Syscheck Installation and Upgrade

Henrik Andreasson

kinneh@users.sourceforge.net

http://sourceforge.net/apps/trac/syscheck/

2010-11-09

Version 1.0

Table of Contents

1 New Installation

1.1 Get Syscheck

```
username@smartcard20-node1:/var/tmp/ wget http://sourceforge.net/projects/syscheck/files/syscheck-1.5.15/syscheck-1.5.15.zip/download
```

1.2 Unpack Syscheck

```
username@smartcard20-node1:/usr/local/# unzip  /var/tmp/syscheck-1.5.15.zip

[...]
```

1.3 Syscheck configuration

Configure the values in config/common.conf and the scripts you're using.

1.3.1 Activate selected syscheck scripts

```
username@smartcard20-node1:/usr/local/certificate-services/syscheck/scripts-enabled> sudo ln -s ../scripts-available/sc_02_ejbca.sh .

username@smartcard20-node1:/usr/local/certificate-services/syscheck/scripts-enabled> sudo ln -s ../scripts-available/sc_03_memory-usage.sh .

username@smartcard20-node1:/usr/local/certificate-services/syscheck/scripts-enabled> sudo ln -s ../scripts-available/sc_07_syslog.sh .
```

And so on for each syscheck-script you want active.

Verify there is soft links, no file copied into scritps-enabled!

```
username@smartcard20-node1:/usr/local/certificate-services/syscheck/scripts-enabled> ls -l

lrwxrwxrwx  1 root root   35 2010-06-04 14:25 sc_02_ejbca.sh -> ../scripts-available/sc_02_ejbca.sh

lrwxrwxrwx  1 root root   42 2010-06-04 14:25 sc_03_memory-usage.sh -> ../scripts-available/sc_03_memory-usage.sh

lrwxrwxrwx  1 root root   36 2010-06-04 14:25 sc_07_syslog.sh -> ../scripts-available/sc_07_syslog.sh

lrwxrwxrwx  1 root root   35 2010-06-04 14:25 sc_12_mysql.sh -> ../scripts-available/sc_12_mysql.sh

lrwxrwxrwx  1 root root   35 2010-06-04 14:25 sc_19_alive.sh -> ../scripts-available/sc_19_alive.sh

lrwxrwxrwx  1 root root   45 2010-06-04 14:25 sc_20_errors_ejbcalog.sh -> ../scripts-available/sc_20_errors_ejbcalog.sh
```

Also check the config for each script you activate, there may or may not be anything to config, but let's check.

```
username@smartcard20-node1:/usr/local/certificate-services/syscheck/> sudo vi config/02.conf

username@smartcard20-node1:/usr/local/certificate-services/syscheck/> sudo vi config/03.conf

username@smartcard20-node1:/usr/local/certificate-services/syscheck/> sudo vi config/07.conf

username@smartcard20-node1:/usr/local/certificate-services/syscheck/> sudo vi config/12.conf

username@smartcard20-node1:/usr/local/certificate-services/syscheck/> sudo vi config/19.conf

username@smartcard20-node1:/usr/local/certificate-services/syscheck/> sudo vi config/20.conf
```

1.3.2 Activate selected related scripts

Make the soft links in the related-enabled directory.

```
han@rp-ca-nod1:/usr/local/certificate-services/syscheck/related-enabled> ln -s ../related-
available/904_make_mysql_db_backup.sh .
```

Verify there is soft links and no copied files in enabled!

```
username@smartcard20-node1:/usr/local/certificate-services/syscheck/syscheck/related-enabled> ls -al

lrwxrwxrwx  1 root root   48 2010-06-04 14:25 904_make_mysql_db_backup.sh -> ../related-
available/904_make_mysql_db_backup.sh
```

Also check the config for each script you activate, there may or may not be anything to config, but let's check.

```
username@smartcard20-node1:/usr/local/certificate-services/syscheck/> sudo vi config/900.conf

username@smartcard20-node1:/usr/local/certificate-services/syscheck/> sudo vi config/904.conf

[...]
```

1.3.3 Mysql database configuration

Config database parameters, you can get the database-username/password information from ejbca it's in the file ejbca/conf/database.properties

```
less /usr/local/ejbca/conf/database.properties

database.url=jdbc:mysql://127.0.0.1:3306/ejbca

database.username=ejbca

database.password=sdfiuh3wrnj
```

Enter those parameters into the syscheck config/common.sh

```
DB_NAME=ejbca

DB_USER=ejbca

DB_PASSWORD="sdfiuh3wrnj"
```

If your installation has a weak(eg. foo123) mysql root password set a new with high security (eg. UiywfeW23)

```
# mysqladmin password  <new-pass> -u root --password=<old-password>

# mysqladmin password UiywfeW23 -u root –password=foo123
```

