

Core Concepts of the Residential Portal

This chapter provides the background knowledge required by developers of the residential portal, also known as the service selection portal (SSP). After understanding these concepts, portal developers can create Web pages that manipulate the core SAE functionality through the portal API.

Topic	Page
Users	1-1
Services	1-4
Subscriptions	1-5
Subscriber Schedules	1-8
Token and Public IP Addresses	1-12
Managing HTTP Requests to Unauthorized Web Resources	1-15
Managing Security for Public Wireless LAN Applications	1-18

Users

A user is a person who obtains network connectivity and network services by connecting a PC or other client device to an edge router managed by an SAE.

Loading Users at Login Time

Every user has a *user profile* stored in an LDAP directory. In the LDAP directory, each user profile is located below a retailer entry. Each retailer entry can contain several domain names (for example, isp3.com, isp4.com, or howie.com).

When a user logs in to the SAE as *john@isp3.com*, the SAE searches the directory for a retailer entry that contains the domain name *isp3.com*. It then invokes a *UserClassification script*, which returns an LDAP query. This LDAP query is executed and if it returns exactly one user profile, the SAE loads that user profile into memory and associates it with the user's current IP address.

If a user logs in without a domain name, then the SAE assumes that the domain name is the name of the virtual router (VR) through which the user connects. The SAE then searches in the directory for a retailer that defines this VR name as its domain name, and if it finds one, proceeds to search for a user profile.

However, if the SAE cannot find in the directory a retailer that defines this VR name as its domain name, the SAE assumes a domain name of default. The SAE then searches for a retailer which defines "*" as its domain name and tries to match the user with that retailer.

PPP User Logins

Users who connect using PPP must enter their login name at their PC before connecting to the network. This PPP login name is passed from the edge router to the SAE automatically. The SAE then loads a user profile from its LDAP directory based on that login name.

PPP User Logouts

Users who connect using PPP are automatically logged in to the SAE based on their PPP login name when they establish their PPP connection. They are automatically logged out of the SAE when that PPP connection is shut down.

DHCP User Logins

Users who connect using DHCP do not provide a login name before connecting to the network. The SAE is automatically informed of their existence, but not of their identity. Until the SAE learns their identity, such users are considered *unauthenticated* users (see *Unauthenticated User*).

To inform the SAE of an unauthenticated DHCP user's identity, a JSP page must prompt the user for a login name and password, and then pass that login name and password to the portal API (that is, to the *loginUser* method of the *Ssp* class), along with the user's current IP address. The SAE then loads the proper user profile from its LDAP directory and

assigns it to the DHCP user. The user profile that SAE loads is specified by the login name, as described in *Loading Users at Login Time*.

DHCP User Logouts

In addition to a JSP page for logging in DHCP users, a portal can also include a JSP page for logging out DHCP users. The logout page must invoke the portal API by calling the *logoutUser* method of the *Ssp* class. This discards the specified user's current user profile, and replaces it with the unauthenticated user profile (see *Unauthenticated User*).

Unauthenticated User

The SAE configuration GUI requires that one user profile in the LDAP directory be designated as the *unauthenticated* user profile. After a DHCP user connects to the network, and before the user logs into SAE through a portal JSP page, the user is assigned a copy of the unauthenticated user profile. When the JSP page calls the *loginUser* method of the *Ssp* class, the unauthenticated user profile is discarded, and the user is assigned a specific user profile loaded from the LDAP directory based on the login name.

Registering a DHCP Login

A JSP page can *register* a DHCP user's login. After doing so, the user is automatically logged in every time he turns on the PC or other client device. A login is registered by passing to the *registerLogin* method of the *Ssp* class the desired login name and password, as well as the *MAC address* that uniquely identifies the user's PC. For currently logged in users, the MAC address is available from the Portal API's user Bean.

This MAC address and a reference to the user profile specified by the login name is stored in the SAE's LDAP directory. The next time the user turns on the PC (that is, issues a DHCP discover), the SAE looks up the PC's MAC address in its LDAP directory, and retrieves the associated user profile.

