CyberSafe
TrustBroker™ Security Server

RSA SecurID Ready Implementation Guide

Last Modified: June 12, 2008

Partner Information

Product Information	
Partner Name	CyberSafe Limited
Web Site	www.cybersafe.com
Product Name	TrustBroker™ Security Server
Version & Platform	4.3
Product Description	The TrustBroker™ Security Server is an authentication server that implements the Kerberos protocol, and is also known as a Kerberos Key Distribution Center, or KDC. Its primary function is to manage principal identities and encryption keys stored in a database on the server, and respond to requests for Kerberos tickets for these principals. It makes sure that a password policy is adhered to, and that changes are propagated between multiple authentication servers on the network. It also detects and blocks *network message* and *authenticator* replay attacks. The product supports password-based authentication for user principals, and can also use the RSA Authentication Manager to authenticate a user principal with an RSA SecurID token passcode or tokencode.
Product Category	Authentication, Smart Cards, Tokens

CYBERSAFE
The Kerberos Solution Provider

Solution Summary

The TrustBroker™ Security Server, used in conjunction with the TrustBroker™ Secure Client products, uses Kerberos pre-authentication to allow the user to obtain an initial Kerberos Ticket (also known as the Ticket Granting Ticket, or TGT), using their RSA SecurID token passcode or tokencode, as well as their principal name and Kerberos password.

The TrustBroker™ Security Server includes the RSA Authentication Agent software, so that no RSA software needs to be installed on the same host; however the `sdconf.rec` file created by the RSA Authentication Manager is required on the same host, in order to allow TrustBroker™ Security Server to find an RSA Authentication Manager on the network, and communicate with it securely when authenticating a user.

The TrustBroker™ Secure Client products are required, and include the functionality to prompt the user for their tokens passcode or tokencode (if their RSA SecurID token is in tokencode-only mode). Also, when synchronizing the token, the user is required to enter the next tokencode from their token, and the TrustBroker™ Secure Client products have the functionality to ask the user for this information and send it back to the TrustBroker™ Security Server, which then provides this information to the RSA Authentication Manager.

Using this solution means that users and/or TrustBroker™ Security Server administrators can authenticate using their RSA SecurID tokens, rather than just a password. The added security given when using two-factor authentication improves the overall security of applications that are being accessed using Kerberos credentials.

Many companies are now using Kerberos to benefit from improved security and Secure Single SignOn, and many vendors have already added, or are adding Kerberos support to their products. This increased usage of Kerberos within networks, will undoubtedly mean an increasing need for stronger user authentication, and TrustBroker™ Security Server provides this. It also includes the following:

- Interoperability with Microsoft Active Directory, which also implements Kerberos Key Distribution Center functionality. It is possible for TrustBroker™ products to request tickets from an Active Directory domain, or use one-way, or two-way trust with an Active Directory domain.

- Allows Kerberos enabled versions of telnet, ftp, rsh, and ssh to take advantage of the stronger two-factor user authentication, provided by the RSA SecurID token. The TrustBroker™ Secure Connection Pack product provides Kerberos-enabled versions of these common utilities/applications, and can be used once users have authenticated using their RSA SecurID token or any other supported authentication method.

