

Testing RSA SecurID Integration with Solaris PAM and Solaris SSH

White Paper - Cookbook

Luc Wijns
July 2007

Table of Contents

Introduction.....	3
RSA SecurID Product Line.....	3
Solaris 10 SSH 1.1.....	3
Architecture.....	4
RSA Products Installation.....	4
RSA Authentication Manager installation.....	4
changes in /etc/system.....	4
Add the following lines in /etc/services.....	4
Create RSA Administrators.....	5
Create RSA Installation Directory.....	5
Rename the License and certificate files to smallcaps.....	5
Install the RSA Authentication Manager Software.....	5
Postinstallation on the Primary Host.....	8
Start the database brokers.....	9
Create user and assign a token.....	9
Restart Everything.....	9
Test user Authentication against the RSA authentication Manager.....	10
Install remote administration and configure remote administrators auth method.....	10
RSA PAM module for UNIX installation.....	10
SSH and PAM Configuration.....	11
SunRAY and PAM Configuration	
.....	13
About the Author.....	15

Introduction

This whitepaper illustrates the work that has been done in order to validate the interoperability of the RSA SecurID product line with the PAM authentication mechanism of Solaris 10 11/06. This integration focused on the the following protocols SSH and rlogin, but also integration with dtlogin especially on Sun Ray Server has been evaluated.

The evaluation was triggered by the fact that RSA Security in its documentation does not reflect Solaris 10 SSH as being a supported platform, but refer to different releases of OpenSSH running on Solaris10 on SPARC as well as on x86 platforms. Therefore the evaluation focused mainly on the integration with Solaris SSH version 1.1 which is the default SSH platform delivered with Solaris10.

For more information on RSA SecurID support matrix see <http://www.rsa.com/node.aspx?id=2573>.

RSA SecurID Product Line

The RSA SecurID product line is very widely used in the industry where strong or multi-factor authentication is mandated by the security policy. The products evaluated are:

- the RSA Authentication Manager 6.1
- the RSA Authentication for PAM 6.0
- the RSA SecurID AES tokens

Note – The RSA RADIUS server has not be evaluated, the usage of the RADIUS protocol for authentication being out of the scope of the current evaluation.

The RSA SecurID support matrix for Solaris depicts the following functionality:

RSA Authentication Agent for PAM 6.0

Sun® Solaris™ 10 x86, 32-bit :

- Standard Tools: login, rlogin, dtlogin, telnet, rsh, su, ftp
- Open SSH Tools: sftp, ssh, scp from OpenSSH 4.3p2

Sun® Solaris™ 10 x86, 64-bit and Sun® Solaris™ 10 SPARC, 64bit

- Standard Tools: login, rlogin, dtlogin, telnet, rsh, su, ftp
- Open SSH Tools: sftp, ssh, scp from OpenSSH 4.5p1

The initial purpose of this evaluation is to confirm that the RSA Authentication Agent for PAM 6.0 works with Solaris SSH 1.1 which can then be considered as a “standard tool” and does not require installation of OpenSSH version.

Solaris 10 SSH 1.1

Solaris SSH was introduced into the Solaris 9 release of Solaris with its version number being Solaris SSH1.0.1. Starting with Solaris 10 release ?? Solaris SSH 1.1 was introduced. Solaris SSH is not derived from OpenSSH. OpenSSH usage on the Solaris platform is not recommended, as it lacks the necessary enhancements implemented into Solaris SSH.

Architecture

The infrastructure is composed of two Solaris SPARC machines.

One Solaris 9 machine running the RSA Authentication Manager 6.1 software using the local Solaris user store as account database. This system is also used to initiate tokens and to provision PIN codes for those ones. The RSA Authentication Manager does not support Solaris 10.

This machine is connected to the Internet via firewalls in order to get accurate NTP synchronization.

Another system runs Sun Ray Software 4.0 u1 and the RSA PAM module for UNIX on top of Solaris 10 u3.

Authentication testing is then done from external clients: Solaris SSH Client, Putty on Windows XP Professional, but also on Sun Ray DTUs directly attached to this host.

RSA Products Installation

RSA Authentication Manager installation

On the Solaris 9 host NTP is configured to synchronize regularly with the NTP server on Internet.