A JSP page can obtain a list of all the registered logins that use a specific login name by calling the *getRegisteredLogins* method of the *Ssp* class.

Unregistering a DHCP Login

A JSP page can later *unregister* a DHCP user's login. After doing so, the user receives the *unauthenticated* user profile every time he turns on his PC or other client device (that is, issues a DHCP discover). A login is

unregistered by passing to the *unregisterLogin* method of the *Ssp* class the MAC address of the user's PC.

Registering and unregistering logins does nothing except write information to and delete information from the SAE's LDAP directory. Nothing else is affected until the next time the client device with the specified MAC address requests a new IP address (that is, it issues a DHCP discover).



Note: Registering or unregistering a login has no immediate effect. It only has an effect the next time the user's PC requests a new IP address.

Anonymous Users

An *anonymous user* is a user who is logged into the SAE with an anonymous user profile. An anonymous user profile is a user profile that has its anonymous flag in the LDAP directory set to TRUE. When the anonymous flag is set to TRUE, then no user can modify the user profile or subscriptions. Anonymous user profiles are intended to be shared by many users, where each user gets a personal copy of the anonymous user profile. For example, this can occur when the user name and password for an anonymous user profile is widely published.



Note: From the portal developer's perspective, an anonymous user is a just like any other user. In particular, the anonymous user can activate and deactivate subscribed services just like any other user.

Services

An *SSP service* represents a network resource to which access can be controlled. A video server that streams movies to end-users is one example of a possible service. There are two types of SSP services:

- Normal services
- ISP services

ISP services are explained in the section *Token and Public IP Addresses*.

Normal Services

When a normal service is activated for a given user, service-specific policies are sent to the router that controls the user's network traffic. The policies are designed by network specialists and stored in the SAE's LDAP directory. The policies are usually designed to grant the user access to a network resource, such as a video server. They can also enforce a

given quality of service for the traffic between the user and the network resource, or redirect the user's network traffic in sophisticated ways.

When a service is deactivated, the service-specific policies are removed from the router that controls the user's network traffic. This usually means the user will lose access to the associated network resource.

Subscriptions

Before a user can activate a service through a JSP page, he or she must be subscribed to that service. Subscriptions are recorded in the user profile in the SAE's LDAP directory. After a user logs in, the names of all of his or her subscribed services are available through the portal API.

Activating and Deactivating Subscribed Services

A subscribed service can be activated and deactivated manually, on login or logout, or according to a schedule.

A manual activation or deactivation occurs when a JSP page, reacting to a user request, calls a method on a portal API subscription Bean.

Subscriptions that are to be activated when a user logs into the SAE and deactivated when the user logs out of the SAE must be marked accordingly in the SAE's LDAP directory. The portal API provides a way to mark subscriptions as activate-on-login subscriptions.

For information on activating and deactivating subscribed services according to a schedule, see *Subscriber Schedules* later in this chapter.

Multiple Subscriptions per Subscriber

Subscribers can have multiple simultaneous subscriptions to the same service. Each subscription must have its own parameter substitutions and can be activated or deactivated independently.

An object for each subscription is created in the directory. The name of the object has the following format:

<ServiceName>%<SubscriptionName>

<ServiceName> – name of the service

<SubscriptionName> – name of the subscription

Other than the naming convention, multiple subscriptions are identical to regular subscriptions.

Multiple Service Sessions per Subscription

A user can have multiple simultaneous service sessions for each subscription. In addition, different parameter substitutions can be associated with each service session.

A user's subscription to a service can have multiple service sessions:

- Each session is identified, relative to its subscription, by a unique session name
- Each session can have its own unique substitutions for service policies. In the substitution acquisition path, the session's substitutions are the most specific. Session substitutions override all other substitutions for the same variable unless the variable is fixed.
- Each session can be activated and deactivated independently. Doing so applies or removes policies from an E-series router interface in the normal manner, and generates accounting data in the normal manner.

This functionality is exposed by the `sessionName` property in the subscription Bean. By setting the `sessionName` property, the subscription Bean is initialized to represent a specific new or existing session. Additionally, the subscription Bean has a method to get the list of all active session names.