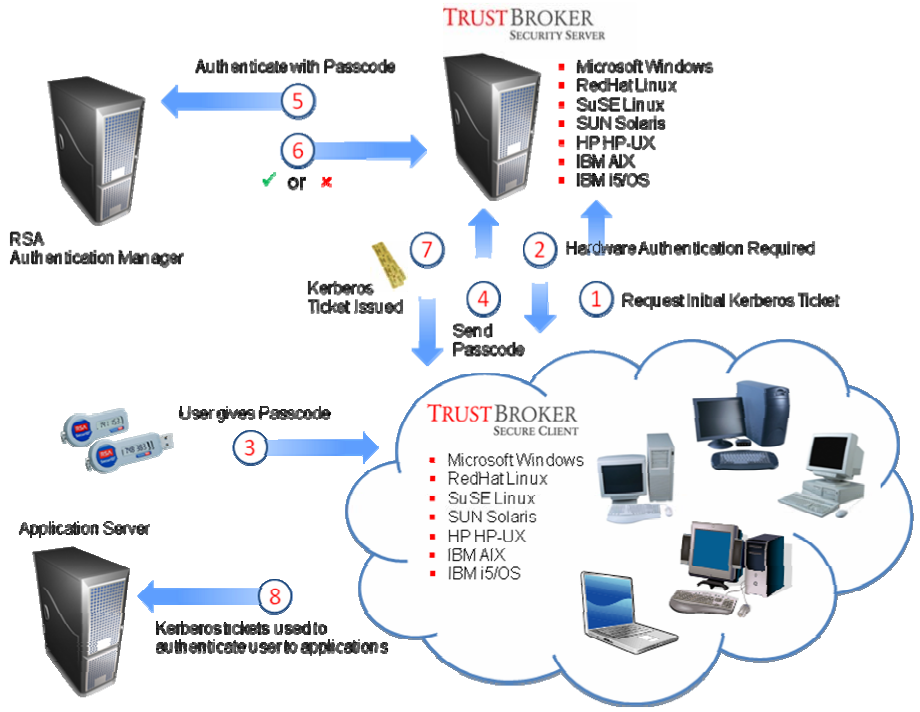
- Allows secure authenticated access to Kerberos-enabled business applications, e.g. products from SAP, and products that use Sybase, or Oracle databases.

An example of how TrustBroker™ Security Server can be used with RSA SecurID tokens and RSA Authentication Manager is given below:

Requirement: Many system administrators need to logon to their company's UNIX or Linux systems and perform administrative tasks when logged on.

Solution: A system administrator would normally need to know the correct password for users on each UNIX or Linux system, e.g. the root user password. Using TrustBroker™ Secure Client with TrustBroker™ Security Server, and TrustBroker™ Secure Connection Pack the administrator only needs to authenticate once on their workstation using their RSA SecurID token passcode, Kerberos password, and principal name. Once authenticated, they can use their Kerberos credentials to logon to one or more systems, and if authorized to do so, logon as root – without knowing the root password. This avoids the need to share the root password between administrators, and strengthens the authentication for administrative access. It also gives the administrator a single sign-on experience.

Partner Integration Overview	
Authentication Methods Supported	Native RSA SecurID Authentication
List Library Version Used	5.0.3
RSA Authentication Manager Replica Support *	Full Replica Support
Secondary RADIUS Server Support	N/A
RSA Authentication Agent Host Type	UNIX
RSA SecurID User Specification	Designated Users
RSA SecurID Protection of Administrative Users	Yes
RSA Software Token and RSA SecurID 800 Automation	Yes (no Automation),



Product Requirements

Partner Product Requirements: TrustBroker™ Security Server	
Version	**Product(s)**
4.3	TrustBroker™ Secure Client for Servers TrustBroker™ Secure Client for Workstations TrustBroker™ Application Security Runtime Library

Operating System	
Platform	**Required Patches**
RedHat Enterprise Linux 4 or 5 on x86_64	All patch levels supported
Novell SuSE Linux Enterprise 8 or 9 on x86_64	All patch levels supported
Novell SuSE Linux Enterprise 8 or 9 on PowerPC	All patch levels supported
SUN Solaris 8,9 or 10 on Sparc	All patch levels supported
SUN Solaris 10 on x86_64	All patch levels supported
HP HP-UX 11i on PA-RISC	All patch levels supported
HP HP-UX 11i on IA-64	All patch levels supported
IBM AIX on PowerPC	All patch levels supported
IBM i5/OS (using PASE) on PowerPC	All patch levels supported
IBM zLinux (SuSE Linux Enterprise 9) on s390	All patch levels supported
IBM zLinux (SuSE Linux Enterprise 9) on s390x	All patch levels supported
IBM zLinux (RedHat Enterprise Linux 4) on s390	All patch levels supported
IBM zLinux (RedHat Enterprise Linux 4) on s390x	All patch levels supported
Microsoft Windows 2000 SP4 on x86	All patch levels supported
Microsoft Windows Server 2003 on x86	All patch levels supported
Microsoft Windows Server x64 Edition on x86_64	All patch levels supported

Additional Software Requirements	
Application	**Additional Patches**
None	N/A

Agent Host Configuration

To facilitate communication between the TrustBroker™ Security Server and the RSA Authentication Manager / RSA SecurID Appliance, an Agent Host record must be added to the RSA Authentication Manager database. The Agent Host record identifies the TrustBroker™ Security Server within its database and contains information about communication and encryption.

