IBM Ported Tools for z/OS: OpenSSH - Using Key Rings

June 19, 2012

Kirk Wolf
Steve Goetze

DovetailedTechnologies

http://dovetail.com
info@dovetail.com

Note: This webinar is a follow-on to:
"**IBM Ported Tools for z/OS: OpenSSH – Key Authentication**"
see: http://dovetail.com/webinars.html

Dovetailed Technologies

We provide z/OS customers world wide with innovative solutions that enhance and transform traditional mainframe workloads:

- Co:Z Co-Processing Toolkit for z/OS
- T:Z Quickstart for Tomcat and z/OS
- JZOS - acquired by IBM in 2005 and now part of the z/OS Java SDK

Co:Z Components

- Co:Z SFTP **
  - OpenSSH SFTP with z/OS exploitation
- Co:Z Batch
  - full featured BPXBATCH replacement
- Co:Z Dataset Pipes
  - convert datasets to streams / streams to datasets
  - other z/OS Unix commands / utilities
- Co:Z Launcher **
  - z/OS hybrid batch processing (distributed apps+data)
- Co:Z Target System Toolkit **
  - used with Co:Z Launcher and Dataset Pipes

** Requires IBM Ported Tools OpenSSH

Ported Tools for z/OS – OpenSSH (review)

- A port of OpenSSH for z/OS
  - z/OS Unix commands: **ssh, sshd, sftp, sftp-server**, etc.
- No support for MVS datasets, spool files, etc.
- Release 1.2 added support for:
  - ***SSH keys in SAF/RACF key rings***
  - SMF logging (new SMF 119 record types)
- PTF UA63842 added:
  - ICSF hardware acceleration for ciphers and MACs
- Co:Z Toolkit requires IBM Ported Tools OpenSSH:
  - Co:Z SFTP client invokes **ssh**
  - Co:Z SFTP server is invoked by **sshd**
  - Co:Z Launcher invokes **ssh**
  - Co:Z Dataset Pipes *can* be used remotely via **sshd**

Agenda

- Review SSH key authentication
  - protecting key material
  - benefits of using digital certificates with SSH
- Creating RACF digital certificates
  - for either server host key or a user key
  - exporting the OpenSSH public key
  - using the certificate's private key with OpenSSH
  - required permissions for using certificates
- ICSF/hardware-protected User keys
  - Using the private key without reading it
  - Co:Z **saf-ssh-agent** key ring support
- Best practices for z/OS SSH keys

Other security packages

- In this presentation, we will illustrate RACF commands for managing key rings and digital certificates
- IBM Ported Tools OpenSSH uses standard SAF and R_datalib interfaces to key rings and certificates
- Other security packages (CA-ACF2, CA-TSS) also support certificates and will work with IBM Ported Tools OpenSSH
  - RACF commands must be translated to commands in your security package

SSH Key Authentication Review

- Host (server) authentication
  - Server (SSHD) has host private key(s)
  - Clients have matching host public key
    - known_hosts is a list of: `<host -> public key>`
- User key authentication
  - User has private key
  - Server has matching public key
    - $HOME/.ssh/authorized_keys is a list of the user's public keys

Safeguarding private key material

- Only root (UID=0) should be able to read the host private key.
- Only the client userid should be able to read the User private key
  - better yet: even the user can't read it, only *use* it

Safeguarding public key registries

- There is no danger in allowing anyone to read a public key
  - but the “registry” files that contain them must be write protected.
- On the client:
  - `~/.ssh/known_hosts` - only updateable by the client userid
  - `/etc/ssh/ssh_known_hosts` – only updateable by root (but readable by everyone)
- On the server:
  - `~/.ssh/authorized_keys` - only updateable by server userid

See “Common Pitfalls” in Part 1 or Co:Z SFTP User's Guide for a detailed list of file permission settings

Digital certificate advantages

- The private key is not in a file. SAF profiles/rules are used to protect access.
- Private keys ***can*** be stored in ICSF / hardware
  - The userid can ***use*** the private key, but can't ***read*** it
  - IBM Ported Tools doesn't currently support this *directly*
  - Co:Z saf-ssh-agent can be used as an “agent” to IBM Ported Tools OpenSSH client for User keys (more later)
- Ported Tools SSH can also use public keys from certificates
  - `known_hosts` or `authorized_keys` files still need to point to them

Note: The SSH standard does not use X.509 cert chains
- z/OS digital certificate is only used as a keys container
- “self-signed” certificates are fine