Persistent Service Sessions

You can customize the SAE to allow residential subscribers to mark persistent sessions associated with normal SSP services that do not require authentication (see *Services*, earlier in this chapter) and *Activations Requiring Authentication*, later in this chapter.) Persistent service sessions correspond to services that are always active. If the subscriber loses the connection to the network, the SAE will restore the appropriate service when the subscriber reconnects.

You cannot customize the SAE to associate persistent sessions with ISP Services or SSP services that require authentication. You will see an error message if you try to customize the SAE to associate persistent sessions with these services.

The SDX software creates one object that holds all the persistent service sessions for a subscriber, whether the subscriber uses a shared profile (see *Anonymous Users*, earlier in this chapter) or an individual profile. The key to the object is the subscriber's `loginName`—the name with which the subscriber logged into the subscriber session.



Caution: When you offer this feature to subscribers, be sure that the directory has sufficient capacity to store objects for all residential subscribers, including those who use shared profiles rather than individual profiles.

In the directory, the SDX software stores objects for persistent service sessions in a dedicated cache in the folder *ou=retailerName, o=PersistentSessions, o=umc*. Because the SDX software stores these objects separately from subscriber profiles, residential subscribers who use a shared profile can mark persistent sessions for subscriptions received through that profile. Similarly, residential subscribers can mark persistent sessions for subscriptions inherited from a parent object.

This feature offers particular benefits to subscribers who use shared profiles. For example, the SDX configuration might assign a shared profile to subscribers for whom the RADIUS application returns the same value for the *serviceBundle* object. Some of the subscribers who use this shared profile may want continuous access to a video service. If you offer the persistent session option with this video service, these subscribers can obtain continuous access to it.

The feature also offers benefits to subscribers who inherit subscriptions from parent objects in the hierarchy. For example, if a retailer subscribes to a video service, all residential subscribers associated with that retailer have access to that service. Some of these subscribers may want continuous access to a video service. If you offer the persistent session option with this video service, subscribers can obtain continuous access to it.

Parameter Substitutions

Substitutions can be specified for any session of a subscription. These substitutions are considered most specific. Session substitutions override all other substitutions for the same variable unless the variable is fixed. If substitutions are specified in a subscription Bean before setting the *sessionName* property, the substitutions affect the default session only.

Activations Requiring Authentication

You can specify that a residential subscriber must enter a username and password before the SAE activates a particular SSP service. When a residential subscriber requests activation of a subscribed service, the JSP page must query the Portal API to determine if the subscribed service requires authentication. If it does, the JSP page must obtain a username and password from the residential subscriber, and pass them to the Portal API before activating the service.

Subscriptions to services that require authentication can never be marked as activate-on-login subscriptions, since they require the subscriber to manually supply a username and password after logging in and before the service is activated. Similarly, subscribers cannot mark service sessions associated with these services for persistent activation (see *Persistent Service Sessions*, earlier in this chapter).

Usage Data

The service-specific policies that affect a user's network traffic while a subscribed service is activated can generate accounting data. For example, if a service defines policies that allow network traffic to flow from a video server to the user, then those policies can also collect information about the amount of such traffic, measured in bytes and IP packets.

This information, along with the length of time that a subscribed service has been active, is available from the portal API's subscription Bean.

Subscriber Schedules

Using the residential portal, subscribers can create schedules, which consist of actions that are performed on a specified schedule. They can activate or deactivate a service at a particular time or on a recurring schedule, or they can deny service activation during particular hours.

There are two types of actions:

- Event action – The SAE activates or deactivates a service at the scheduled time. For event actions you specify the action time.
- Authorize action – The SAE authorizes or denies an action based on the time that the service is requested. For example, a subscriber may not want to allow a service to be accessed between the hours of 1700 and 2000. In this case the subscriber would set up a schedule that denies activation of the service during that time period. In addition to denying access to a service, the action can be configured to deactivate current sessions that use the service at the scheduled time. For authorization actions, you specify a time period (start time to end time).

Object of Schedules

Subscriber schedules apply to subscription and session objects and are a property of subscribers, not services. The schedule is created independent of the subscription or session object.