Like described into the RSA documentation the following installation script is executed:

changes in /etc/system

```
shmsys:shminfo_shmseg 16
```

Add the following lines in /etc/services

```
securid 5500/udp
securidprop_00 5505/tcp
securidprop_01 5506/tcp
securidprop_02 5507/tcp
securidprop_03 5508/tcp
securidprop_04 5509/tcp
securidprop_05 5510/tcp
securidprop_06 5511/tcp
securidprop_07 5512/tcp
securidprop_08 5513/tcp
securidprop_09 5514/tcp
securidprop_10 5515/tcp
sdlog 5520/tcp
sdserv 5530/tcp
sdreport 5540/tcp
sdadmind 5550/tcp
sdlockmgr 5560/tcp
sdcommd 5570/tcp
sdoad 5580/tcp
tacacs 49/tcp
```

Create RSA Administrators

```
create unix group rsa
create user securid
```

Create RSA Installation Directory

```
# mkdir /opt/rsa -----> The installation directory
# transfer license files from the cd into /opt/rsa
```

Rename the License and certificate files to smallcaps

```
# mv LICENSE.REC license.rec
# mv SDTI.CER sdti.cer
# mv SERVER.CER server.cer
# mv SERVER.KEY server.key
```

Install the RSA Authentication Manager Software

Chosen root being the administrator, and install the software as being the primary host (no secondary was installed)

```
# cd /opt/rsa
# /cdrom/cdrom0/aceserv/sol/sdsetup -primary
```

The installation is performed completely automatically

```
==== license agreement ==== not pasted
If you have not done so already, ensure that the system time is set
accurately. To obtain the current Coordinated Universal Time
(UTC) you can call a reliable time service. You can call the
following United States telephone number +1 303-499-7111.
```

The system clocks on the Primary and Replica Servers
must be set to the same time (to within one minute).

The overall installation may take from 5 to 90 minutes.

```
The current time is Fri Jun 15 15:00:52 MEST 2007
Moving RSA Authentication Manager Primary Server software to
'/opt/rsa/ace/prog'...
The current time is Fri Jun 15 15:01:23 MEST 2007
Moving RSA Authentication Manager database software to
'/opt/rsa/ace/rdbms'...
The current time is Fri Jun 15 15:01:27 MEST 2007
Uncompressing Progress software...
The current time is Fri Jun 15 15:01:27 MEST 2007
Uncompressing RSA Authentication Manager Primary Server software...
The current time is Fri Jun 15 15:01:42 MEST 2007
```

```
Moving all utility files into
/opt/rsa/ace/utills...
The current time is Fri Jun 15 15:01:42 MEST 2007
Moving README files into
/opt/rsa/ace/doc...
Moving Documentation files into
/opt/rsa/ace/doc...
The current time is Fri Jun 15 15:01:55 MEST 2007
Moving all admin utility files into
/opt/rsa/ace/utills...
The current time is Fri Jun 15 15:01:55 MEST 2007
Creating RSA Authentication Manager Primary Server runtime scripts...
Changing file group IDs...
The current time is Fri Jun 15 15:01:57 MEST 2007
Changing file ownerships...
The current time is Fri Jun 15 15:01:57 MEST 2007
Changing file permissions...
Created symbolic link /usr/lib/libsdxauthr.so to reference
external authorization library /opt/rsa/ace/prog/libsdxauthr.so
The current time is Fri Jun 15 15:01:57 MEST 2007
Setup database for Primary Server...
    Copyright 1994 - 2005 by RSA Security Inc.
    RSA Authentication Manager 6.1
```

---ALL RIGHTS RESERVED---

```
Creating server database...
Creating log database...
New log database creation completed
Server database has been initialized
New server database creation completed
The current time is Fri Jun 15 15:02:02 MEST 2007
Added current host as Primary Server to the database.
Creating realm administrator 'root.'
    Copyright 1994 - 2005 by RSA Security Inc.
    RSA Authentication Manager 6.1
```

---ALL RIGHTS RESERVED---

Realm administrator root created.

```
WARNING: Evaluation License
Your RSA Authentication Manager has an Evaluation license.
Your RSA Authentication Manager is within license limits.
Active User Limit: 2
Active Users in the Database: 0
You can add 2 more active user(s) to the database.
Replica Limit: 1
Replicas in the Database: 0
You can add 1 more Replica(s) to the database.
Your license will expire on 01/03/2008.
The RSA Authentication Manager will stop authenticating on this date.
Order a new Base or Advanced license by 12/29/2007
```

Licensee:

RSA
Promo
Promo
Promo

License: Evaluation

License Status: COMPLIANT

Number Licensed Realms: 1

License ID: 11111111

Expiration Date: Jan 3 2008 16:45:01

Starting RSA key generation, this could take up to 15 seconds.

