



## RSA SecurID Ready Implementation Guide

Last Modified: March 04, 2009

### Partner Information

---

Product Information	
Partner Name	Open System Consultants
Web Site	<a href="http://www.open.com.au">http://www.open.com.au</a>
Product Name	Radiator RADIUS Server, with AuthBy RSAAM
Version & Platform	4.3.1 All Platforms
Product Description	A full featured, flexible, configurable, full source RADIUS server with RSA Authentication Manager Server Web Services API support.
Product Category	Radius Server





## Solution Summary

---

This document describes how the Radiator AuthBy RSAAM authentication module can be used to integrate with RSA Authentication Manager Server 7.1.

Radiator RADIUS Server integrates with RSA Authentication Manager Server:

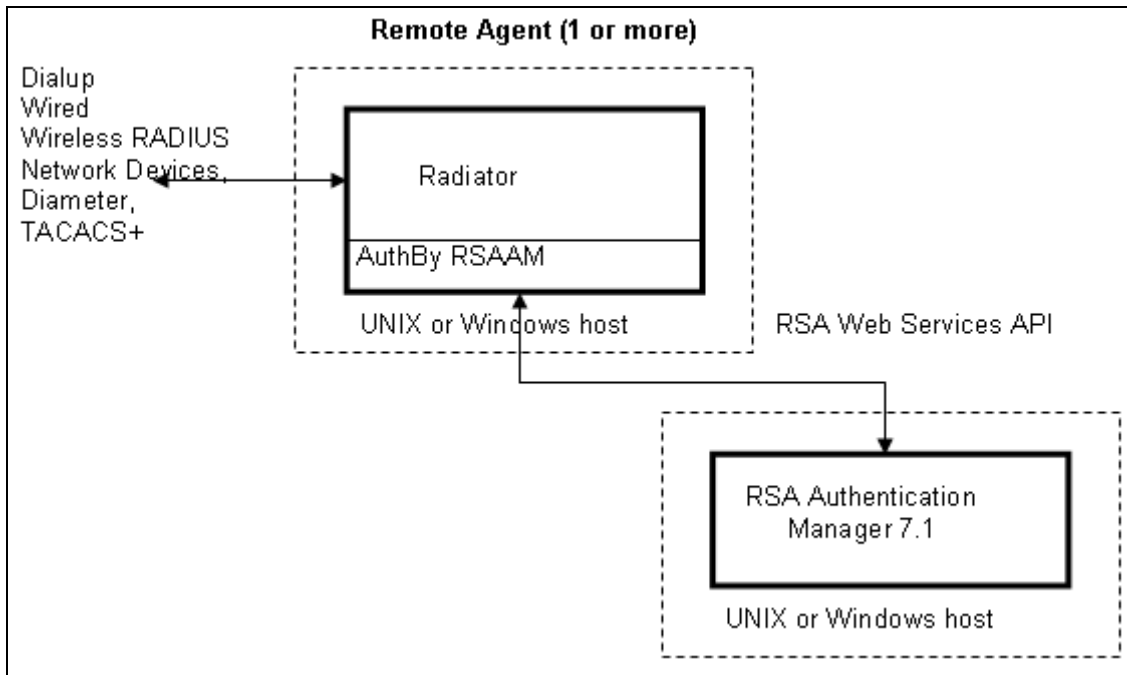
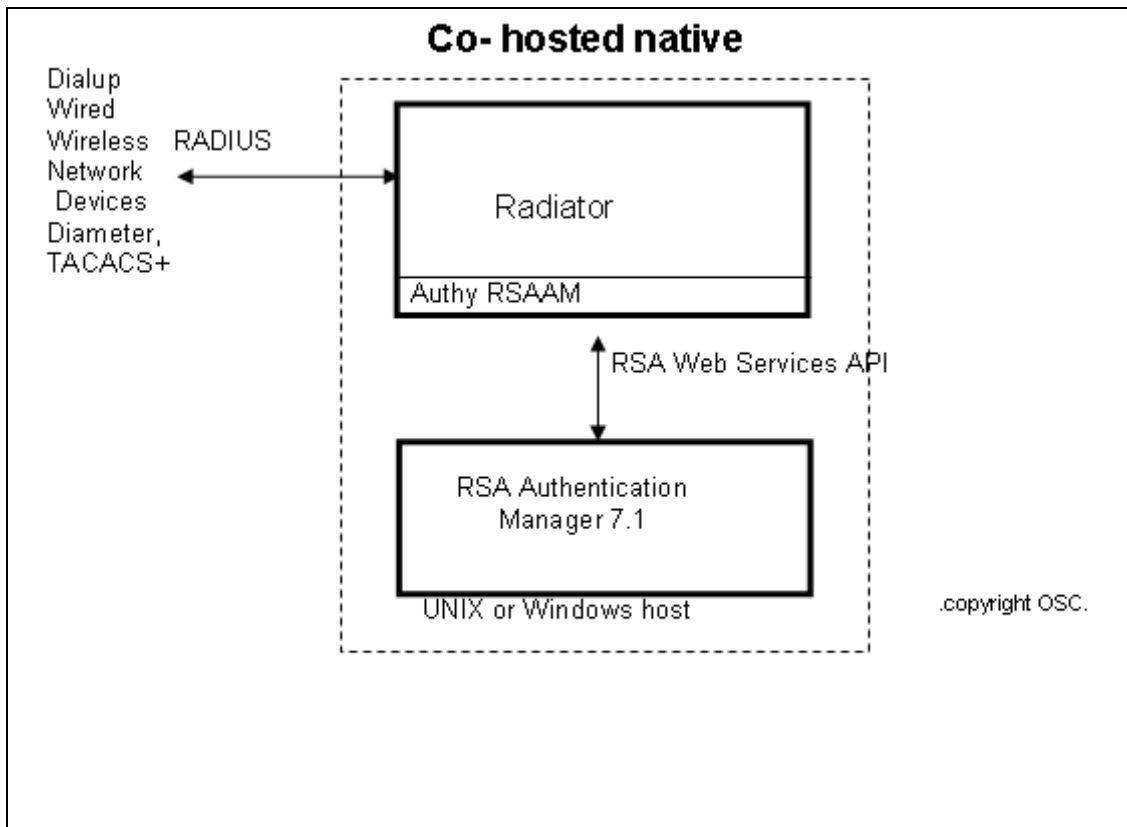
- Via the Web Services API agent for direct authentication
- As a proxy for by sending some or all RADIUS requests to RSA RADIUS server

