

Juniper Networks Steel-Belted Radius SDK

Juniper Networks Steel-Belted Radius Service Provider Edition is a fully RFC compliant AAA server software solution based on the RADIUS protocol. With this product, Juniper Networks offers a carrier-grade, flexible, high performance solution which can address the most

demanding AAA requirements for both wireline and wireless networks. While the standard feature set for the product meets the requirements of most of our customers, some customers need additional enhancements and custom functionality for a variety of purposes.

To this end, Juniper Networks offers a fully documented software development kit. This SDK complements SBR SP exclusively and includes the following suite of plug-in classes:

- Authentication modules
- Extensible Authentication Protocol (EAP) modules
- Attribute editing modules
- Accounting modules
- Sessions modules

These classes include APIs that allow for the coding of plug-in modules which can be dropped into the product. Each of these five classes of plug-in provides a different type of functionality. Plug-ins allow new functionality to be added to SBR SP in a uniform and modular way, using separate modules that are dynamically linked to the core RADIUS server at runtime.

The major benefit of the plug-in architecture is that it allows third-party vendors and end users to customize their use of SBR SP as necessary, based on their specific requirements or timetables. Additional benefits include the advantages of enforced architectural modularity, such as clarity of design, increased capacity for concurrent development, and better testability.

Supported platforms include Sun Solaris (7, 8 and 9); Microsoft Windows NT, 2000, XP, and Windows Server 2003; and Linux SuSE Enterprise Server 9 (SLES9) and Red Hat (Enterprise and Advanced Server 3).

Each plug-in comprises at least two files:

- **The implementation module.** This is a code module that is dynamically loaded by the SBR SP server and runs in the server's process. This module is an SO under Solaris or a DLL under Windows.
- **The header file.** This is a text file which contains basic information that allows SBR SP to load and start the implementation module. Configuration information specific to the implementation module may also be included in this file.

Plug-ins may be coded in the C and C++ programming languages. The entry points, however, are C function calls that must be exported without name decoration. All entry points into SBR SP available to plug-in modules are also C function calls.

Authentication Modules

Authentication modules are used to perform custom authentications. For example, users can be authenticated against a proprietary database or other (non-RADIUS) authentication server. Using the SBR SP Administrator program, authentication modules can be chained together in any order, and, if desired, Native SBR SP authentication can be skipped.

EAP Modules

EAP modules are used to implement all the logic necessary for a given EAP protocol. These modules do not contain any user-specific information; they simply perform EAP protocol functions with the goal of generating standard credentials that can then be passed to the SBR SP for authentication by an authentication module. For example, the end-result of an EAP MD5-Challenge protocol can be presented to SBR SP as CHAP credentials.

Attribute Editing Modules

Attribute Editing plug-ins are similar to authentication plug-ins and may generally use all the API functions available to authentication plug-ins. Once other configured authentication methods (including native authentication, authentication plug-ins, and directed authentication) have completed, Attribute Editing plug-ins are called in the order of their listing in the [AttributeEditing] section of RADIUS.INI. (N.B.: The initialization string of the module's initialization file (*.att) MUST appear in the [AttributeEditing] section for the module to be called.) Since they have access to the reply packet, Attribute Editing plug-ins may modify the reply list by deleting, modifying, or adding attributes, or may fail the authentication based on the presence or absence of attributes in the reply.

Accounting Modules

Accounting plug-ins are called when an accounting request is received and may perform such functions as forwarding the request to a local or remote database, or adding attributes to the reply.

Session Modules

Session plug-ins are called whenever there is a change in the SBR SP database of current sessions. Uses of Session modules include performing additional logging of session changes and sending notification to external databases.

Plug-ins may allocate IP addresses from IP address pools managed by SBR SP. The plug-ins will be responsible for releasing the IP addresses (via APIs) once they are no longer being used.

Sample code in C and C++ for both platforms (Unix & Win32) is provided. Documentation includes a detailed function reference section for all three types of functions which the implementation module uses to communicate with the RADIUS server, as well as build, debug and installation instructions.

