Open System Consultants
Radiator RADIUS Server

RSA SecurID Ready Implementation Guide

Last Modified: July 9, 2013

Partner Information

Product Information	
Partner Name	Open System Consultants
Web Site	www.open.com.au
Product Name	Radiator RADIUS Server with AuthBy RSAAM
Version & Platform	4.11 all Platforms
Product Description	A full featured, flexible, configurable, full source RADIUS server with RSA Authentication Manager Server Web Services API support.

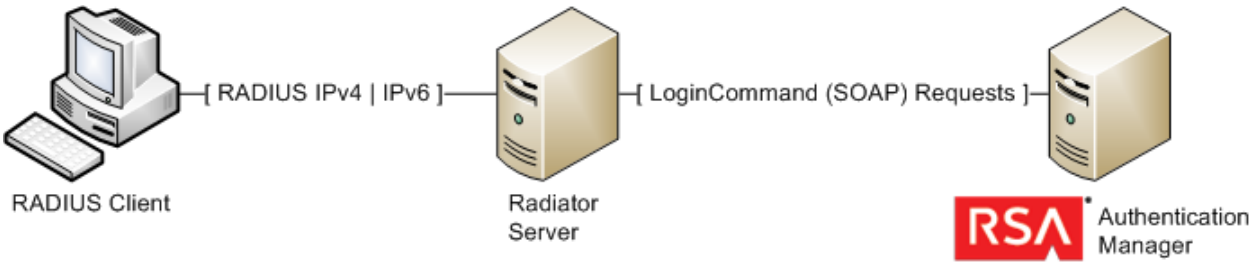
Radiator
www.open.com.au

Solution Summary

This document describes how the Radiator AuthBy RSAAM authentication module can be used to integrate with RSA Authentication Manager. Radiator integrates with RSA Authentication Manager via the Web Services API for direct authentication.

Radiator can be used to extend or enhance RSA Authentication for added value. It can add RSA Authentication to existing RADIUS, TACACS+ or Diameter-based user management or billing systems. This can be either custom or 3rd party. The use of Radiator with RSA Authentication Manager enables 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. The Radiator AuthBy RSAAM module uses the RSA Authentication Manager Web Services API to authenticate RSA tokens, static passwords, On-demand Authenticators and Security Questions against Authentication Manager.



Authentication Agent Configuration

The Open System Consultants Radiator communicates with the RSA Authentication Manager using the Authentication Manager Administrative API. To facilitate communication between Radiator and RSA Authentication Manager, 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

Partner Product Configuration

Before You Begin

This section provides instructions for configuring Radiator 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 Radiator components must be installed and working prior to the integration. Perform the necessary tests to confirm that this is true before proceeding.

Configuring Radiator for SecurID Authentication

In order for Radiator to be able to authenticate against RSA Authentication Manager, the Radiator host must have Perl, Radiator, and the SOAP::Lite Perl module and its prerequisites 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 host.

The basic steps to install Radiator with RSA Authentication Manager on the same 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 and all of their prerequisite modules.
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 radpwtst program to send sample RADIUS authentication requests to Radiator which will then authenticate them against the RSA Authentication Manager whose details are in configured into the Radiator configuration file.
5. Complete configuration of Radiator, based on specific requirements for deployment.
6. Arrange for Radiator to start automatically when the Radiator Host is booted.

Getting SessionUsername and SessionPassword

In order for Radiator to connect to RSA Authentication Manager, it needs to be configured with the SessionUsername and SessionPassword that will be used to authenticate the connection to RSA Authentication Manager. The username and password are generated automatically by RSA Authentication Manager when it is installed. An RSA supplied utility program will return what the username and password are.

```
rsautil manage-secrets -a list
```

When prompted, supply the AM Master Password (AM 7.1) or Operations Console administrator credentials (AM 8.0).

The command will print out the 'Command 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 authenticated through that clause. It controls what information is to be entered as the user's password during RADIUS authentication. 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 their 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-customized security questions. After each challenge they enter the correct pre-configured answer. The user can configure their own security questions 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 be used to authenticate a given user.

Depending on authentication needs and groupings, there may be more than one AuthBy RSAAM in the Radiator configuration, each with a different Policy. If this is the case, 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 radpwtst

The Radiator distribution contains the radpwtst program which can be used to test the complete Radiator/RSA Authentication Manager installation. It is recommended that such tests are conducted before testing with the production NAS client.

In order to use radpwtst, open a shell (on UNIX) or a Command Prompt (on Windows) on the Radiator host:

```
perl radpwtst –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 a blank password is entered:

```
perl radpwtst –noacct -interactive –timeout 1000 -user username –password ""
```

The user 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

Date Tested: July 9, 2013

Certification Environment		
Product Name	**Version Information**	**Operating System**
RSA Authentication Manager	8.0	Virtual Appliance
Radiator RADIUS Server	4.11 with patches	Centos 6.2 x86 and x64

Mandatory Functionality			
RSA Authentication API		**RSA Login Command API**	
New PIN Mode			
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	✓
Passcode			
16 Digit Passcode	N/A	16 Digit Passcode	✓
4 Digit Fixed Passcode	N/A	4 Digit Fixed Passcode	✓
Next Tokencode Mode			
Next Tokencode Mode	N/A	Next Tokencode Mode	✓
Load Balancing / Reliability Testing			
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**	
On-demand			
System Generated PIN	N/A	System Generated PIN	✓
User Defined (8 Digit Numeric)	N/A	User Defined (8 Digit Numeric)	✓
User Disabled	N/A	User Disabled	✓
User Expired	N/A	User Expired	✓
Security Questions			
Questions with Answers	N/A	Questions with Answers	✓
Questions without Answers	N/A	Questions without Answers	✓
User Expired	N/A	User Expired	✓
User Disabled	N/A	User Disabled	✓

MRQ

✓ = 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.

Authentication Manager 7.1 SP3 API connection authentication errors

RSA Authentication Manager SP3 and later requires Radiator 4.6 plus patches of 2010-05-12 (or later). Without this Radiator patch, the API connection between Radiator and AM will fail with a '401: Unauthorized' error.

Appendix

Partner Integration Details	
RSA SecurID API	Authentication Manager SOAP WSDL
RSA Authentication Agent Type	N/A
RSA SecurID User Specification	Designated Users
Display RSA Server Info	No
Perform Test Authentication	Yes
Agent Tracing	Yes

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.