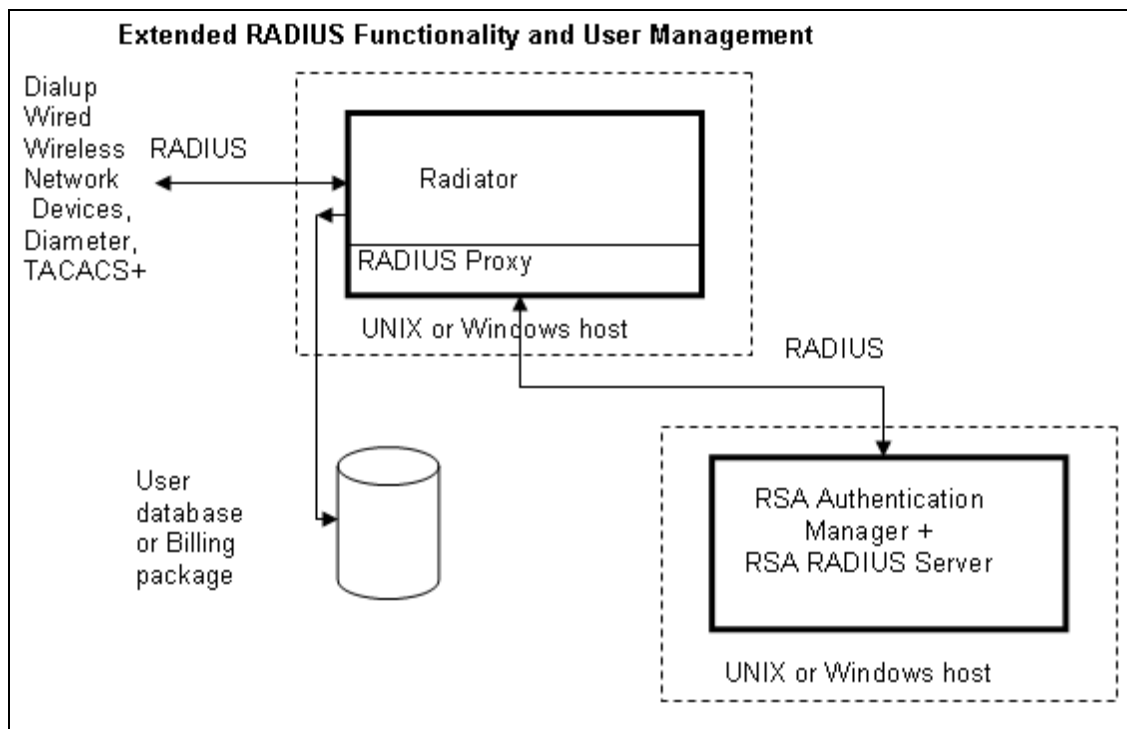
In either case, Radiator can be used to extend or enhance RSA authentication for added value or to add RSA authentication to existing RADIUS, TACACS+ or Diameter-based user management or billing systems, either custom or 3<sup>rd</sup> party. The use of Radiator with RSA Authentication Manager enables authentication solutions and flexibility that is not possible with either product alone.