Creating a RACF key ring and certificate

```
RACDCERT ID(ALICE) GENCERT
           SUBJECTSDN(CN('Alice Kingsleigh'))
           SIZE(2048)NOTAFTER(DATE(2015-01-01))
           WITHLABEL('SSH01')
RACDCERT ID(ALICE) ADDRING(SSH-RING)
RACDCERT ID(ALICE) CONNECT ( ID(ALICE)
             RING(SSH-RING) LABEL('SSH01')
             DEFAULT USAGE(PERSONAL))
```

- This generates a “self-signed” certificate with RSA key pair
- Key ring and cert labels must be unique within a userid
- Default expiry is one year if `NOTAFTER` is not specified
- Other `SUBJECTSDN` keywords may be specified
- The certificate “`ALICE/SSH-RING SSH01`” has the RSA public key; the private key is held in the RACF database. (more later on storing private key in ICSF/hardware)

Exporting the public key from a certificate

- On the client z/OS system:

```
export _ZOS_SSH_KEY_RING_LABEL="ALICE/SSH-RING SSH01"
ssh-keygen -e > ssh01.pub
```

- Transfer text file `ssh01.pub` to the target system and import it:

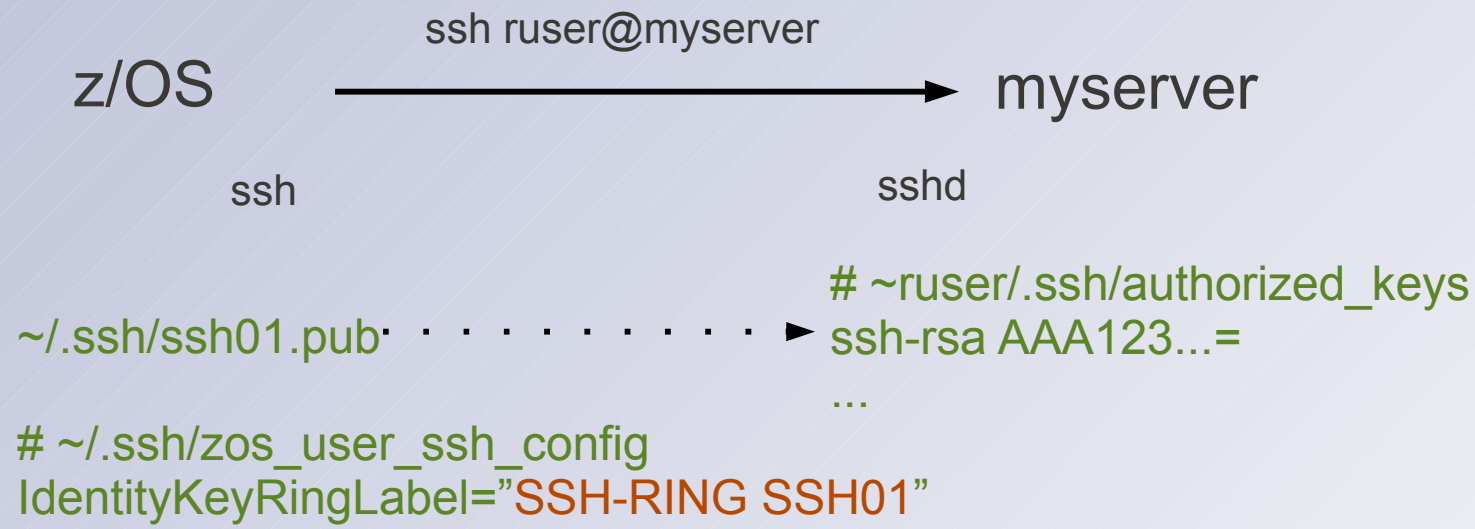
```
ssh-keygen -i -f ssh01.pub >> ~/.ssh/authorized_keys
```

"`-e`" exports the public key in the RFC-4716 text file format.
"`-i`" converts a RFC-4716 format key to OpenSSH format

- Alternatively, the following Co:Z command directly exports the OpenSSH format public key:

```
saf-ssh-agent -x -f ssh01.pub ALICE/SSH-RING:SSH01
```

User authentication using RACF key ring



Note: The client file "ssh01.pub" is not actually used; the certificate has the public key and the private key is in the RACF database.