The schedules are persistent and are stored in the directory server. During SAE startup, the subscriber schedule is retrieved from the directory server. As a result, the schedule is maintained after an SAE failover, start, or restart event.

Subscribers can view their schedule and delete or modify schedule entries. The subscriber must be online when setting a schedule.

Setting the Time Pattern

The time pattern format used to create schedules is modeled on the UNIX crontab entry format. It consists of seven fields separated by space or tabs.

Table 1-1 describes the fields in the time pattern. When users set these fields in the portal, they can use the following:

- * – asterisks are interpreted as follows:
 - > Minutes and hours – 0 (zero)
 - > Time zones – local SAE time zone
 - > All other fields – first through last

A value of * (asterisk) for the end time is equivalent to “deny service activation after this start date,” whereas a value of * for the start time is equivalent to “deny service activation before this end date.”

- Letters for day of week or month. Use the first three letters of the day or month; for example, mon, tue, or jan, feb.
- Range of numbers or letters separated by a hyphen. The range is inclusive; for example, 1-5 for the hour specifies hours 1, 2, 3, 4, and 5. A range of mon-wed specifies Monday, Tuesday, and Wednesday.
- List of numbers, letters, or ranges separated by commas. For example, 1,2,5,9 or 0-4,8-12 or mon-wed,fri-sat.
- Step values with ranges. To skip a number’s value through the range, follow a range with /<number>. For example, 0-23/2 used in the hours field specifies that the event occurs every other hour.
- Step values with *. You can also use step values with *; so if you want to specify every two hours, use */2.

If you set both a day of the month and a day of the week, the day of the month is used.

Table 1-1 Time pattern format

Field	Description
Year	Four digits that indicate the year
Month	Month of the year (1-12 or first three letters of the month)
Day	Day of the month (1-31)
Hour	Hour of the day (0-23)
Minute	Minutes past the hour (0-59)
DOW	Day of the week (0-6 with 0=Sunday, or first three letters of the day)
TZ	Name of the time zone; for example, Canada/Eastern or America/New York. You can also set a customized time zone. An * indicates the local time zone of the SAE. To set a time zone as an offset to GMT, use the format: GMT (+ -) (hh:mm hh mm hh) hh=hour mm=minute For example, GMT+5 sets the time zone to 5 hours behind GMT.

One-Time Events and Recurring Events

The SDX software provides much flexibility in how it allows schedules to be specified. For example, you can set up:

- A one-time event – Performs the action at a specified time.
- A recurring event – An event can recur over a period of time at given time intervals, such as a schedule that activates gold service at 6:00 PM every evening.
- A working-hours service – For example, a service that is activated Monday through Friday at 8:00 AM and deactivated Monday through Friday at 5:00 PM. This type of service requires two schedule entries—one that activates the service and one that deactivates the service.

Specifying Threshold or Preparation Times

You can build delay or preparation times into schedules so that subscribers have flexibility in the time they need to perform a task, and the system has time to complete a transition from one state to another.

SAE uses preparation and delay times with subscriber schedules to provide flexibility in processing the actions of the schedule. The threshold time and the preparation time are configured globally for the SAE server.

Action Threshold

The action threshold indicates the maximum delay that a service allows for a time-related change to occur. For example, you can allow a 15-minute delay so that if an event is scheduled for 5:00 but the system is not able to perform the event at 5:00 or the subscriber is not logged in at 5:00, the SAE attempts to perform the action for the subscriber until 5:15.

Preparation Time

Because the transition from one state to another does not occur instantaneously, SAE uses a preparation time to allow for the time that the SAE needs to make the transition. For example, if you have a pay-per-view service and many subscribers need to have the service activated by a certain time, you can configure the preparation time to begin the process early to make sure that everyone gets his or her service activated by the time the event starts. Or you could schedule a few minutes of preparation time for setting up a video conference.

Setting the Action Threshold and Preparation Time

To set the action threshold and preparation time for an SAE:

- 1 In the SDX Admin navigation pane, access the configuration object in I=SAE, ou=staticConfiguration, ou=configuration, o=management, o=UMC.

The SSP Configuration pane appears.