Radiator is a highly flexible, full source, multi-platform RADIUS server that integrates with RSA Authentication Manager Server. The Radiator AuthBy RSAAM module uses the RSA Authentication Manager 7.1 Web Services API to authenticate RSA tokens, static passwords, On-demand Tokencodes and Security Questions against Authentication Manager Server 7.1.

Partner Integration Overview	
Authentication Methods Supported	SecurID, On-demand, Security Questions
List Library Version Used	Admin API 7.1
RSA Authentication Manager Replica Support	N/A
Secondary Authentication Server Support	Yes (unlimited)
RSA Authentication Agent Host Type for 6.1	N/A
RSA Authentication Agent Host Type for 7.1	N/A
RSA SecurID User Specification	Designated Users, All Users
RSA SecurID Protection of Administrative Users	Yes
RSA Software Token and RSA SecurID 800 Automation	No

Radiator can be deployed with RSA Authentication Manager in a number of ways, depending on exact requirements. Radiator can be deployed on one or more hosts (UNIX or Windows), on the same or different hosts as the ones where Authentication Manager is deployed. This permits maximum flexibility and scalability. Some example architectures are shown below. Radiator's flexibility also permits many other types of deployments.







## Product Requirements

Partner Product Requirements: Radiator RADIUS Server with AuthBy RSAAM	
Version	4.3.1 + patch set 1.1000 or 4.3.2 or later
CPU	All platforms
Memory	512MB
Storage	100MB

Operating System	
Platform	Required Patches
Windows NT, Server 2003, 2007, XP	All Patch Levels Supported
Solaris 8, 9, 10	All Patch Levels Supported
HP-UX 10	All Patch Levels Supported
AIX	All Patch Levels Supported
Linux	All Patch Levels Supported

Additional Software Requirements	
Application	Additional Patches
Perl 5.x or later	All Patch Levels Supported
SOAP::Lite Perl module	All Patch Levels Supported

## Authentication Agent Configuration

The Open Systems Consultants Radiator communicates with the RSA Authentication Manager using the Administrative API. To facilitate communication between the Open Systems Consultants Radiator and the RSA Authentication Manager 7.1 and above / RSA SecurID Appliance 3.0 and above, the default RSA Authentication Manager administrative user account is used. This type of connection does not use Authentication Agent records for authorization, but instead authorization is done using the privileges of the user account.

Please refer to the appropriate RSA Security documentation for additional information about connecting to the RSA Authentication Manager Server using the Administrative API.

## RSA SecurID files

RSA SecurID Authentication Files	
Files	Location
sdconf.rec	N/A
Node Secret	N/A
sdstatus.12	N/A
sdopts.rec	N/A

 **Note:** Go to the appendix of this document to get detailed information regarding these files.



## Partner Product Configuration

---

### ***Before You Begin***

This section provides instructions for integrating the partners' product with RSA SecurID Authentication. This document is not intended to suggest optimum installations or configurations.

It is assumed that the reader has both working knowledge of all products involved, and the ability to perform the tasks outlined in this section. Administrators should have access to the product documentation for all products in order to install the required components.

All vendor products/components must be installed and working prior to the integration. Perform the necessary tests to confirm that this is true before proceeding.

### ***Documenting the Solution***

In order for Radiator to be able to authenticate against an RSA Authentication Manager Server, the Radiator host must have Perl, Radiator and the Soap::Lite Perl module installed. The instructions below describe how to install and configure the Radiator host(s), which may or may not be the same host as the RSA Authentication Manager Server host.

The basic steps to install Radiator with RSA Authentication Manager Server support on the Radiator host are shown below. More detailed instructions are provided in `goodies/ace.txt` in the Radiator distribution.

### **Integration Overview: Windows**

1. Install ActivePerl 5.8 or later.
2. Download and install the Radiator distribution. On Windows the self-extracting executable is easiest and preferred.
3. Install the SOAP::Lite perl module from a command prompt:  

```
ppm install SOAP::Lite
```
4. Install the Net::SSLeay perl module from a command prompt:  

```
ppm install http://www.open.com.au/radiatot/free-downloads/Net-SSLeay.ppd
```
5. Configure Radiator as described below.