IdentityKeyRingLabel may also be specified as an option to the ssh or sftp client commands

Virtual Key Rings

- All certificates are automatically connected to the userid's “virtual key ring”
- May be referred to as key ring “*”
  - For example:

`"ALICE/* SSH01"`	(Alice's cert with label SSH01)
`"* SSH01"`	(The current user's cert with label SSH01)

Demo: User key auth with RACF certificate

Creating a RACF certificate for host auth

```
RACDCERT ID(SSHD) GENCERT
           SUBJECTSDN(CN('SSHD myserver.myco.com'))
           SIZE(2048) NOTAFTER(DATE(2015-01-01))
           WITHLABEL('SSHD01')
```

- This is really no different than creating a User key
- The certificate can be referred to as “SSHD/* SSHD01” using virtual key ring syntax. (virtual rings also work with User keys)
- Exporting the OpenSSH-format public key:

```
saf-ssh-agent -x -f sshd01.pub SSHD/*:SSHD01
```

(Note: saf-ssh-agent uses a colon to separate the label)

Server authentication using RACF key ring

ssh

```
# file: /etc/ssh/ssh_known_hosts
#   or ~/.ssh/known_hosts
myserver ssh-rsa  AA123....=
```

sshd

```
# /etc/ssh/zos_sshd_config
HostKeyRingLabel “SSHD/* SSHD01”
```

/etc/ssh/sshd01.pub

Notes: public key file on the server is not actually used by sshd –
This file is exported from the RACF certificate “SSHD/* SSHD01”
in case you want to pre-distribute to your client known_host files

Permissions for using a Key Ring

- Ring-specific profiles

  `CLASS(RDATALIB) <userid>.<ring-name>.LST`

  `ACCESS(READ)` - userid's own key ring

  `ACCESS(UPDATE)` - another user's key ring

  For the virtual key ring (any certificate for the user):
  `<userid>.IRR_VIRTUAL_KEYRING.LST`

Note: CLASS RDATALIB must be active and RACLIST refreshed

Permissions for using a Key Ring (cont.)

- Global profiles
  - used if there isn't a matching specific profile

```
CLASS(FACILITY) IRR.DIGTCERT.LISTRING
   ACCESS(READ)   - userid's own key rings
   ACCESS(UPDATE) - all user's key rings (Yikes!)
```

Note: CLASS FACILITY must be active and RACLISTed

Creating a certificate with ICSF private key

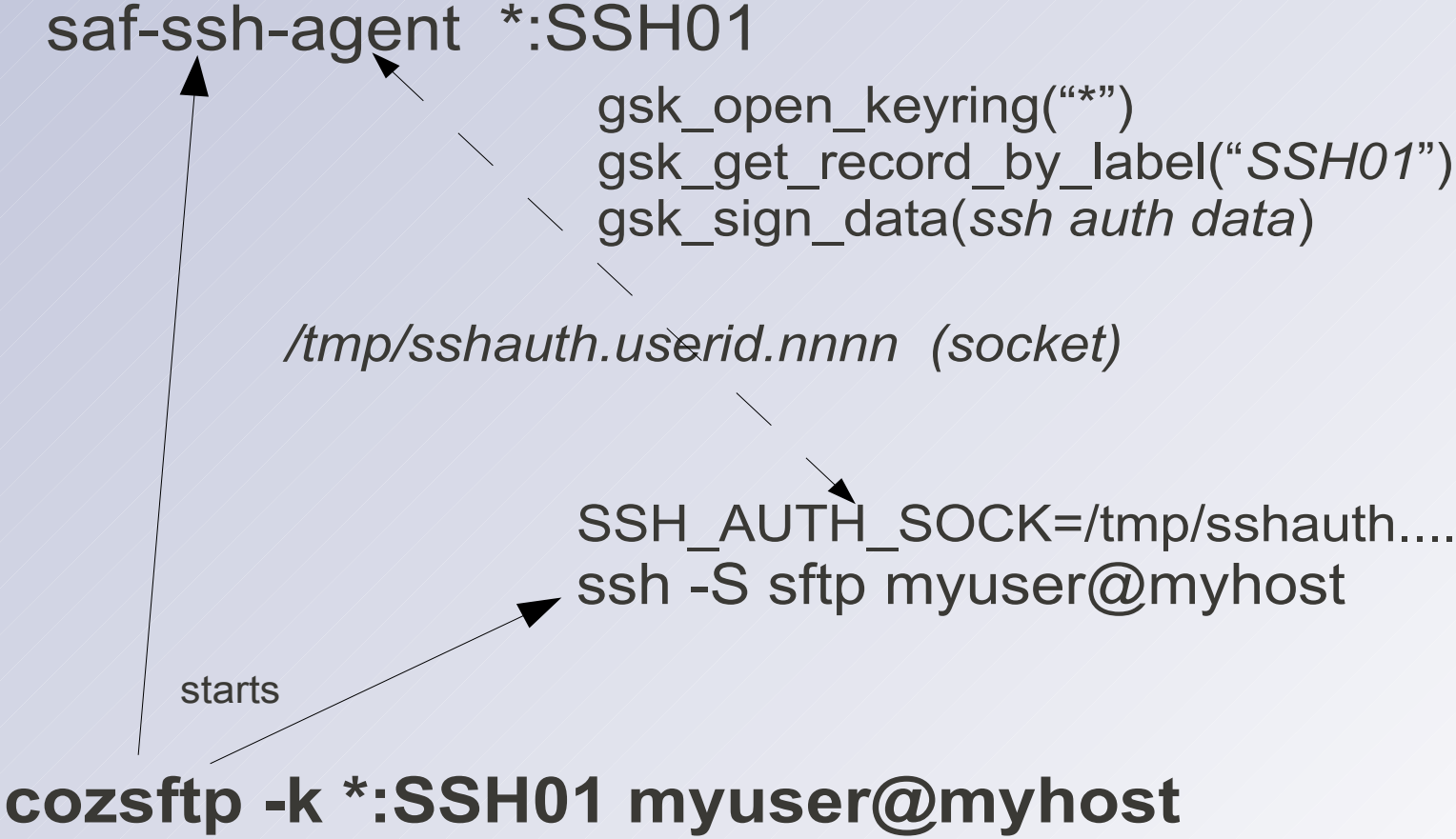
```
RACDCERT ID(ALICE) GENCERT
                SUBJECTSDN(CN('Alice Kingsleigh'))
                SIZE(2048)NOTAFTER(DATE(2015-01-01))
                WITHLABEL('SSH01')
                PCICC
```