Alternativly use the mysql_secure_installation, this command removes test users and databases and sets a new mysql-root password

```
# mysql_secure_installation
```

Enter the new mysql root password into syscheck/config/common.conf

```
#Password for Mysql root

MYSQLROOT_PASSWORD="UiywfeW23"
```

1.4 Make the new version the default

```
username@smartcard20-node1:/usr/local> sudo rm syscheck

username@smartcard20-node1:/usr/local> sudo ln -s syscheck-1.5.11 syscheck
```

2 Upgrade

2.1 Get Syscheck

```
username@smartcard20-node1:/var/tmp/ wget http://sourceforge.net/projects/syscheck/files/syscheck-1.5.15/syscheck-1.5.15.zip/download
```

2.2 Unpack Syscheck

```
username@smartcard20-node1:/usr/local/# unzip  /var/tmp/syscheck-1.5.15.zip

[...]
```

2.3 Migration of config from a previous version

If migrating a host to new hardware make a backup of syscheck to be transferred to the new hardware.

```
root@smartcard20-node1:/usr/local # zip -r9 /tmp/syscheck-backup-x.y.z.zip syscheck-x.y.z
```

Transfer the zip to the new host, then unpack the backup.

```
root@smartcard20-node1:/usr/local # unzip /tmp/syscheck-backup-x.y.z.zip
```

Copy enabled scripts

```
root@smartcard20-node1:/usr/local/syscheck # cp -a ../syscheck-<last-version>/related-enabled/* ./ related-enabled/
```

root@smartcard20-node1:/usr/local/syscheck # **cp -a ../syscheck-<last-version>/scritps-enabled/* ./ scritps-enabled/**

Run the copy config command to loop through the configs and check wheter you want to use ther old or the new config.
Tip: Use the old if it's configured and only differs to the new config is the values you entered. Is there new options you need, use the new one and migrate the config manually
root@smartcard20-node1:/usr/local/syscheck # **./lib/copy-config-from-old-version.sh /path/to/old/syscheck-<last-version>/config ./config/**

Migrate old resouces.sh to the new common.conf

Note: ONLY NEEDED IF UPGRADING FROM 1.4.0 or earlier

root@smartcard20-node1:/usr/local/syscheck # **diff -uw ../syscheck-<last-version>/resources.sh ./config/common.conf**

Go though the differences and manually enter the changes you need to keep into the new common.conf

2.4 Make the new version the default

username@smartcard20-node1:/usr/local> **sudo rm syscheck**

username@smartcard20-node1:/usr/local> **sudo ln -s syscheck-1.5.15 syscheck**

3 SSH Keys Installation

This chapter assumes that your application user is called “jboss”. The purpose of the installation of ssh-keys is to automaticly be able to run commands on the other node in a cluster.

3.1 Generate keys on both nodes

on node1 run:
username@smartcard20-node1:/usr/local> sudo su - jboss

jobss@smartcard20-node1:~> ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/jboss/.ssh/id_rsa):

Created directory '/home/jboss/.ssh'.

Enter passphrase (empty for no passphrase): **(press enter to use no passphrase)**

Enter same passphrase again:

Your identification has been saved in /home/jboss/.ssh/id_rsa.

Your public key has been saved in /home/jboss/.ssh/id_rsa.pub.

on node2 run:
username@smartcard20-node2:/usr/local> sudo su - jboss

jobss@smartcard20-node2:~> ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/jboss/.ssh/id_rsa):

Created directory '/home/jboss/.ssh'.

Enter passphrase (empty for no passphrase): **(press enter to use no passphrase)**

Enter same passphrase again:

Your identification has been saved in /home/jboss/.ssh/id_rsa.

Your public key has been saved in /home/jboss/.ssh/id_rsa.pub.

3.2 Add the own public key to the own authorized_keys to allow local ssh

on node1 run:

jboss@smartcard20-node1:~> **cp .ssh/id_rsa.pub .ssh/authorized_keys**

jboss@smartcard20-node1:~> ssh localhost

The authenticity of host 'localhost (127.0.0.1)' can't be established.

RSA key fingerprint is 73:c8:8f:20:73:83:62:c7:5e:a0:7d:cb:38:c8:04:43.

Are you sure you want to continue connecting (yes/no)? **yes**

Warning: Permanently added 'localhost' (RSA) to the list of known hosts.

jboss@smartcard20-node1:~> logout

on node2 run:

jboss@smartcard20-node2:~> **cp .ssh/id_rsa.pub .ssh/authorized_keys**

jboss@smartcard20-node2:~> ssh localhost

The authenticity of host 'localhost (127.0.0.1)' can't be established.

RSA key fingerprint is 73:c8:8f:20:73:83:62:c7:5e:a0:7d:cb:38:c8:04:43.