Regarding sample code, the SDK contains seven samples to illustrate the use of the plug-in interface. While the plug-in interface is strictly a C-language interface, the plug-ins can be written in C or C++. To illustrate this point, most of the samples are made available in both C and C++. The seven samples are:

Acct	sample accounting plug-in that records certain RADIUS attributes to a log file
Auth	sample authentication plug-in that authenticates users based on information contained in its configuration file
Authcr	same as "auth" sample, but also includes challenge of end-user to illustrate interactive authentication
Autheap	same as "authcr" sample, but illustrates authentication using Extensible Authentication Protocol (EAP) (available in C++ only)
Autheapprotocol	sample EAP protocol implementation that uses other SBR SP authentication plug-ins to actually authenticate the user (available in C++ only)
Attr	sample attribute editing plug-in (available in C++ only)
Sess	sample session plug-in that watches for certain activity against the session list and records it to a file (available in C++ only)

The sample directories each contain an "nt" and a "unix" subdirectory that contain the make files necessary for building the samples on Microsoft/NT and Sun Solaris.

The SDK also includes a RADIUS client emulation program for testing purposes. This is a Win32 program which can generate and send RADIUS authentication and accounting requests to the RADIUS server to facilitate plug-in testing. Documentation includes instructions for this test tool.

Additionally, the SDK supports Control Points. Control Points are specific points in the SBR SP server's processing of authentication and accounting requests at which a class of plug-in methods can request to be invoked. By developing and installing this type of plug-in, a third party can modify authentication and accounting requests and the resulting responses without actually performing authentications or recording accounting records. All control point plug-in methods, including those invoked for accounting requests, expose an API similar to the API used by attribute editing plug-ins.

Support of Control Points allows the SDK to be usable in the various scenarios where SBR SP is processing RADIUS proxy data. For example, in a wholesale architecture where SBR SP is not performing local authentication for certain customer realms, yet SDK-level attribute editing of the proxy packets is required. Another example would be to control proxy RADIUS data passing between roaming partners, where a third party vendor using SBR SP as a proxy server can provide custom data management services for each network (partner) by coding plug-ins to facilitate access to the data for integration with custom service interfaces. Control Points are fully documented within the SDK.

There are currently eight control point instances defined in SBR SP: four for authentication requests and four for accounting requests. These are:

- Initial-Request
- Outbound to Proxy
- Inbound from Proxy
- Final-Response

Finally, note that SDK support of Control Points and their application to RADIUS proxy data may be especially useful for carriers and wholesale providers that deploy Juniper Networks' Service Level Manager. RADIUS proxy data is an integral part of the SLM client/server architecture and providers may need customized control over the proxy data for a variety of purposes.



CORPORATE HEADQUARTERS
AND SALES HEADQUARTERS
FOR NORTH AND SOUTH AMERICA
Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: 888-JUNIPER (888-586-4737)
or 408-745-2000
Fax: 408-745-2100
www.juniper.net

EAST COAST OFFICE
Juniper Networks, Inc.
10 Technology Park Drive
Westford, MA 01886-3146 USA
Phone: 978-589-5800
Fax: 978-589-0800

ASIA PACIFIC REGIONAL
SALES HEADQUARTERS
Juniper Networks (Hong Kong) Ltd.
Suite 2507-11, Asia Pacific Finance Tower
Citibank Plaza, 3 Garden Road
Central, Hong Kong
Phone: 852-2332-3636
Fax: 852-2574-7803

EUROPE, MIDDLE EAST, AFRICA
REGIONAL SALES HEADQUARTERS
Juniper Networks (UK) Limited
Juniper House
Guildford Road
Leatherhead
Surrey, KT22 9JH, U. K.
Phone: 44(0)1372-385500
Fax: 44(0)1372-385501

Copyright 2006, Juniper Networks, Inc. All rights reserved. Juniper Networks and the Juniper Networks logo are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered trademarks, or registered service marks in this document are the property of Juniper Networks or their respective owners. All specifications are subject to change without notice. Juniper Networks assumes no responsibility for any inaccuracies in this document or for any obligation to update information in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.