OutboundProxy

The OutboundProxy if defined contains the default outbound proxy through which SIP requests, not explicitly routed, are routed as specified in [RFC3261].

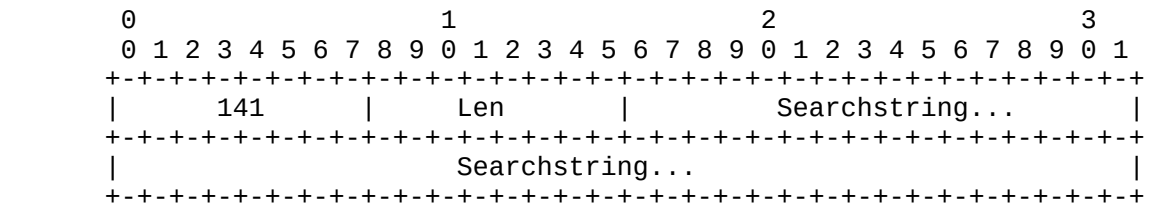
3.2. Reset User Agent to Default Configuration

The earlier sections of this document define methods by which the UA can be automatically provisioned. Some User Agents allow certain user specific settings (e.g., Contact Directory, specialized ring-tones, etc.) to be set by a user, and possibly stored locally in the User Agent. Since it may be necessary to later re-assign a UA, designers of configuration data formats may want to provide for explicit controls for any such locally configured settings, including the ability to explicitly delete them to return the UA to a completely unconfigured state.

4. IANA Considerations

4.1. DHCP SIP User Agent Configuration Service Domains Option

This specification defines DHCP option code 141, the "SIP UA Configuration Service Domains" for inclusion in the IANA registry "BOOTP Vendor Extensions and DHCP Options" defined by [RFC2939].



In the above diagram, Searchstring is a string specifying the searchlist. If the length of the searchlist exceeds the maximum permissible within a single option (255 octets), then multiple options MAY be used, as described in [RFC3396] "Encoding Long DHCP Options in the Dynamic Host Configuration Protocol (DHCPv4)".

To enable the searchlist to be encoded compactly, searchstrings in the searchlist MUST be concatenated and encoded using the technique described in Section 4.1.4 of [RFC1035], "Domain Names - Implementation and Specification". In this scheme, an entire domain name or a list of labels at the end of a domain name is replaced with a pointer to a prior occurrence of the same name. Despite its complexity, this technique is valuable since the space available for encoding DHCP options is limited, and it is likely that a domain searchstring will contain repeated instances of the same domain name. Thus, the DNS name compression is both useful and likely to be effective.

For use in this specification, the pointer refers to the offset within the data portion of the DHCP option (not including the preceding DHCP option code byte or DHCP option length byte).

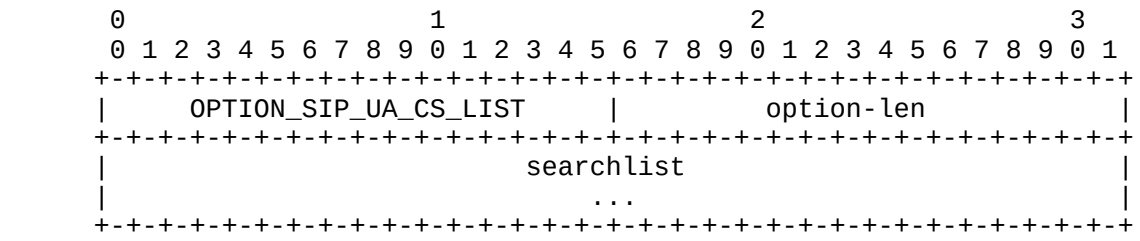
If multiple SIP UA Configuration Service Domains options are present, then the data portions of all the SIP UA Configuration Service Domains options are concatenated together as specified in RFC 3396, and the pointer indicates an offset within the complete aggregate block of data.

For examples of encoding this option, see Section 3 of [RFC3397], "Dynamic Host Configuration Protocol (DHCP) Domain Search Option", which uses the same encoding for option 119.

4.2. DHCPv6 SIP User Agent Configuration Service Domains Option

This specification defines DHCPv6 option code 58, OPTION_SIP_UA_CS_LIST, for inclusion in the IANA registry "Dynamic Host Configuration Protocol for IPv6 (DHCPv6), DHCP Option Codes" defined by RFC 3315.

The format of the SIP User Agent Configuration Service Domains option is:



option-code OPTION_SIP_UA_CS_LIST (58)

option-len Length of the 'searchlist' field in octets

searchlist The specification of the list of domain names in the SIP User Agent Configuration Service Domains

The list of domain names in the 'searchlist' MUST be encoded as specified in Section 8, "Representation and Use of Domain Names" of RFC 3315.

4.3. U-NAPTR Registration

This document registers the following U-NAPTR application service tag in the registry defined by [RFC3958], "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)":

Application Service Tag	SFUA.CFG

This tag is used to obtain the base URL of the Configuration Service from the DNS name of a SIP domain as specified in Section 2.3.1, "Obtaining a Configuration Service Base URL".

4.4. SIP Forum User Agent Configuration Parameter Registry

IANA has established a registry for "SIP Forum User Agent Configuration Parameters". This registry records the HTTPS request parameters for the initial configuration data request sent by a User Agent to a Configuration Service as described in Section 2.3.2, "Adding Configuration Request Parameters".

Each entry in the registry must include the Parameter Name and a Description that specifies the value syntax and usage of the parameter:

Parameter Name The name of the parameter, which MUST match the ABNF production for 'token' from [RFC3261].

Value Syntax The syntax of the value, if any (a parameter may just be a name with no associated value).