- 2 Add one or both of the following properties to the Property field:

DelayedActions.PreparationTime=<Value>

DelayedActions.ActionThreshold=<Value>

Specify the value in milliseconds. For example, to set the preparation time to five minutes, add the following property:

DelayedActions.PreparationTime = 300000

Token and Public IP Addresses

This section applies to users connected through DHCP, and not to users connected through other access methods.

When a DHCP user connects to the network, his PC or other client device is uniquely identified by its hardware's MAC address. This MAC address is available through the Portal API.

When the PC is turned on or rebooted, it usually issues a DHCP discover message, which is a request for an IP address. In response, the router that the SAE manages assigns it an IP address. Once it has an IP address, it can communicate with other devices on the network.

The DHCP server in the E-series router distinguishes between token and public IP addresses. Token IP addresses are assigned to a PC without authenticating the DHCP request. Public IP addresses require a successful authentication of the request.

The SAE can force the PC to switch from an unauthenticated Token IP address to an authenticated public IP address. This feature is especially useful when the owner of the SSP provides access to several ISP retailers, each of which controls a separate pool of public IP addresses. The first time that a user connects to the network through DHCP, his PC is assigned an unauthenticated or token IP address.

Granting a Public IP

By calling the *grantPublicIp* method of the Portal API, a JSP page can tell the SAE to switch a user from having a token IP address to having a public IP address.

When an IP address is assigned to a user's PC, it is assigned for a specific period, known as the IP address lease time. *This lease time is configured on the E-series router, and can range from about half a minute to days or months.* The user's PC must continually renew this lease to keep its IP address. For example, if the lease time is 30 seconds, then the user's PC has to issue a *DHCP renew* before the current lease expires (typically 15 seconds). If the E-series system refuses to renew the IP address, then the user's PC loses all connectivity to other computers on the network, except to issue *DHCP discovers*, which are requests for a new IP address.

When a JSP page calls the *grantPublicIp* method, the E-series router stops renewing the user's token IP address lease. The next time the user's PC tries to renew its token IP address, it is refused. It loses its IP address and then must ask for a new address. When it asks for a new address the E-series router uses the credentials sent in the */grantPublicIp/* method to authenticate the request and grants a public IP address, if the authentication succeeds. This process can take a while, up to the token IP lease time.



Note: See *E-series router documentation* for details on how IP addresses are managed.

Since the different pools of public IP addresses are often controlled by different organizations, such as ISPs, the JSP page must obtain a password from the user and pass it to the Portal API's *grantPublicIp* method.

Revoking a Public IP

By calling the *revokePublicIp* method of the Portal API, a JSP page can tell the E-series router to switch a user from having a public IP address to having a token IP address.

When a JSP page calls the *revokePublicIp* method, the E-series router stops renewing the user's public IP address lease. The next time the user's PC tries to renew its public IP address, it is refused. It loses its IP address and then asks for a new address. When it asks for a new address the E-series router grants it a token IP address. This process can take a while up to the public IP lease time (which is usually significantly longer than the token IP address lease time).



Note: Every time a user's IP address is changed, all of his currently active services are automatically deactivated.

Registering Equipment

A JSP page can *register* the granting of a public IP address. After doing so, the user's PC or other device (that is, the user's equipment) is automatically granted a public IP address every time it is turned on. Equipment is registered by passing to the *registerEquipment* method of the *Ssp* class the same login name and password as is passed to the *grantPublicIp* method, along with the *MAC address* that uniquely identifies the user's PC. For currently logged in users, the MAC address is available from the Portal API's user Bean.

This MAC address and login name and password are stored in SAE's LDAP directory. The next time the user turns on his PC (that is, issues a DHCP discover), the SAE looks up the PC's MAC address in its LDAP directory and retrieves the associated login name and password. If the E-series router can authenticate these credentials, a public IP address is handed out from the appropriate pool of IP addresses.

A JSP page can obtain a list of all the equipment registered by a specific user by calling the *getRegisteredEquipment* method of the *Ssp* class.

