NOMACHINE	How to use different keys or certificates with NoMachine	
Prepared by: Silvia Regis	**N°**: D-705_011-HWT-SDF	
Approved by: Sarah Dryell	**Last modified**: 2018-05-15	**Amended**: A

How to use different keys or certificates with NoMachine

Table of Contents

Introduction

1. Keys and Certificates

Keys and Certificates are used to secure the communication between two entities. In a NoMachine infrastructure, communications are established between:

I the end-user's device (NoMachine client or the browser) and the NoMachine server to which the user connects;
II the end-user's device and the end-point host machine, e.g. a NoMachine server federated under a Cloud Server;
III the Cloud Server and the NoMachine servers federated under this Cloud Server (multi-server environment);
IV the Enterprise Terminal Server and its Terminal Server Nodes (multi-node environment);
IV two NoMachine servers in a failover cluster.

All these communications channel are encrypted and protected by Certificates or pairwise Keys provided with the NoMachine installation. If you may wish to replace any of them with your own Keys and Certificates, please follow instructions in the next sections.

TIPS

I All instructions are intended to be run from console and require an account with administrative privileges: 'root' on Linux and Mac (use a 'sudo' user if you don't have the 'root' account on your system and and add the *sudo* prefix to all commands) and an administrator user on Windows. On Windows, execute the CMD shell as administrator.

II Instructions use the NoMachine nxkeygen tool, as an alternative, you can use the standard ssh-keygen command from OpenSSH.

III By default, keys and certificates are generated with 2048 bit lenght, specify the -n option for a different lenght.

III Instructions refers to *installation directory* which is the installation directory of the NoMachine server, by default:
/usr/NX on Linux
/Applications/NoMachine.app/Contents/Frameworks/ on Mac
/C:\PROGRA~1\NoMachine on Windows, i.e. C:\Program Files (x86)\NoMachine on 64bit systems or C:\Program Files\NoMachine on 32bit systems.

2. The SSL certificate for nxd

The nxd program is the NoMachine Network Daemon resident on the server host (any of the NoMachine servers and the Terminal Server Node) necessary to accept connections through NX protocol. Its SSL certificate is made of:
installation directory/etc/keys/host/nx_host_rsa_key.crt
installation directory/etc/keys/host/nx_host_rsa_key

CA certificates will be supported with the implementation of https://www.nomachine.com/FR02L02810.

How to generate and use a new certificate and private key
STEP 1- generate a new certificate and private key for nxd. The general format of the is:

```
installation directory/bin/nxkeygen -k privatekey -c certificate [-n length]

On Linux and Mac it's necessary to set LD_LIBRARY_PATH, for example:

LD_LIBRARY_PATH=/usr/NX/lib
installation directory/bin/nxkeygen -k nx_host_rsa_key -c nx_host_rsa_key.crt -n 4096
```

Ensure that the new certificate and key have the same name of the original ones and proper permissions and ownership.
On **Linux** they should look like:
-rw------- 1 nx root 1675 2013-11-18 12:18 nx_host_rsa_key
-rw-r--r-- 1 nx root 1090 2013-11-18 12:18 nx_host_rsa_key.crt
and on **Mac**:
-rw------- 1 nx wheel 1679 Apr 8 16:21 nx_host_rsa_key

-rw-r--r-- 1 nx wheel 1090 Apr 8 16:21 nx_host_rsa_key.crt

Restarting nxd is not necessary.

STEP 2- For **web sessions** it's necessary to update the client.crt by adding content of the new certificate nx_host_rsa_key.crt.

The certificate is:
/var/NX/nxhtd/.nx/config/client.crt on Linux
/Library/Application\ Support/NoMachine/var/nxhtd/.nx/config/client.crt on Mac
C:\ProgramData\NoMachine\nxhtd\.nx\config\client.crt on Windows.

For example on Linux, if the new certificate is placed in /usr/NX:

```
echo "Host:localhost" > /var/NX/nxhtd/.nx/config/client.crt

cat /usr/NX/etc/keys/host/nx_host_rsa_key.crt >> /var/NX/nxhtd/.nx/config/client.crt

echo "Host:127.0.0.1" >> /var/NX/nxhtd/.nx/config/client.crt

cat /usr/NX/etc/keys/host/nx_host_rsa_key.crt >> /var/NX/nxhtd/.nx/config/client.crt
```

Both entries for Host:localhost and Host:127.0.0.1 must be present in client.crt which should look like:
Host:localhost
-----BEGIN CERTIFICATE-----
MIIC9zCCAd+gAwIBAgIRAP4YLqSxLm9xey/k41vmu+cwDQYJKoZIhvcNAQEFBQAw
(.....)
-----END CERTIFICATE-----
Host:127.0.0.1
-----BEGIN CERTIFICATE-----
MIIC9zCCAd+gAwIBAgIRAP4YLqSxLm9xey/k41vmu+cwDQYJKoZIhvcNAQEFBQAw
(....)
-----END CERTIFICATE-----

3. The SSL certificate for nxhtd

The nxhtd program is the NoMachine Apache-based web server included in any of the NoMachine server installations (except NoMachine free) and necessary for accepting connections by the web. In case of multi-node environments (Enterprise Terminal Server + Terminal Server Nodes) it's provided by the Enterprise Terminal Server and it's not installed on the remote nodes.