Verifying the RSA keys in the key pair file.

Success generating key pair file: /opt/rsa/ace/data/radius.key.

Creating a PKCS#10 certificate request file.

Verifying the certificate request file.

Success generating request file: /opt/rsa/ace/data/radius.req.

Generating certificate file.

Verifying certificate file.

Success generating certificate file: /opt/rsa/ace/data/radius.cer.

Changing file group IDs...

The current time is Fri Jun 15 15:02:05 MEST 2007

Changing file ownerships...

Copyright 1994 - 2005 by RSA Security Inc.

RSA Authentication Manager 6.1

---ALL RIGHTS RESERVED---

Server License and Configuration 6.1 [300]

LICENSE CREATION:	Dec 18 2006
LICENSE ID:	11111111
EVALUATION:	This is an Evaluation license.
EXPIRATION:	01/03/2008
RSA Authentication Manager version:	6.1 [300]
Config File Version:	14
FILE OWNERSHIP:	root
AGENT RETRY:	5 times
AGENT TIMEOUT:	5 sec
DES/RC5 ENCRYPTION:	allowed and enabled
TACACS PLUS:	disabled
PRIMARY RSA Authentication Manager:	eid-snap9
PRIMARY RSA Authentication Manager ADDRESS:	192.168.254.252
THIS SERVER:	eid-snap9
THIS SERVER ADDRESS:	192.168.254.252
ACTING MASTER SERVER:	not configured
ACTING SLAVE SERVER:	not configured

```

REPLICA TIMEOUT                30 sec
REPLICA HEARTBEAT              300 sec

AUTHENTICATION SERVICE:       securid
  PORT NUMBER:                 5500

ADDRESSES:                     By IP address in the database
ADMINISTRATION SERVICE:       sdadmin
  PORT NUMBER:                 5550

LOCK MANAGER SERVICE:         sdlockmgr
  PORT NUMBER:                 5560

OFFLINE AUTH DATA SERVICE:    sdoad
  PORT NUMBER:                 5580

BAD PASSCODES before setting NEXT TOKENCODE:
  UNIX agents:                 3
  Communications servers:      3
  Single transaction agents:   3
  NOS agents:                  3

BAD PASSCODES before Disabling Token:
  UNIX agents:                 10
  Communications servers:      10
  Single transaction agents:   10
  NOS agents:                  10
RESPONSE DELAY:                2

ALIAS IP ADDRESS LIST:         no aliases configured

LICENSE CONFIGURATION

LICENSE:                        Base
NUMBER LICENSED ACTIVE USERS:  2
NUMBER LICENSED REPLICAS:      1

```

This license was created for:

```

RSA
Promo
Promo Promo
Installation of RSA Authentication Manager Primary Server software complete at Fri Jun 15
15:08:23 MEST 2007.
#

```

Postinstallation on the Primary Host

Log on to the Primary as the RSA Authentication Manager file owner.

In this particular case this is root.

Start the database brokers

```
# ./sdconnect start
      Copyright 1994 - 2005 by RSA Security Inc.
      RSA Authentication Manager 6.1
      ---ALL RIGHTS RESERVED---
Message: Starting server database broker and page writers.
15:17:15 BROKER 0: Multi-user session begin. (333)
15:17:15 BROKER 0: Begin Physical Redo Phase at 0 . (5326)
15:17:15 BROKER 0: Physical Redo Phase Completed at blk 0 off 4034 upd 0. (7161)
15:17:15 BROKER 0: This server accepts secure clients only. (8947)
15:17:15 BROKER 0: Started for sdserv using TCP, pid 2919. (5644)
Message: Starting log database broker and page writers.
15:17:15 BROKER 0: Multi-user session begin. (333)
15:17:15 BROKER 0: Begin Physical Redo Phase at 0 . (5326)
15:17:15 BROKER 0: Physical Redo Phase Completed at blk 0 off 1269 upd 0. (7161)
15:17:15 BROKER 0: This server accepts secure clients only. (8947)
15:17:15 BROKER 0: Started for sdlog using TCP, pid 2927. (5644)
Message: Starting Automated Audit Log Maintenance.
Message: Starting Administration Daemon.
Message: Starting Job Executor Daemon.
Message: Starting Offline Auth Data Daemon.
Message: Starting Quick Admin Daemon.
Message: Starting Replication.
Message: Database broker start operation completed
```