- Can also use ***ICSF*** in place of ***PCICC*** **–** check doc for details
- By default, the private key is stored in PKDS with same label
- ***Key difference:*** A user **can not read** the private key from his own certificate
- ***But:*** Ported Tools OpenSSH can not (on its own) use a certificate private key unless it can read it.

Using Co:Z saf-ssh-agent for User key auth

- Uses the OpenSSH “ssh-agent” protocol to act as a user key authentication agent for Ported Tools ssh client
- ssh client uses a private Unix-domain socket to communicate with saf-ssh-agent:
  - agent tells ssh which public key is has
  - ssh sends agent a signature request
  - agent signs the data ***using*** the certificate private key
    - without actually reading the private key
- saf-ssh-agent is managed automatically by Co:Z -
  - `cozsftp -k ring:label myuser@myhost`
  - Co:Z Launcher option: `saf-cert=ring:label`
  - or manually as a command “wrapper”:
    `saf-ssh-agent -c ring:label ssh … user@host`

Co:Z sftp with saf-ssh-agent with ssh client



Additional permissions for hardware keys

- To ***use*** private keys in ICSF/hardware:

```
CLASS(CSFSERV) CSFDSG ACCESS(READ)
CLASS(CSFSERV) CSFDSV ACCESS(READ)
```

Note: CLASS CSFSERV must be active and RACLIST refreshed

Note: Actually, these permissions are always required for `saf-ssh-agent`, even if the private key is not in hardware since it uses these ICSF APIs for signatures (and exploits a co-processor if you have one).

Also see: “SA22-7521 - ICSF Administrator's Guide” - “Using RACF to protect Keys and Services”

Best Practices with z/OS SSH keys

- Consider using digital certificates for security-sensitive SSH private keys
  - z/OS Ported Tools sshd Host keys
  - z/OS Ported Tools ssh User keys
- Implement procedures to manage expired certificates
  - either renew certificates before they expire
  - or, transition with two authorized keys at once – the old, and the new
- Use ICSF managed hardware for best SSH private key security, using Co:Z saf-ssh-agent (User keys only)
- Implement procedures to manage and distribute host public keys

More information

- IBM Ported Tools for z/OS: OpenSSH User's Guide
- Co:Z SFTP User's Guide
- http://dovetail.com/forum (public bulletin board)
- Our webinar archives: http://dovetail.com/webinars
- Previous webinar (part 1):

  "*IBM Ported Tools for z/OS: OpenSSH - Key Authentication*"