To create the Agent Host record, you will need the following information.

- Hostname
- IP Addresses for all network interfaces

When adding the Agent Host Record, you should configure the host where TrustBroker™ Security Server is installed as UNIX. This setting is used by the RSA Authentication Manager to determine how communication with the TrustBroker™ Security Server will occur.

> **Note: Hostnames within the RSA Authentication Manager / RSA SecurID Appliance must resolve to valid IP addresses on the local network.**

Please refer to the appropriate RSA Security documentation for additional information about Creating, Modifying and Managing Agent Host records.

RSA SecurID files

The files listed in the table below should be created using the RSA Authentication Manager and stored on the servers where TrustBroker™ Security Server is installed. You will probably find that the `/var/ace` directory needs to be created before copying the files to this directory, unless the server has been used before by software that uses the RSA Authentication Agent.

RSA SecurID Authentication Files	
Files	**Location**
sdconf.rec	/var/ace/sdconf.rec
securid (node secret file)	/var/ace/securid
sdstatus.12	/var/ace/sdstatus.12
sdopts.rec	/var/ace/sdopts.rec

Partner Product Configuration

Before You Begin

This section provides instructions for integrating the TrustBroker™ products 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 and configure the required components.

Before attempting to enable RSA SecurID Authentication on the TrustBroker™ Security Server product, ensure that you have correctly configured the RSA Authentication Manager, created users, and assigned one or more tokens to the users.

The TrustBroker™ Secure Client and TrustBroker™ Security Server products must be installed and working prior to the integration. To confirm, you can use the `kinit` utility which was installed with the TrustBroker™ Secure Client product to check that it is possible to obtain an initial Kerberos ticket (e.g. TGT) for a principal (see example below).

Once this has been confirmed, you can proceed with the integration steps described later in this section.

```
# kinit testuser1
Password for testuser1@TEST:
#
# klist -efv
          Cache Type: Kerberos V5 Credentials Cache
          Cache File: /krb5/tmp/cc/krb5cc_0
       Cache Version: 0502
   Default Principal: testuser1@TEST

Valid From                Expires                   Service Principal
------------------------  ------------------------  -----------------
Sun Dec 16 11:18:40 2007  Sun Dec 16 19:18:32 2007  krbtgt/TEST@TEST
   Session Key EType: 23 (ARCFOUR-HMAC-MD5)
                         (ArcFour encryption, with HMAC, and MD5 checksum)
         Ticket EType: 23 (ARCFOUR-HMAC-MD5)
                         (ArcFour encryption, with HMAC, and MD5 checksum)
         Ticket Flags:  I (Initial)
                        A (Pre-Authenticated)
#
```

In the example above, the `klist` command is used to display the credentials (e.g. Kerberos tickets) stored in the users credentials cache, and issued by the TrustBroker™ Security Server when the correct password was entered for the principal.

Note: The instructions found in this section, and the examples provided assume that TrustBroker™ Secure Client and TrustBroker™ Security Server products are installed on UNIX or Linux systems. However, it should be noted that it is also possible to install Microsoft Windows versions of these products – if you have requirements which involve the use of Microsoft Windows please consult the CyberSafe TrustBroker™ product documentation, and/or contact CyberSafe Support for assistance.

Documenting the Solution

The steps listed below should be followed in order to configure the TrustBroker™ Security Server so users can obtain an initial Kerberos ticket (e.g. TGT) using RSA SecurID Authentication. Once the TGT has been obtained, it can then be used to request service tickets, and authenticate the user to Kerberos enabled applications, or services on the network.