Create user and assign a token

```
# ./sadmadmin
Token
install token
User
add user and assign a token in the same menu
add Agent host
test user authentication
```

Restart Everything

```
# ./sdconnect start
      Copyright 1994 - 2005 by RSA Security Inc.
      RSA Authentication Manager 6.1
      ---ALL RIGHTS RESERVED---
Message: Server database broker is already running
Message: Log database broker is already running
Message: Automated Audit Log Maintenance already running.
Message: sdadmind is already running.
Message: Job Executor Daemon is already running.
Message: oad is already running.
Message: Quick Admin Daemon is already running.
```

```

Message: Replication is already running.
Message: Database broker start operation completed
# ./aceserver start
      Copyright 1994 - 2005 by RSA Security Inc.
      RSA Authentication Manager 6.1

      ---ALL RIGHTS RESERVED---
Message: Starting RSA Authentication Manager.
Message: RSA Authentication Manager start operation completed

```

Test user Authentication against the RSA authentication Manager

```

./sdtestauth
then enter <PIN><Tokencode>

```

Install remote administration and configure remote administrators auth method

```

./sdadmin
System -> edit system parameters
choose user password
install auth manager on pc

```

RSA PAM module for UNIX installation

```

# mkdir /var/ace
copy server.cer and sdconf.rec there from the configuration directory on the Authentication Manager Host

```

```

copy the CD in /export/rsa
cd authenti/sol
rename sd_pam.a.tar by sd_pam.agent.tar

```

Run the install script. Type:

```
./install_pam.sh
```

```

Do you accept the License Terms and Conditions stated above? (Accept/Decline) [D] A
Enter Directory where sdconf.rec is located [/var/ace]
Please enter the root path for the RSA Authentication Agent for PAM directory [/opt]
/opt/rsa
The RSA Authentication Agent for PAM will be installed in the /opt/rsa directory.
x pam, 0 bytes, 0 tape blocks
x pam/doc, 0 bytes, 0 tape blocks
x pam/lib, 0 bytes, 0 tape blocks
x pam/lib/pam_securedid.so, 365772 bytes, 715 tape blocks
x pam/bin, 0 bytes, 0 tape blocks
x pam/bin/acestatus, 182928 bytes, 358 tape blocks
x pam/bin/acetest, 323764 bytes, 633 tape blocks
cp: cannot access /export/rsa/uninstall*

```

```
Checking /etc/sd_pam.conf:
```

```

VAR_ACE does not exist - entry will be appended
ENABLE_GROUP_SUPPORT does not exist - entry will be appended
INCL_EXCL_GROUPS does not exist - entry will be appended
LIST_OF_GROUPS does not exist - entry will be appended
PAM_IGNORE_SUPPORT does not exist - entry will be appended
AUTH_CHALLENGE_USERNAME_STR does not exist - entry will be appended
AUTH_CHALLENGE_RESERVE_REQUEST_STR does not exist - entry will be appended
AUTH_CHALLENGE_PASSCODE_STR does not exist - entry will be appended
AUTH_CHALLENGE_PASSWORD_STR does not exist - entry will be appended

```

```

*****
* You have successfully installed RSA Authentication Agent 6.0 for PAM
*****

```

Create the same users than on the auth manager

(with same user id)

Go to the authentication manager and add the agent host via sadmin

```
cd /opt/rsa/pam/bin
```

```
./acetest
```

SSH and PAM Configuration

With SecurID we need to configure the ssh server to use Keyboard Interactive Authentication Method. This method supports generic PAM-like conversations, whereas "password" user Authentication does not. By default the Keyboard Interactive Authentication is enabled into the SSH Server configuration file `sshd_conf`, but the password authentication is also enabled by default.