Installation comes with a self-signed a SSL Certificate File and SSL Certificate Key file intended to be just a sample. They are, respectively:
installation directory/etc/keys/host/ht_host_rsa_key.crt
installation directory/etc/keys/host/ht_host_rsa_key

Administrators have to replace the sample SSL Certificate File and Key File with their own certificate self-signed or acquired from a CA.

How to generate a new certificate
The general format of the command to generate a new certificate and private key for nxhtd is:

```
installation directory/bin/nxkeygen -k privatekey -c certificate [-n length]

On Linux and Mac it's necessary to set LD_LIBRARY_PATH:

export LD_LIBRARY_PATH=/usr/NX/lib
installation directory/bin/nxkeygen -k new_ht_host_rsa_key -c new_ht_host_rsa_key.crt
```

How to use the new certificate
STEP 1- Edit the nxhtd configuration file to point to the new certificate.
Let's assume that the new certificate is made of: new_ht_host_rsa_key.crt and new_ht_host_rsa_key.

Edit the *installation directory*/etc/htd.cfg file and set:
SSLCertificateFile "*installation directory*/etc/keys/host/new_ht_host_rsa_key.crt"
SSLCertificateKeyFile "*installation directory*/etc/keys/host/new_ht_host_rsa_key"

STEP 2- To make changes effective, restart nxhtd. This will terminate all running web sessions.

To restart nxhtd, run from console:

```
installation directory/bin/nxserver --restart nxhtd

On Linux and Mac you can use:

/etc/NX/nxserver --restart nxhtd
```

Otherwise you can restart the nxhtd program from the NoMachine Server preferences GUI.

STEP 3- On Linux and Mac it's necessary to update certificate permissions. Run:

```
installation directory/bin/nxwebplayer --update
```

When executing "nxwebplayer --update", the nxhtd server is automatically restarted.

As an alternative, you can update permissions by hand to have:
--rw------- 1 nxhtd nxhtd 1,7K lis 20 18:40 new_ht_host_rsa_key
-rw-r--r-- 1 nxhtd nxhtd 1,1K lis 20 18:39 new_ht_host_rsa_key.crt

Then it's necessary to manually restart nxhtd.

4. The RSA key pair for nxsshd

The nxsshd program is the NoMachine SSH server installed on Windows by any of the enterprise packages (NoMachine free doesn't have it). It's necessary to accept connections through the SSH protocol.

Its RSA keys are:
installation directory/etc/keys/host/ssh_host_rsa_key
installation directory/etc/keys/host/ssh_host_rsa_key.pub

How to generate a new certificate
To generate a new SSH key pair, run from the CMD console:

```
installation directory/bin/nxkeygen -k ssh_host_rsa_key -p ssh_host_rsa_key.pub
```

How to use the new certificate
STEP 1- Configure nxsshd to use a different private key by editing this file:
installation directory/etc/sshd_config

Uncomment and set te proper value for the HostKey configuration key.

For example, if the RSA key is placed at: "C:\Program Files (x86)\NoMachine\etc\keys\host\new_ssh_host_rsa_key":
- edit the sshd_config file
- uncomment the '#HostKey /etc/ssh/ssh_host_rsa_key' entry (i.e. remove '#')
- and change this key to the appropriate value:
HostKey "C:\Program Files (x86)\NoMachine\etc\keys\host\new_ssh_host_rsa_key"

STEP 2- Then, it's necessary to restart nxsshd. This can be easily done via the NoMachine Server preferences GUI.

TIP

The public key must be stored with the same file name of the private key but with .pub as postfix. For example, if the new private key is new_rsa_key, the public key must be named as new_rsa_key.pub

5. The RSA key pair for the Terminal Server Nodes

NoMachine Enterprise Terminal Server authenticates on the Terminal Server Node with a RSA key pair. This RSA key pair is generated during the installation and its server specific. This means that if the node is added to a different server, also the RSA key pair will be different.

This key pair is made of:
installation directory/etc/keys/node.localhost.id_rsa
installation directory/etc/keys/node.localhost.id_rsa.pub

When adding the node to the server (by means of 'nxserver --nodeadd' command), the public part of

the key pair is automatically added to the remote node.