### **Integration Overview: Unix (including Linux, Solaris etc)**

1. Ensure Perl 5.x is installed.
2. Obtain and install Perl Net::SSLeay and SOAP::Lite modules for your platform.
3. Download and install the Radiator distribution. The full source distribution is preferred.
4. Configure Radiator as described below.

### **Configuring Radiator**

1. Create a Radiator configuration file with an `<AuthBy RSAAM>` clause. Use the sample configuration file in `goodies/rsaam.cfg` as a starting point.
2. Obtain the `SessionUsername` and `SessionPassword` as described below. Add them to the Radiator configuration file.
3. Start Radiator with the configuration file.
4. Test basic Radiator authentication. Use the `radpwstst` program to send sample RADIUS authentication requests to Radiator which will then authenticate them against the RSA Authentication Manager Server whose details are configured into the Radiator configuration file.
5. Complete configuration of Radiator, based on your specific requirements.
6. Arrange for Radiator to start automatically when the Radiator Host is booted.



## Getting SessionUsername and SessionPassword

In order for Radiator to connect to the RSA Authentication Manager Server, it needs to be configured with the SessionUsername and SessionPassword that will be used to authenticate the connection to RSA Authentication Manager Server. The username and password are generated automatically by RSA Authentication Manager Server when it is installed. You must run a special utility program to get RSA Authentication Manager Server to tell you what the username and password are.

Windows

```
cd "c:\Program Files\RSA Security\RSA Authentication Manager\Utils  
rsautil manage-secrets -m <MASTERPWD> -a list
```

Unix:

```
rsautil manage-secrets -m <MASTERPWD> -a list
```

Where <MASTERPWD> is the RSA Authentication Manager Server master administration password.

The rsautil will print out the 'Comamnd API Client User ID and Password. Transfer these to the SessionUsername and SessionPassword parameters in the Radiator configuration file.

## Selecting a Policy

The Radiator <AuthBy RSAAM> clause in the Radiator configuration file contains a Policy parameter that specifies what type of authentication policy to use for all users that are authenticate through that clause. It controls what information is to be entered as the user's password during RADIUS authentication. You will need to change that Policy setting to suit your needs. The following policies are supported:

- **RSA SecurID\_Native**  
This Policy requires the user to enter their current PIN followed immediately by the tokencode currently showing on their physical token or software token.
- **OnDemand**  
This Policy requires the user to enter their PIN. If a correct PIN is entered, RSA Authentication Manager Server will send (by SMS or email as configured) a temporary tokencode to the user, and challenge the user to enter a Tokencode. The user then enters the tokencode they received.
- **LDAP\_Password**  
This Policy requires that the user enter thier current static password. The password is stored in an LDAP database accessed by the RSA Authentication Manager Server.
- **Security\_Questions**  
The user enters a blank password. They are then challenged to answer a number of user-customised security questions. After each challenge they enter the correct pre-configured answer. The user can configure their own security qurstions and answers using the RSA AM self-service console (see RSA AM documentation for details).
- **RSA\_Password**  
This is the default. This Policy requires that the user enter their current static password. The password is stored in the RSA Authentication Manager Server internal database.

It is possible to configure RSA Authentication Manager to support some or all of these policies for any given user (see the RSA Authentication Manager Server documentation for how to do this). It is the Policy setting of the AuthBy RSAAM which controls which one will actually be used to authenticate a given user.

Depending on your authentication needs and groupings, you may have more than one AuthBy RSAAM in your Radiator configuration, each with a different Policy. If so, you will need to configure Radiator to direct incoming requests to the appropriate AuthBy RSAAM clause. Radiator has a wealth of features that allow such configurations to be achieved.



A common way of authenticating different groups of users in different ways is to assign a different Realm for each category of user (and for each RSAAM Policy), and then use the Radiator Realm clause to direct requests from users in each realm to the appropriate RSAAM clause. For example, the following excerpt from a Radiator configuration file directs users who log in as **username@management.company.com** to authenticate with SecurID\_Native tokens, and users who log in as **username@noc.company.com** will be authenticated with OnDemand tokencodes.

```
# Skeleton config... incomplete...
<Realm management.company.com>
  <AuthBy RSAAM>
    Policy SecurID_Native
  ...
</AuthBy>
</Realm>
<Realm noc.company.com>
  <AuthBy RSAAM>
    Policy OnDemand
  ...
</AuthBy>
</Realm>
```

## Testing Radiator with radpwst

The Radiator distribution contains the radpwst program which can be used to test the complete Radiator/RSA Authentication Manager Server installation. It is recommended that you conduct such tests before testing with the production NAS client that you intend to use.