Confirm the required files are in /var/ace directory

1. On each of the servers where TrustBroker™ Security Server is installed verify the ***sdconf.rec*** file is present.. This file is normally created by the RSA Authentication Manager software, and then copied to each server.

```
# ls /var/ace
sdconf.rec
#
```

2. Optionally, you might also want to create an ***sdopts.rec*** file (described in the RSA Authentication Manager Administrator's Guide) and put this file into the `/var/ace` directory with the ***sdconf.rec*** file.

3. If you find other files in the `/var/ace` directory, these might have been created if the TrustBroker™ Security Server has already been started on this server, in which case you can leave them, or remove them so that they are re-created when you start the TrustBroker™ Security Server daemons.

Configure TrustBroker™ Security Server for RSA SecurID Authentication

4. On each of the servers where TrustBroker™ Security Server is installed you need to change the startup script so that the TrustBroker™ RSA SecurID plugin is enabled. This is performed by editing the startup script called `/etc/rc.krb5` and removing the comment from the line where `–a securid` is shown, so that the `–a securid` option will be used when the script starts the KDC daemon process (`kdcd`).

```
#!/bin/sh
#
# @(#)Copyright (c) 1994-1997, 2000, CyberSafe Corporation.
# @(#)Copyright (c) 2001-2008, CyberSafe Limited.
# @(#)All rights reserved.
#
# Startup or Shutdown the CyberSafe TrustBroker Security Server daemons
#

# Customization section starts here

# Remove comment from one of the following to enable two-factor authentication
TWOFACTOR="-a securid";START2FACT=" with SecurID"
#TWOFACTOR="-a enigma";START2FACT=" with Enigma Logic"
#TWOFACTOR="-a vasco";START2FACT=" with Vasco"
```

Restarting the TrustBroker™ Security Server

5. In order for the KDC daemon to recognize and use the new `-a` startup option and parameter you need to restart the TrustBroker™ Security Server daemons. To do this, you can run the `/etc/rc.krb5` script with the `-f` option, or you can kill the existing `kdcd` daemon process and start it again from the command line.

```
# /etc/rc.krb5 -f
TrustBroker Key Distribution Center :
  Started kdcd daemon as PID 743.
TrustBroker Administration Server :
  Started admd daemon as PID 751.
TrustBroker Propagation System :
  Startup propd daemon as PID 752.
#
```

Checking syslog or log file messages

6. You need to check the syslog output (or log file if you are directing the daemon output to log files using the `-l` option) and look for messages similar to the following, indicating a successful startup.

```
$KDCD-S-00000518, Key Distribution Center starting...
$KDCD-S-00000548, kdcd, Version 4.3.0-33974. Copyright (c) 2003-2008, CyberSafe Limited. All
Rights Reserved.
$KDCD-S-0000063B, [SecurID]: RSA SecurID plugin, Version 2.1.0-33974. Copyright (c) 2003-2008,
CyberSafe Limited. All Rights Reserved.
$KDCD-S-0000063C, [SecurID]: Portions copyright (c) 1995-2008 RSA Security Inc. All Rights
Reserved. "SecurID" is either a registered trademark or trademark of RSA Security Inc. in the
United States and/or other countries.
```

Configuring principals

7. When a principal is added to the TrustBroker™ Security Server database, the principal is given a Kerberos password, which the user provides when authenticating, to get an initial Kerberos ticket. Each principal also has one or more attributes that can be set ON or OFF, one of which is the hardware authentication (`hwauth`) attribute. When the `hwauth` attribute is set ON, the TrustBroker™ Security Server knows that RSA SecurID Authentication will be required in order to issue an initial Kerberos ticket for the principal.

8. To change an existing principal in the TrustBroker™ Security Server so that it requires RSA SecurID Authentication, you can use a command such as:

```
# kadminl –C"mod –a hwauth <principal>"
```

9. If you are adding a new principal and want this principal to use RSA SecurID Authentication, you can use a command such as:

```
# kadminl –C"add –p <password> <principal>;mod –a nopwchg,hwauth <principal>"
```

Note: It is also possible to use the `kadmin` remote administration tool or the Windows or X-Windows GUI based administration tools, or a program that is coded using the functions provided by the TrustBroker™ Administration SDK.