The RSA key is deleted from the node when the node is removed from the list (with the 'nxserver --nodedel' command).

The 'nxserver --nodeupdate NODENAME:PORT' command will add the new key (if set on the main server host) on the node, but it will not remove the old key.

To adopt a custom key pair for server-node authentication, follow all steps below. If not otherwise specified, commands are intended to be run on the Enterprise Terminal Server host.

STEP 1- Generate a new RSA key pair and name the keys as node.localhost.id_rsa and node.localhost.id_rsa.pub (i.e. the new keys must have the same name of the original ones):

```
export LD_LIBRARY_PATH=/usr/NX/lib
/usr/NX/bin/nxkeygen -k /usr/NX/etc/keys/node.localhost.id_rsa -p /usr/NX/etc/keys/node.localhost.id_rsa.pub -t rsa
```

Then ensure that proper permissions and ownership are set:

```
chmod 600 /usr/NX/etc/keys/node.localhost.id_rsa

chown nx:root /usr/NX/etc/keys/node.localhost.id_rsa

chmod 644 /usr/NX/etc/keys/node.localhost.id_rsa.pub

chown nx:root /usr/NX/etc/keys/node.localhost.id_rsa.pub
```

STEP 2- Stop the server to prevent users from starting new sessions while replacing the server public key on the nodes. This will not terminate running sessions:

```
/etc/NX/nxserver --stop
```

STEP 3- Make a backup of the original RSA keys on the server machine in *installation directory*/etc/keys.

STEP 4- Place the new RSA key pair in the same directory, *installation directory*/etc/keys.

STEP 5- Propagate the new RSA sever public key on the node by running:

```
/etc/NX/nxserver --nodeupdate NODENAME:PORT

where NODENAME:PORT is the name of the remote node as it appears in the output of the 'nxserver --nodelist' command.
```

The 'nxserver --nodeupdate' command will not remove the old key on the node. To remove it, delete the node:

```
/etc/NX/nxserver --nodedel NODENAME:PORT
```

```
and re-add it so that the new key will be automatically added:

/etc/NX/nxserver --nodeadd NODENAME
```

As an alternative, it's possible to execute a manual procedure to remove the old RSA public key **(*)** and add the new one **(**)** on the remote Terminal Server Node.

STEP 6- Only if you have two NoMachine servers in a **failover cluster**, update the cluster configuration to synchronize the new RSA key pair. Run on the primary or on the secondary server the following command:

```
/etc/NX/nxserver --clusterupdate
```

(*) How to manually remove the old server RSA key from the node
Removing the old server RSA keys by hand is an alternative to deleting and re-adding the node when replacing the default server-node RSA key pair.

In a particular case, i.e. if the server is unable to connect to the node while executing 'nxserver --nodedel', it's necessary to adopt this manual procedure as well. That's because the node is removed from the NoMachine db but the server key is left on the node host.

To remove the old server public key manually:
***1)** On the server host read the current server RSA key that is going to be replaced:*

```
cat /usr/NX/etc/keys/node.localhost.id_rsa.pub
```

***2)** On each of the node hosts remove the line containing the current server public key from the following files:*
nx_home_directory/.nx/config/authorized.crt for server-node connections by NX protocol and nx_home_directory/.ssh/authorized_keys2 for server-node connections by SSH protocol.

(**) How to add the new RSA public key (node.localhost.id_rsa.pub) on the remote node
To add a RSA public key on node, you can run the following command on the node host:

```
/etc/NX/nxserver --keyadd public_key_file

where public_key_file is path to the new node.localhost.id_rsa.pub key.
```

This will add the key to the authorized.crt file if server-node protocol is NX or to the authorized keys file if server-node protocol is SSH.

Alternatively, the RSA server public key can be manually added to the proper files:
nx_home_directory/.nx/config/authorized.crt if server-node protocol is NX and
nx_home_directory/.ssh/authorized_keys2 if server-node protocol is SSH

'authorized_keys2' is the standard name used in the SSHD configuration, replace it with the appropriate name if your SSHD has custom settings.

6. The RSA Key Pair and the SSL certificate for the Failover Cluster

The failover cluster uses (i) a SSH key pair to authenticate each other the primary and secondary server and (ii) a SSL certificate on the shared IP to avoid users having to accept again the server host fingerprint when the failover occurs.