Unregistering Equipment

A JSP page can later *unregister* the granting of a public IP address. After doing so, the user's PC is assigned a token IP every time it is turned on (that is, every time it issues a DHCP discover). Equipment is unregistered by passing its MAC address to the *unregisterEquipment* method of the *Ssp* class.

Registering and unregistering equipment does nothing except write information to and delete information from the SAE's LDAP directory. Nothing else is affected until the next time the client device with the specified MAC address requests a new IP address (that is, issues a DHCP discover).



Note: *Registering or unregistering equipment has no immediate effect. It will only have an effect the next time user's PC requests a new IP address.*

ISP Subscriptions

ISP services behave just like normal services (see *Normal Services*) in that they apply policies that affect a user's network traffic when activated, and remove those policies when deactivated.

However, when a user activates an ISP subscription, the user is granted a public IP address, and when he or she deactivates the ISP subscription, his or her public IP address is revoked.

ISP services always require that a login name and password be supplied before they are activated (see *Activations Requiring Authentication*). As a result, a user's subscription to an ISP service can never be marked as an activate-on-login subscription.

ISP services provide another way to switch a user's IP address. To avoid confusion, it is strongly recommended that a JSP portal use either the activation and deactivation of ISP services or the *grantPublicIp* and *revokePublicIp* methods, but not both.

Implicit ISP Subscriptions

As described in *Users*, user profiles are stored in the SAE's LDAP directory under retailer entries. If a given retailer entry contains the name of an ISP service, then every user stored below that retailer is implicitly subscribed to the named ISP service.

An implicit ISP subscription is unique in that it is always activated on login, whereas normal ISP subscriptions are never activated on login.

The login name and password used to automatically activate the implicit ISP subscription on login are taken from the user's profile in the SAE's LDAP directory.



Note: *If you have a choice between using implicit ISP subscriptions and using equipment registration, always choose equipment registration. Implicit ISP subscriptions are much less efficient than equipment registration. Every time a user with an implicit ISP subscription turns on the PC and logs in to the SAE, the user is first assigned a token IP address and then immediately switched to a public IP. In contrast, a user with registered equipment who turns on the PC is immediately assigned a public IP address, without first having to obtain and lose a token IP.*

Managing HTTP Requests to Unauthorized Web Resources

The SDX software employs the captive portal feature to intercept HTTP requests sent from a client to an unauthorized Web resource and redirect the requests to a dedicated Web page, the captive portal page.

For example, you might want to capture HTTP requests to a content provider network when the corresponding content service has not been activated for that subscriber. The subscriber would be taken to the captive portal page, which might simply display an error message, or which could prompt the subscriber to log in to or activate the requested service.

The captive portal requires four components:

- Default policies installed on the E-series router. The default policies on the E-series router must include a forwarding or rate limiting policy that permits access to the portal server and a next-hop rule to intercept the unauthorized access request packets. The target of the next-hop rule is the host on which the redirect engine resides; this host is typically the same host where you install the SAE.
- An instance of the redirect engine (the *UMCredir* package) installed on a host in the same network as the E-Series router. The host must be no more than one hop away from the router.

- The IP Filter tool redirects incoming HTTP requests to the redirect engine. It must be installed and configured on the same host as the redirect engine.

You configure the IP Filter `/etc/opt/ipf/ipnat.conf` file to specify what traffic is redirected and where it goes. You'll need a rule to direct unauthorized traffic. For convenience, you may include a rule that directs authorized traffic for the SAE host. You can have more rules depending on how you want to manage subscriber traffic. Here is an example of a rule to redirect unauthorized traffic:

```
rdr hme0 0.0.0.0/0 port 80 > 10.227.1.163 port 8800 tcp
```

See *SDX Installation and Configuration Guide, Chapter 3, Initial Configuration of the SDX Software* for more information on IP Filter; the UNIX man pages for **ipnat** and **ipf** may also be helpful.

- A portal server serving the captive portal pages. In a demonstration or test environment this would be the SSP demo portal. In a production environment, it would be your modified version of the SSP that is deployed to the subscribers.