Are you sure you want to continue connecting (yes/no)? **yes**

Warning: Permanently added 'localhost' (RSA) to the list of known hosts.

jboss@smartcard20-node2:~> logout

3.3 Add node1 public key to node2 authorized_keys to allow ssh from node1 to node2

on node1 run:

jboss@smartcard20-node1:~> cat .ssh/id_rsa.pub

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAqAxIMkP3or7xGr/PTSpomW+zUh9Id5yL6GKhaoctgI1wJtM7wbiQhq4V6VX
mL2onHIKDAJTJ+A10qzy8+iolCmAjnHbVw+Xy3YhPbQuT21dvntTuXtv+4xMgeAagbAg6hRI+3qcidmLkQGsSvJncJkJ/

```
vftcw0e0rbRToDEiS3jbK6VFBo+k3mccBlPG3KqlvM6YoHajoAkxU11IIWY+sesnsfOPMjwt7bfXqT32XBNzIpAwBIJVQ
0cBIs+Sz551hR5ERvWW9Qp8nl4hcEwVSrsiflyGJKxI7AnMxzaIJdfoBsZTj/jUtDpK/PnGuJzNMkwQlSmdCND+N2QW7
n9WxQ== jboss@smartcard20-node1
```

on node2 run:

```
jboss@smartcard20-node2:~> vi .ssh/authorized_keys

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAqAxIMkP3or7xGr/PTSpomW+zUh9Id5yL6GKhaoctgI1wJtM7wbiQhq4V6VX
mL2onHIKDAJTJ+A10qzy8+iolCmAjnHbVw+Xy3YhPbQuT21dvntTuXtv+4xMgeAagbAg6hRI+3qcidmLkQGsSvJncJkJ/
vftcw0e0rbRToDEiS3jbK6VFBo+k3mccBlPG3KqlvM6YoHajoAkxU11IIWY+sesnsfOPMjwt7bfXqT32XBNzIpAwBIJVQ
0cBIs+Sz551hR5ERvWW9Qp8nl4hcEwVSrsiflyGJKxI7AnMxzaIJdfoBsZTj/jUtDpK/PnGuJzNMkwQlSmdCND+N2QW7
n9WxQ== jboss@smartcard20-node2

~
```

- To add a new line below the only line in authorized_keys press “o”
- then paste the key from node1
- press escape to exit edit mode
- make sure that the new key is on one line only, press “J” to join to lines if the line broke in copy-and-paste, delete space with “x”
- press :wq to exit

Change the window size so any line-wraps problem reveals it self. Make sure each key is on one line

on node2 run:

```
jboss@smartcard20-node2:~> cat  .ssh/authorized_keys

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAqAxIMkP3or7xGr/PTSpomW+zUh9Id5yL6GKhaoctgI1wJtM7wbiQhq4V6VX
mL2onHIKDAJTJ+A10qzy8+iolCmAjnHbVw+Xy3YhPbQuT21dvntTuXtv+4xMgeAagbAg6hRI+3qcidmLkQGsSvJncJkJ/
vftcw0e0rbRToDEiS3jbK6VFBo+k3mccBlPG3KqlvM6YoHajoAkxU11IIWY+sesnsfOPMjwt7bfXqT32XBNzIpAwBIJVQ
0cBIs+Sz551hR5ERvWW9Qp8nl4hcEwVSrsiflyGJKxI7AnMxzaIJdfoBsZTj/jUtDpK/PnGuJzNMkwQlSmdCND+N2QW7
n9WxQ== jboss@smartcard20-node2

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEArzafoK1mzu0WYr1c0I8efdkOE5+58ccM83fJ32EuUxKfrvEi3GsJKyZPvuSZr4
VXUgRvyyWzg8JtwJR7s+8NSqcHhdeN+zeVQx9/YCw8xOQLx9SOORato76WfwhjM7kUrZd/bIQQGRDwh1rvCVRlNr7
kFFRsEe+afxP42AZkZaD6bCBsG4BF/iR5+YCERzuUymrN5g8qWU5Hw1Hegedz+7oCJZ8FO3UVB6SREDA1n4c651g
Wa+xtx+5KgWLcfQU8V019fzuVz45c1mF0r0zQF4Z66gq/HEkYUjttk7hgCrRaeIR1GTu+9HNF2+aSV44ZqKtd5L1cX1xs
xJCJijjnSw== han@smartcard20-node1

jboss@smartcard20-node2:~> wc -l  .ssh/authorized_keys

2 .ssh/authorized_keys
```

verify ssh access to the other node:

```
jboss@smartcard20-node1:~> ssh jboss@smartcard20-node2

Last login: Sat Jun  5 08:41:53 2010 from localhost

jboss@smartcard20-node2:~>
```