Usage The purpose served by the parameter, including any default method the UA should use to construct it if applicable.

The initial values for the registry are the parameters described in Section 2.3.2.1, "Configuration Request Parameters". The policy for future additions to this registry depends on the parameter name value:

If the name of the parameter begins with the characters 'sfua-' in any case, then the policy for addition to this registry is "RFC Required" as described by [RFC5226].

Any other parameter entry may be added to this registry using a "First Come First Served" policy as described by [RFC5226].

5. Security Considerations

Initial discovery of the Configuration Service Domain name relies on a number of operations that are normally unsecured: a DHCP response could be provided by an attacker to replace the values of any of the IP Network Parameters (Section 2.1.2, "Network Layer Provisioning") including the Local Network Domain which is the default choice for the Configuration Service Domain name. Confirmation by the human user of the Configuration Service Domain name, especially when it differs from a previously used value, could be used to mitigate this (perhaps unintentional) potential reconfiguration. Note that previously loaded configuration MAY constrain which parts of the discovery and location procedures are used: for example, the Configuration Service Domain name might be fixed so that it cannot be modified by discovery.

The connection to the Configuration Service is made over TLS. As the TLS server, the CS always provides a server certificate during the TLS handshake; if possible, the UA should validate that certificate and confirm that it contains as a subject the Configuration Service Domain name or at least the host name from the Configuration Service Base URL (see [RFC2818]). While it may not be possible to have the

information needed to perform a full validation of the CS server certificate prior to the first configuration (for example, the UA may not have a current CA certificate for the CA that signs the CS server certificate), implementors are advised to provide that information in configuration data so that it can be used for subsequent reconfigurations; this narrows the window of vulnerability to the first configuration attempt.

To secure initial configuration attempts, the CS can deny requests from unknown devices and/or could implement other measures such as restricting the time window during which it will accept an initial configuration request from a given device. A more secure approach would be to provide the user with a password, perhaps a one-time password valid only for the initial access. In high security environments, the Configuration Service could require that the User Agent provide a client certificate for authentication in the TLS connection for configuration data requests. This would necessitate some prior manual configuration of the User Agent, and possibly the Configuration Service, and that configuration should also include sufficient information for the UA to fully validate the CS certificate.

The values of some or all of the request parameters sent by the UA on the initial request for configuration data (see Section 2.3.2, "Adding Configuration Request Parameters") may be sensitive information. Since the configuration data request is made over a TLS connection, the confidentiality of that information is protected on the network. Configuration Service implementations should take all necessary measures to ensure that the request parameter data is appropriately protected within the CS itself.

The Configuration Change Request Subscription (Section 2.5.1, "Configuration Change Subscriptions") is established only after the configuration data has been loaded by the User Agent, so all security mechanisms available in SIP (including request digest authentication and the use of TLS) can be configured and required by either the CS or the UA. Note that a configuration change notice does not actually provide any new configuration data, nor can it change where the UA sends a request for the new configuration data. This means that an attacker cannot reconfigure a UA by subverting only the change notice subscription; the most the attacker can do is trigger checks for new data. In order to actually modify the configuration data itself, the attacker must subvert the CS or the steps leading to the CS discovery (subject to the checks described above).

Implementations of TLS typically support multiple versions of the Transport Layer Security protocol as well as the older Secure Sockets Layer (SSL) protocol. Because of known security vulnerabilities, SIP

UAs, SIP Service Provider, and the Configuration Service Host MUST NOT request, offer, or use SSL 2.0. See Appendix E.2 of [RFC5246] for further details.

6. Acknowledgements

Contributing Members of the SIP Forum User Agent Configuration Working Group:

Francois Audet, Nortel Networks, Inc.

Eric Burger, SIP Forum

Sumanth Channabasappa, Cable Television Laboratories, Inc. (CableLabs)

Martin Dolly, AT&T Labs

John Elwell, Siemens Enterprise Communications

Marek Dutkiewicz, Polycom, Inc.

Andy Hutton, Siemens Enterprise Communications

Lincoln Lavoie, University of New Hampshire

Scott Lawrence, Avaya, Inc.

Paul Mossman, Avaya, Inc.

Michael Procter, VoIP.co.uk

Marc Robins, SIP Forum

Henning Schulzrinne, Columbia University

Rifaat Shekh-Yusef, Avaya, Inc.

Robert Sparks, Tekelec

Simo Veikkolainen, Nokia

The Editor would like to also acknowledge valuable contributions by Leslie Daigle and Margaret Wasserman.

7. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC2939] Droms, R., "Procedures and IANA Guidelines for Definition of New DHCP Options and Message Types", BCP 43, RFC 2939, September 2000.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3319] Schulzrinne, H. and B. Volz, "Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers", RFC 3319, July 2003.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, November 2002.

[RFC3397] Aboba, B. and S. Cheshire, "Dynamic Host Configuration Protocol (DHCP) Domain Search Option", RFC 3397, November 2002.

[RFC3958] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4366] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 4366, April 2006.

[RFC4848] Daigle, L., "Domain-Based Application Service Location Using URIs and the Dynamic Delegation Discovery Service (DDDS)", RFC 4848, April 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, October 2009.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5989] Roach, A., "A SIP Event Package for Subscribing to Changes to an HTTP Resource", October 2010.

[ANSI.TIA-1057-2006]
American National Standards Institute, "Telecommunications IP Telephony Infrastructure Link Layer Discovery Protocol for Media Endpoint Devices", April 1993.

Authors' Addresses

Scott Lawrence (editor)
Linden Research, Inc.

EMail: scott-ietf@skrb.org

John Elwell
Siemens Enterprise Communications

Phone: +44 1908 817801
EMail: john.elwell@siemens-enterprise.com