Adding the following lines into the `pam.conf` file require that a user authenticate with his unix password and then get prompted for his SecurID Token and this is only for the Keyboard Interactive Authentication Method which applies only to SSHv2. Those lines does not apply to either SSHv1 nor SSHv2 "password" authentication method.

```

sshd-kbdint  auth requisite    pam_authtok_get.so.1
sshd-kbdint  auth required    pam_dhkeys.so.1
sshd-kbdint  auth required    pam_unix_cred.so.1
sshd-kbdint  auth required    pam_unix_auth.so.1
sshd-kbdint  auth required    pam_secured.so

```

Let see what happens with different configurations:

Client on sol10 and server on sol10:

```

$ uname -a
SunOS sunray 5.10 Generic_118833-33 sun4u sparc SUNW,Sun-Fire-480R
$ ssh token1@192.168.254.244
Password:

```

Enter PASSCODE:

Last login: Mon Jun 18 09:43:48 2007 from 129.150.117.166

\$ uname -a

SunOS smf-490-1 5.10 Generic_118833-36 sun4u sparc SUNW,Sun-Fire-V490

\$

The PASSCODE is the concatenation of the PIN and the SecurID Token Code.
With putty on the client side it works also.

Client on sol9 and server on sol10:

With the ssh 1.0.1 client on solaris 9 and sshd on the same solaris 10 server than in the previous example there is a failure:

Once the user's password is entered the system does not ask for the PASSCODE and the user is granted access

ssh -V

SSH Version Sun_SSH_1.0.1, protocol versions 1.5/2.0.

eid-snap9[2908]# ssh token1@192.168.254.244

token1@192.168.254.244's password:

Last login: Mon Jun 18 10:39:48 2007 from eid-snap9

\$

Here we see that we do not use the keyboard interactive method because the SSH1.0.1 client on Solaris 9 “prefers the “password” authentication method. As such the password authentication method of SSHv2 is evaluated and the PAM entries are not evaluated (we do not see anymore the unix password request).

We can validate this statement, typing only invalid passwords, and see that the “password” authentication method is evaluated first and then only the “keyboard interactive” method, which use the sshd-kbdint PAM entries (we see the request for the unix password and then the SecurID PASSCODE).

ssh token1@192.168.254.244

token1@192.168.254.244's password:

Permission denied, please try again.

token1@192.168.254.244's password:

Permission denied, please try again.

token1@192.168.254.244's password:

Password:

Enter PASSCODE:

Password:

Enter PASSCODE:

Unable to find an authentication method

Now I I do the same exercise (typing only invalid passwords), but this time from a SSH1.1 client I see that the SSH1.1 client prefers "keyboard-interactive" userauth to "password" userauth because we do not see the ssh password prompt

anymore.

here is what happens on a solaris10 client:

```
ssh token1@129.159.232.90
```

```
Password:
```

```
Enter PASSCODE:
```

```
Password:
```

```
Enter PASSCODE:
```

```
Password:
```

```
Enter PASSCODE:
```

```
Permission denied (gssapi-keyex,gssapi-with-mic,publickey,keyboard-interactive).
```

```
$
```

If we test with ssh 1.1 on solaris 9 it will give exactly the same behaviour.

If you want Unix password + SecurID and support multiple clients then you MUST disable SSHv1/2 "password" userauth. There are two ways to do this.

1. edit /etc/ssh/sshd_config and set

```
PasswordAuthentication=no
```

then restart the ssh service.

AND/OR

2. edit /etc/pam.conf and add

```
sshd-password auth requisite pam_deny.so.1
```

```
sshd-password account requisite pam_deny.so.1
```

No need to restart the ssh service.

If you want SecurID only (no Unix password), then copy the ssh-kbdint configuration you showed me to sshd-password and ***remove*** the line that references pam_unix_auth.so.1.

SunRAY and PAM Configuration

I can also do password + securid authentication on Sunray.