Sequence for Redirecting Traffic

Suppose a subscriber opens a Web browser and attempts to access the restricted server `http://a.com`. A next-hop policy on the E-series router sends this request to the SAE instead of to the requested server. The policy does not affect the destination address (resolved from `a.com`), in the IP packets.



Note: Your network configuration must not have any routers between the E-series router and the SAE. An intermediate router would look at the destination address for `a.com` that is still present in the packets and would route the packets there rather than to the SAE.

The IP Filter process running on the SAE host filters and redirects traffic arriving on port 80 on the SAE's incoming interface. The captured request is redirected to an address and port where the redirect engine listens.

The redirect engine opens a TCP port (8800 by default) and sends an HTTP 302 (found) response to the subscriber's browser for the requests. The portal server then displays the captive portal page in the subscriber's browser.

Redirect Engine Redundancy

Redirect engine redundancy works in the same way as SAE redundancy. You install the redirect engine software on two different hosts; then you

configure one redirect engine as the primary redirect engine, and the other as the redundant redirect engine. The active and redundant redirect engines regularly poll each other to confirm each other's availability. If the primary redirect engine becomes unavailable, the redundant engine assumes the active role.

To ensure that clients can always reach the active redirect engine, both engines must share a virtual IP address in the DNS. When a redirect engine assumes the primary role, it configures on the router a static route from the virtual IP address to the engine's real IP address. Clients send requests to the virtual IP address, and the router automatically sends the request to the active redirect engine via a static route.

Configuring the Redirect Engine

To configure the redirect engine:

- 1 Install the redirect engine software as follows:
 - If you want to configure redundancy for the redirect engine, install the software on two hosts.
 - If you do not want to configure redundancy for the redirect engine, install the software on one host.
- 2 (For redundancy only) In the DNS, configure the hosts to share one IP address, which differs from the actual IP address of each host.

The DNS should also contain entries that map the hosts to their actual IP addresses.
- 3 On each host on which you installed the redirect engine software, access the directory in which you installed the redirect engine, and run the configuration script.

```
# cd /opt/UMC/redirect
# etc/config
```
- 4 Follow the instructions on the screen to configure the redirect engine.

If you are configuring redundancy for the redirect engine, assign one redirect engine as the primary engine, and the other as the redundant engine.

Protection Against Denial-of-Service Attacks

The redirect engine incorporates a number of properties to protect against denial-of-service attacks (default values shown):

- The redirect engine can serve no more than 12,000 requests per minute, with a burst of 18,000 requests.
- The redirect engine can serve no more than 25 requests per client per minute, with a burst of 50 requests.
- Incoming requests can be no larger than 4 KB.
- Incoming requests have a time limit of 2 seconds.

Edit the `/opt/UMC/redis/lib/RewriteServer.py` file to modify the values for any of these properties.

Logging

The redirect engine logs incoming HTTP requests via the UNIX **syslog** command with a priority of INFO and log-facility of LOCAL7. See *SDX Administration Guide, Appendix G, Configuring Logging* for information on system logging.

Managing Security for Public Wireless LAN Applications

You can include in the residential portal a Web page that automatically refreshes itself and provides a keep-alive application that verifies the HTTP session. If the keep-alive application cannot verify the HTTP session, the portal terminates the subscriber session. This feature improves security for public wireless LAN applications.

If you include this Web page in the residential portal, the following sequence of events occurs:

- 1 When a subscriber logs in through the portal, the SDX software starts the keep-alive application.
- 2 The keep-alive application creates a session key and sends it to the residential portal.
- 3 The residential portal stores the session key in its corresponding HTTP session.
- 4 The keep-alive application sets the timeout for the subscriber session to a value greater than the refresh time.
- 5 When the Web page refreshes itself, the keep-alive application sends the session key to the residential portal.
- 6 The portal responds as follows:

- If the session key matches the value in the portal's HTTP session, the portal updates the timeout for the subscriber session, creates a new session key, and sends the new key to the keep-alive page.
 - If the session key does not match the value in the portal's HTTP session, the portal terminates the subscriber session.
- 7 If the Web page does not refresh itself before the timeout expires (for example, if the subscriber closes the Web browser or turns off the PC without logging out), the portal terminates the subscriber session.