In order to use radpwst, you need a shell (on UNIX) or a Command Prompt (on Windows) on the Radiator host. Use a command something like:

```
perl radpwst -noacct -interactive -timeout 1000 -user username -password
1111222222
```

Where *username* is the username of the user to authenticate, and where 1111 is the user's PIN and 222222 is the user's current tokencode etc.

Note that if you enter a blank password:

```
perl radpwst -noacct -interactive -timeout 1000 -user username -password ""
```

You will be challenged to enter the type of information required by the Policy setting.

## Failover

Radiator can be configured to implement failover between 2 or more RSA Authentication Manager Servers. Whenever an RSA Authentication Manager Server cannot be contacted, the AuthBy RSAAM clause returns IGNORE. If the AuthByPolicy is ContinueWhileIgnore, then Radiator will try the next AuthBy RSAAM in sequence until a server is successfully contacted.

A typical configuration excerpt might be:

```
# Failover from amserver1 to amserver2
<Realm DEFAULT>
  AuthByPolicy ContinueWhileIgnore
  <AuthBy RSAAM>
    Host amserver1.company.com: 7002
  ...
</AuthBy>
<AuthBy RSAAM>
  Host amserver2.company.com: 7002
  ...
</AuthBy>
</Realm>
```

# Certification Checklist for RSA Authentication Manager 7.1

Date Tested: 02/20/2009

Certification Environment		
Product Name	Version Information	Operating System
RSA Authentication Manager	7.1	Windows 2003 Server Enterprise
Radiator RADIUS Server	4.3.1 + patch set 1.1000	Windows XP

Mandatory Functionality			
RSA Authentication API		RSA Login Command API	
<b>New PIN Mode</b>			
Force Authentication After New PIN	N/A	Force Authentication After New PIN	✓
System Generated PIN	N/A	System Generated PIN	✓
User Defined (4-8 Alphanumeric)	N/A	User Defined (4-8 Alphanumeric)	✓
User Defined (5-7 Numeric)	N/A	User Defined (5-7 Numeric)	✓
Deny 4 and 8 Digit PIN	N/A	Deny 4 and 8 Digit PIN	✓
Deny Alphanumeric PIN	N/A	Deny Alphanumeric PIN	✓
Deny Numeric PIN	N/A	Deny Numeric PIN	✓
PIN Reuse	N/A	PIN Reuse	✓
<b>Passcode</b>			
16 Digit Passcode	N/A	16 Digit Passcode	✓
4 Digit Fixed Passcode	N/A	4 Digit Fixed Passcode	✓
<b>Next Tokencode Mode</b>			
Next Tokencode Mode	N/A	Next Tokencode Mode	✓
<b>Load Balancing / Reliability Testing</b>			
Failover (3-10 Replicas)	N/A	Failover (3-10 Replicas)	✓
No RSA Authentication Manager	N/A	No RSA Authentication Manager	✓

Additional Functionality			
RSA Authentication API		RSA Login Command API	
<b>On-demand</b>			
System Generated PIN		System Generated PIN	✓
User Defined (8 Digit Numeric)		User Defined (8 Digit Numeric)	✓
User Disabled		User Disabled	✓
User Expired		User Expired	✓
<b>Security Questions</b>			
Questions with Answers		Questions with Answers	✓
Questions with out Answers		Questions with out Answers	✓
User Expired		User Expired	✓
User Disabled		User Disabled	✓

BSD / PAR

✓ = Pass ✗ = Fail N/A = Non-Available Function



## Known Issues

---

### ***Identities in Multiple Identity Sources***

This solution does not have support for an identity to exist in multiple identity sources. In order for this solution to operate properly, each identity must be unique.



## Appendix

---

### ***Node Secret:***

This file is used by the Authentication Agent when using the RSA SecurID native protocol for encrypting the communication. The LoginCommand API does not create a node secret; instead communications are encrypted with SSL.

### ***sdconf.rec***

This file is used by the Authentication Agent for configuration. It provides the agent with information regarding the RSA Authentication Manager Server. The LoginCommand API does not make use of the sdconf.rec but instead uses the Radiator XML configuration file.

### ***sdopts.rec:***

This file is used by the Authentication Agent for configuration. It provides the agent with information such as the IP Address to use as a parameter for protocol encryption. The LoginCommand API does not make use of the sdopts.rec.

### ***sdstatus.12:***

This file is used by the Authentication Agent for configuration. The LoginCommand API does not make use of the sdstatus.12 file.