Test RSA SecurID Authentication using kinit

10. In order to test the authentication you can now use the `kinit` command, as shown in the example below:

```
# kinit testuser1
Password for testuser1@TEST:
Enter Passcode (PIN+Tokencode) or Tokencode from your SecurID Token:
#
# klist -efv
          Cache Type: Kerberos V5 Credentials Cache
          Cache File: /krb5/tmp/cc/krb5cc_0
       Cache Version: 0502
   Default Principal: testuser1@TEST

Valid From                Expires                   Service Principal
------------------------  ------------------------  -----------------
Sun Dec 16 13:18:40 2007  Sun Dec 16 21:18:32 2007  krbtgt/TEST@TEST
   Session Key EType: 23 (ARCFOUR-HMAC-MD5)
                         (ArcFour encryption, with HMAC, and MD5 checksum)
        Ticket EType: 23 (ARCFOUR-HMAC-MD5)
                         (ArcFour encryption, with HMAC, and MD5 checksum)
        Ticket Flags:  I (Initial)
                       H (Hardware Authenticated)
                       A (Pre-Authenticated)
#
```

11. The syslog or daemon log file on the server running TrustBroker™ Security Server should show output similar to:

```
$KDCD-S-0000061E, [SecurID]: Authentication request received: user = testuser1, session id =
00100001.
$KDCD-D-40000627, [SecurID] [00100001]: Passcode or Tokencode required: user = testuser1.
$KDCD-W-8000058A, [AS]: Request: host=10.100.1.73 (sunny.dev.local), client=testuser1@TEST,
server=krbtgt/TEST@TEST, etype=(23), option=(C), patypes=(2,-19). Warning: Required hardware
authentication data not present (0x00000019)
$KDCD-D-4000061F, [SecurID] [00100001]: Passcode or Tokencode received: user = testuser1.
$KDCD-S-00000625, [SecurID] [00100001]: Authentication successful: user = testuser1.
$KDCD-S-00000589, [AS]: Request: host=10.100.1.73 (sunny.dev.local), client=testuser1@TEST,
server=krbtgt/TEST@TEST, etype=(23), option=(C), patypes=(-7,2,-19). Ticket Issued:
authtime=1197811120 (Sun Dec 16 13:18:40 2007), etypes={reply=23, session key=23, ticket=23},
kvno=0, flags=(I,H,A)
```

Test New PIN mode

12. In RSA Authentication Manager the token that is assigned to a user is edited, and “New PIN mode” is selected. Now, when the user authenticates using this token they get:

```
# kinit testuser1
Password for testuser1@TEST:
Enter Passcode (PIN+Tokencode) or Tokencode from your SecurID Token:
You need to change your PIN.
After changing your PIN, you MUST wait for the next Tokencode from your SecurID Token.
You may either specify your own new PIN or use the system generated PIN.
Your new PIN must contain between 4 and 8 alphanumeric characters.
Your new system generated PIN: m7hv
Please enter a new PIN (default: m7hv):
Enter Passcode (PIN+Tokencode) from your SecurID Token:
#
# klist -efv
          Cache Type: Kerberos V5 Credentials Cache
          Cache File: /krb5/tmp/cc/krb5cc_0
       Cache Version: 0502
```

```
   Default Principal: testuser1@TEST

Valid From                Expires                   Service Principal
------------------------  ------------------------  -----------------
Sun Dec 16 13:34:57 2007  Sun Dec 16 21:34:15 2007  krbtgt/TEST@TEST
   Session Key EType: 23 (ARCFOUR-HMAC-MD5)
                         (ArcFour encryption, with HMAC, and MD5 checksum)
         Ticket EType: 23 (ARCFOUR-HMAC-MD5)
                         (ArcFour encryption, with HMAC, and MD5 checksum)
         Ticket Flags:  I (Initial)
                        H (Hardware Authenticated)
                        A (Pre-Authenticated)
#
```