I can do dual authN factor with xscreensaver on jds but not on the dtssession-dtscreen (I can do only one factor)

Here is typically how the SecurID PAM can be integrated into pam.conf file for Sun Ray authentication.

```
# added to xscreensaver by SunRay Server Software -- xscreensaver
xscreensaver auth sufficient /opt/SUNWut/lib/pam_sunray.so syncondisplay
xscreensaver auth requisite pam_authtok_get.so.1
xscreensaver auth required pam_dhkeys.so.1
xscreensaver auth required pam_unix_cred.so.1
xscreensaver auth required pam_unix_auth.so.1
xscreensaver auth required pam_secuid.so
xscreensaver account requisite pam_roles.so.1
xscreensaver account required pam_unix_account.so.1
xscreensaver session required pam_unix_session.so.1
xscreensaver password required pam_dhkeys.so.1
xscreensaver password requisite pam_authtok_get.so.1
xscreensaver password requisite pam_authtok_check.so.1
xscreensaver password required pam_authtok_store.so.1
# added to dtlogin-SunRay by SunRay Server Software -- dtlogin-SunRay
dtlogin-SunRay session required pam_unix_session.so.1
dtlogin-SunRay password required pam_dhkeys.so.1
dtlogin-SunRay password requisite pam_authtok_get.so.1
dtlogin-SunRay password requisite pam_authtok_check.so.1
dtlogin-SunRay password required pam_authtok_store.so.1
dtlogin-SunRay auth sufficient /opt/SUNWut/lib/pam_sunray.so
dtlogin-SunRay auth required pam_secuid.so
dtlogin-SunRay auth requisite /opt/SUNWut/lib/sunray_get_user.so.1
property=username
dtlogin-SunRay auth required /opt/SUNWut/lib/pam_sunray_amgh.so.1
dtlogin-SunRay auth requisite /opt/SUNWut/lib/sunray_get_user.so.1 prompt
dtlogin-SunRay auth required /opt/SUNWut/lib/pam_sunray_amgh.so.1 clearuser
dtlogin-SunRay auth requisite pam_authtok_get.so.1
dtlogin-SunRay auth required pam_dhkeys.so.1
dtlogin-SunRay auth required pam_unix_cred.so.1
dtlogin-SunRay auth required pam_unix_auth.so.1
dtlogin-SunRay auth required pam_secuid.so
dtlogin-SunRay account sufficient /opt/SUNWut/lib/pam_sunray.so
dtlogin-SunRay account requisite pam_roles.so.1
dtlogin-SunRay account required pam_unix_account.so.1
# added to dtsession-SunRay by SunRay Server Software -- dtsession-SunRay
dtsession-SunRay auth sufficient /opt/SUNWut/lib/pam_sunray.so syncondisplay
dtsession-SunRay auth requisite pam_authtok_get.so.1
dtsession-SunRay auth required pam_dhkeys.so.1
dtsession-SunRay auth required pam_unix_cred.so.1
dtsession-SunRay auth required pam_unix_auth.so.1
dtsession-SunRay auth required pam_secuid.so
dtsession-SunRay account requisite pam_roles.so.1
dtsession-SunRay account required pam_unix_account.so.1
dtsession-SunRay session required pam_unix_session.so.1
dtsession-SunRay password required pam_dhkeys.so.1
dtsession-SunRay password requisite pam_authtok_get.so.1
dtsession-SunRay password requisite pam_authtok_check.so.1
dtsession-SunRay password required pam_authtok_store.so.1
# added to utnsclogin by SunRay Server Software -- utnsclogin
utnsclogin account requisite pam_roles.so.1
utnsclogin account required pam_unix_account.so.1
utnsclogin session required pam_unix_session.so.1
utnsclogin password required pam_dhkeys.so.1
utnsclogin password requisite pam_authtok_get.so.1
```

```
utnsclogin password requisite pam_authtok_check.so.1
utnsclogin password required pam_authtok_store.so.1
utnsclogin auth requisite /opt/SUNWut/lib/sunray_get_user.so.1 property=username
utnsclogin auth required /opt/SUNWut/lib/pam_sunray_amgh.so.1
utnsclogin auth requisite pam_authtok_get.so.1
utnsclogin auth required pam_dhkeys.so.1
utnsclogin auth required pam_unix_cred.so.1
utnsclogin auth required pam_unix_auth.so.1
```

About the Author

Luc Wijns is Principal Engineer in the Global Systems Engineering division of Sun. Luc is working in Belgium in the position of Chief Technologist for Belgium and Luxembourg.

For more than 10 years working for Sun, Luc held many positions into the Systems Engineering division, like Systems Engineer, Senior Systems Engineer and Principal Architect. Luc's focus is on architectures, but spends also a lot of his time on Security. Luc is also Security Ambassador for Sun.