

Chapter 13

Enabling SDX Actions from IDP Manager

This chapter describes how to complete IDP integration with IDP by configuring a script that the IDP Manager uses. This chapter contains the following sections:

- Overview of How to Enable SDX Actions from IDP Manager on page 195
- Configuring SDX Scripts on page 195

Overview of How to Enable SDX Actions from IDP Manager

After you complete all the configuration to integrate IDP with the SDX software, you configure the SDX **idpsdx.py** script—a script that implements the messaging to record problem incidents and identifies the action for the SDX software to take: redirect traffic to an IDP captive portal, activate services, and send e-mail.

In a testing environment, you can use the **idpsdx.sh** script to set up and troubleshoot a configuration that integrates IDP into an SDX-managed environment. The **idpsdx.sh** script sets the library paths, redirects debugging output, and executes the **idpsdx.py** script. Do not use the **idpsdx.sh** script in a production environment.

The **idpsdx.py** script requires Python version 2.3 and the following Python libraries installed on the system that runs the IDP management server:

- SOAPpy (0.11.6)
- PyXML (0.8.3)
- fpconst (0.7.0)
- logging (4.8.1)

The SMCpython and UMCpyadd packages in the SDX software distribution contain Python version 2.3 and the libraries listed.

Configuring SDX Scripts

The **idpsdx.py** script provides documented source code as well as configuration properties to allow you to create customized e-mail messages and implementations. You can locate the scripts in the */opt/UMC/idp/scripts* directory.

Before You Configure Scripts

Before you configure scripts:

- Complete all other configuration for IDP integration with SDX.
- Verify the location where Python is installed on the system. If you installed Python from the SDX software distribution, the default installation directory is */opt/UMC/python*. If you installed Python to a different directory, update the paths in *idpsdx.py* and in *idpsdx.sh* (if you use this file).
- For a production environment, start the IDP management server in an environment in which the library path includes the Python libraries.

Configuring Scripts

To configure SDX scripts:

1. Edit the *idpsdx.py* file to specify the actions to be taken.

See *Properties in the idpsdx.py File* on page 196.

2. Copy the *idpsdx.py* file and the *idpsdx.sh* file (if you use this file) to the appropriate directory for IDP Manager. For the location of this directory, see the IDP documentation at

<http://www.juniper.net/techpubs/software/management/idp/>

Properties in the idpsdx.py File

You can modify the following properties in the *idpsdx.py* file.

RECORD URL

- URL of the record interface of the IDP captive portal that stores information received from IDP. The interface records information about detrimental traffic under the source and destination of the traffic. The security rules configured in IDP determine the type of incidents recorded.
- Value—“ < URL > ”
- Guidelines—Enclose the URL in quotation marks because this entry is a Python string. The value “<http://<IP-address>/idpPortal/Record>” is the default URL specified in the *WEB-INF/web.xml* file for *idpPortal.war*.
- Example—“<http://192.0.2.25/idpPortal/Record>”

DSA URL

- URL of the Web application server running Dynamic Service Activator and the path to the Web service description of Dynamic Service Activator.
- Value—URL in the form “[http\(s\)://<user>:<password>@<host>:<port>/dsa/services/DynamicServiceActivation?wsdl](http(s)://<user>:<password>@<host>:<port>/dsa/services/DynamicServiceActivation?wsdl)”
 - < user > —Client ID configured for Dynamic Service Activator

- `<password >` —Password associated with the client ID configured for Dynamic Service Activator
- `<host >` —Hostname or IP address of the server on which Dynamic Service Activator runs
- `<port >` —Port number used by Dynamic Service Activator on the server
- `wSDL`—Indicates Web Services Description Language
- Guidelines—Enclose the URL in quotation marks because this entry is a Python string.
In the sample implementation, Dynamic Service Activator is used by the **idpsdx.py** integration script to send e-mail and activate the captive portal service.
- Example—“`http://idp:secret@10.227.6.171:8080//dsa/services/DynamicServiceActivation?wSDL`”

DEBUG

- Specifies whether or not to print diagnostic messages to the screen.
- Value—True or False

RECORD

- Specifies whether or not to send messages to the captive portal to record the details of an incident. The portal stores these messages and provides information about the incidents to a subscriber when Web requests for the subscriber are redirected to the captive portal.
- Value—True or False

CAPTIVE

- Specifies whether or not to activate a captive portal to notify subscribers that IDP detected malicious traffic sent to or received from them.
- Value—True or False

CAPTIVE SERVICE

- Specifies the name of the service that activates a captive portal.
For a subscriber to have Web requests redirected to a captive portal, the subscriber must have or inherit a subscription to the service.
- Value—“ `< service name >` ”
- Guidelines—Enclose the service name in quotation marks because this entry is a Python string.
- Example—“Quarantine”

EMAIL

- Specifies whether or not to send notification e-mail messages to subscribers that IDP detected malicious traffic sent or received by them.
- Value—True or False

Sample *idpsdx.py* Script

Through Dynamic Service Activator, the sample **idpsdx.py** script activates the service that redirects subscribers to the captive portal. Because Dynamic Service Activator does not support persistent activation, the sample portal activates the service for the captive portal only for users who are logged in to their account.

If you want subscribers to see the IDP captive portal at any time—for example, when they log out of their account, and then log back in to their account but do not try to access the Web—you can write an SAE extension script and invoke it from the `invokeScript` method in Dynamic Service Activator.