13. The syslog or daemon log file output should show output similar to:

```
$KDCD-S-0000061E, [SecurID]: Authentication request received: user = testuser1, session id =
00101807.
$KDCD-D-40000627, [SecurID] [00101807]: Passcode or Tokencode required: user = testuser1.
$KDCD-W-8000058A, [AS]: Request: host=10.100.1.73 (sunny.dev.local), client=testuser1@TEST,
server=krbtgt/TEST@TEST, etype=(23), option=(C), patypes=(2,-19). Warning: Required hardware
authentication data not present (0x00000019)
$KDCD-D-4000061F, [SecurID] [00101807]: Passcode or Tokencode received: user = testuser1.
$KDCD-D-40000629, [SecurID] [00101807]: New PIN required: user = testuser1.
$KDCD-W-8000058A, [AS]: Request: host=10.100.1.73 (sunny.dev.local), client=testuser1@TEST,
server=krbtgt/TEST@TEST, etype=(23), option=(C), patypes=(-7,2,-19). Warning: Preauthentication
required  (0x00000019)
$KDCD-D-40000621, [SecurID] [00101807]: New PIN received: user = testuser1.
$KDCD-S-0000062F, [SecurID] [00101807]: New PIN accepted: user = testuser1.
$KDCD-W-8000058A, [AS]: Request: host=10.100.1.73 (sunny.dev.local), client=testuser1@TEST,
server=krbtgt/TEST@TEST, etype=(23), option=(C), patypes=(-7,2,-19). Warning: Preauthentication
required  (0x00000019)
$KDCD-D-4000061F, [SecurID] [00101807]: Passcode or Tokencode received: user = testuser1.
$KDCD-S-00000625, [SecurID] [00101807]: Authentication successful: user = testuser1.
$KDCD-S-00000589, [AS]: Request: host=10.100.1.73 (sunny.dev.local), client=testuser1@TEST,
server=krbtgt/TEST@TEST, etype=(23), option=(C), patypes=(-7,2,-19). Ticket Issued:
authtime=1197812097 (Sun Dec 16 13:34:57 2007), etypes={reply=23, session key=23, ticket=23},
kvno=0, flags=(I,H,A)
```

Certification Checklist For RSA Authentication Manager

Date Tested: June 12, 2008

Certification Environment		
Product Name	**Version Information**	**Operating System**
RSA Authentication Manager	6.1.2	Windows 2003 SP1
TrustBroker	4.3.0	RedHat Enterprise Linux 4 x86_64

Mandatory Functionality			
RSA Native Protocol		**RADIUS Protocol**	
New PIN Mode			
Force Authentication After New PIN	✓	Force Authentication After New PIN	N/A
System Generated PIN	✓	System Generated PIN	N/A
User Defined (4-8 Alphanumeric)	✓	User Defined (4-8 Alphanumeric)	N/A
User Defined (5-7 Numeric)	✓	User Defined (5-7 Numeric)	N/A
User Selectable	✓	User Selectable	N/A
Deny 4 and 8 Digit PIN	✓	Deny 4 and 8 Digit PIN	N/A
Deny Alphanumeric PIN	✓	Deny Alphanumeric PIN	N/A
Passcode			
16 Digit Passcode	✓	16 Digit Passcode	N/A
4 Digit Password	✓	4 Digit Password	N/A
Next Tokencode Mode			
Next Tokencode Mode	✓	Next Tokencode Mode	N/A
Load Balancing / Reliability Testing			
Failover (3-10 Replicas)	✓	Failover	N/A
Name Locking Enabled	✓	Name Locking Enabled	
No RSA Authentication Manager	✓	No RSA Authentication Manager	N/A

Additional Functionality			
RSA Software Token Automation			
System Generated PIN	N/A	System Generated PIN	N/A
User Defined (8 Digit Numeric)	N/A	User Defined (8 Digit Numeric)	N/A
User Selectable	N/A	User Selectable	N/A
Next Tokencode Mode	N/A	Next Tokencode Mode	N/A
RSA SecurID 800 Token Automation			
System Generated PIN	N/A	System Generated PIN	N/A
User Defined (8 Digit Numeric)	N/A	User Defined (8 Digit Numeric)	N/A
User Selectable	N/A	User Selectable	N/A
Next Tokencode Mode	N/A	Next Tokencode Mode	N/A
Credential Functionality			
Determine Cached Credential State	N/A	Determine Cached Credential State	
Set Credential	N/A	Set Credential	
Retrieve Credential	N/A	Retrieve Credential	

CMY/PAR

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