3.4 Add node2 public key to node1 authorized_keys to allow ssh from node2 to node1

on node2 run:

```
jboss@smartcard20-node2:~> cat .ssh/id_rsa.pub

ssh-rsa
```

```
AAAAB3NzaC1yc2EAAAABIwAAAQEAqAxIMkP3or7xGr/PTSpomW+zUh9Id5yL6GKhaoctgI1wJtM7wbiQhq4V6VX
mL2onHIKDAJTJ+A10qzy8+iolCmAjnHbVw+Xy3YhPbQuT21dvntTuXtv+4xMgeAagbAg6hRI+3qcidmLkQGsSvJncJkJ/
vftcw0e0rbRToDEiS3jbK6VFBo+k3mccBlPG3KqlvM6YoHajoAkxU11IIWY+sesnsfOPMjwt7bfXqT32XBNzIpAwBIJVQ
0cBIs+Sz551hR5ERvWW9Qp8nl4hcEwVSrsiflyGJKxI7AnMxzaIJdfoBsZTj/jUtDpK/PnGuJzNMkwQlSmdCND+N2QW7
n9WxQ== jboss@smartcard20-node2
```

on node1 run:

```
jboss@smartcard20-node1:~> vi .ssh/authorized_keys


ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAqAxIMkP3or7xGr/PTSpomW+zUh9Id5yL6GKhaoctgI1wJtM7wbiQhq4V6VX
mL2onHIKDAJTJ+A10qzy8+iolCmAjnHbVw+Xy3YhPbQuT21dvntTuXtv+4xMgeAagbAg6hRI+3qcidmLkQGsSvJncJkJ/
vftcw0e0rbRToDEiS3jbK6VFBo+k3mccBlPG3KqlvM6YoHajoAkxU11IIWY+sesnsfOPMjwt7bfXqT32XBNzIpAwBIJVQ
0cBIs+Sz551hR5ERvWW9Qp8nl4hcEwVSrsiflyGJKxI7AnMxzaIJdfoBsZTj/jUtDpK/PnGuJzNMkwQlSmdCND+N2QW7
n9WxQ== jboss@smartcard20-node1


~
```

- To add a new line below the only line in authorized_keys press “o”
- then paste the key from node2
- press escape to exit edit mode
- make sure that the new key is on one line only, press “J” to join to lines if the line broke in copy-and-paste, delete space with “x”
- press :wq to exit

Change the window size so any line-wraps problem reveals it self. Make sure each key is on one line

on node2 run:

```
jboss@smartcard20-node1:~> cat  .ssh/authorized_keys


ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAqAxIMkP3or7xGr/PTSpomW+zUh9Id5yL6GKhaoctgI1wJtM7wbiQhq4V6VX
mL2onHIKDAJTJ+A10qzy8+iolCmAjnHbVw+Xy3YhPbQuT21dvntTuXtv+4xMgeAagbAg6hRI+3qcidmLkQGsSvJncJkJ/
vftcw0e0rbRToDEiS3jbK6VFBo+k3mccBlPG3KqlvM6YoHajoAkxU11IIWY+sesnsfOPMjwt7bfXqT32XBNzIpAwBIJVQ
0cBIs+Sz551hR5ERvWW9Qp8nl4hcEwVSrsiflyGJKxI7AnMxzaIJdfoBsZTj/jUtDpK/PnGuJzNMkwQlSmdCND+N2QW7
n9WxQ== jboss@smartcard20-node1

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEArzafoK1mzu0WYr1c0I8efdkOE5+58ccM83fJ32EuUxKfrvEi3GsJKyZPvuSZr4
VXUgRvyyWzg8JtwJR7s+8NSqcHhdeN+zeVQx9/YCw8xOQLx9SOORato76WfwhjM7kUrZd/bIQQGRDwh1rvCVRlNr7
kFFRsEe+afxP42AZkZaD6bCBsG4BF/iR5+YCERzuUymrN5g8qWU5Hw1Hegedz+7oCJZ8FO3UVB6SREDA1n4c651g
Wa+xtx+5KgWLcfQU8V019fzuVz45c1mF0r0zQF4Z66gq/HEkYUjttk7hgCrRaeIR1GTu+9HNF2+aSV44ZqKtd5L1cX1xs
xJCJijjnSw== han@smartcard20-node2

jboss@smartcard20-node2:~> wc -l  .ssh/authorized_keys


2 .ssh/authorized_keys
```

verify ssh access to the other node:

```
jboss@smartcard20-node2:~> ssh smartcard20-node1


Last login: Sat Jun  5 08:41:53 2010 from localhost


jboss@smartcard20-node1:~>
```

4 References

4.1 Syscheck mysql database backup management

See the instruction for mysql database backup “syscheck-backup-management.pdf”

4.2 Using syscheck for database replication and failover

See the instruction in “database_replication_and_failover.pdf”

4.3 Using syscheck for certificate and revovation archival

See the instruction in “certificate_and_revovation_archival.pdf”