(i) How to generate and use a new SSH key pair for the primary-secondary server authentication

You can generate a new SSH key pair on the primary server host. Name the new keys as the original ones:

```
installation directory/bin/nxkeygen -k installation directory/etc/keys/cluster.id_rsa -p installation directory/etc/keys/cluster.id_rsa.pub -t rsa

On Linux and Mac it's necessary to set LD_LIBRARY_PATH, for example:

export LD_LIBRARY_PATH=/usr/NX/lib
/usr/NX/bin/nxkeygen -k /usr/NX/etc/keys/cluster.id_rsa -p /usr/NX/etc/keys/cluster.id_rsa.pub -t rsa
```

Then ensure that the new keys have proper permissions and ownership. For example on Linux:

```
chmod 600 /usr/NX/etc/keys/cluster.id_rsa

chown nx:root /usr/NX/etc/keys/cluster.id_rsa

chmod 644 /usr/NX/etc/keys/cluster.id_rsa.pub

chown nx:root /usr/NX/etc/keys/cluster.id_rsa.pub
```

How to use the new key-pair
Propagate the new key to the secondary server by running on the primary server the following command:

```
installation directory/bin/nxserver --clusterupdate
```

(ii) How to generate and use a new SSL certificate and private key on cluster shared IP

The SSL certificate used for connections by NX protocol when the failover cluster is set-up is made of:

<installation directory>/etc/keys/host/nx_cluster_rsa_key
<installation directory>/etc/keys/host/nx_cluster_rsa_key.crt

Â

To generate a new certificate and key:

```
installation directory/bin/nxkeygen -k privatekey -c certificate [-n length]

On Linux and Mac it's necessary to set LD_LIBRARY_PATH, for example:

export LD_LIBRARY_PATH=/usr/NX/lib
installation directory/bin/nxkeygen -k nx_cluster_rsa_key -c nx_cluster_rsa_key.crt -n 4096
```

Then ensure that the new keys have proper permissions and ownership. On Linux they should look like:

-rw------- 1 nx root 1675 2013-11-18 12:18 nx_cluster_rsa_key
-rw-r--r-- 1 nx root 1090 2013-11-18 12:18 nx_cluster_rsa_key.crt

and on Mac:

-rw------- 1 nx wheel 1679 Apr 8 16:21 nx_cluster_rsa_key
-rw-r--r-- 1 nx wheel 1090 Apr 8 16:21 nx_cluster_rsa_key.crt

How to use the new certificate
Propagate the new certificate to the secondary server by running on the primary server the following command:

```
installation directory/bin/nxserver --clusterupdate
```

7. The RSA key pair for the Federated Servers

NoMachine Cloud Server authenticates on any of the federated servers with a RSA key pair. This RSA key pair is generated during the installation and its server specific. This means that if the server is added to a different Cloud Server, also the RSA key pair will be different.

This key pair is made of:
installation directory/etc/keys/node.localhost.id_rsa
installation directory/etc/keys/node.localhost.id_rsa.pub

When adding a server to the Cloud Server (by means of 'nxserver --serveradd' command), the public part of the key pair is automatically added to the remote server host.

The RSA key is deleted from the server host when the server is removed from the multi-server environment (with the 'nxserver --serverdel' command).

The 'nxserver --serverupdate SERVERNAME:PORT' command will add the new key (if set on the Cloud

Server host) on the server, but it will not remove the old key.

To adopt a custom key pair for the authentication, follow steps below.

Instructions apply to Cloud server and its first-level servers. If the Cloud Server is a sub-level server, be sure to execute instructions on its host and not on the main Cloud Server.

STEP 1- Make a backup of the original RSA keys on the Cloud Server machine in *installation directory*/etc/keys.

STEP 2- Generate a RSA key pair and name keys as node.localhost.id_rsa and node.localhost.id_rsa:

```
installation directory/bin/nxkeygen -k installation directory/etc/keys/node.localhost.id_rsa -p installation directory/etc/keys/node.localhost.id_rsa.pub -t rsa

On Linux and Mac it's necessary to set LD_LIBRARY_PATH, for example:

export LD_LIBRARY_PATH=/usr/NX/lib
installation directory/bin/nxkeygen -k installation directory/etc/keys/node.localhost.id_rsa -p installation directory/etc/keys/node.localhost.id_rsa.pub -t rsa
```

Ensure also that the new keys have proper permissions and ownership, as the original ones.

STEP 3- Stop the Cloud Server to prevent users from starting new sessions while replacing the server public key on the federate servers. This will not terminate running sessions.

```
installation directory/bin/nxserver --stop
```

STEP 4- Place the new RSA key pair in the same directory, *installation directory*/etc/keys.

STEP 5- Propagate the new RSA sever public key on the federated server by executing on the Cloud Server host:

```
installation directory/bin/nxserver --serverupdate SERVERNAME:PORT

where SERVERNAME:PORT is the name of the federated server as it appears in the output of the 'nxserver --serverlist' command.
```

Note that this command will not remove the old key on the federated server. To remove it, delete the server:

```
installation directory/bin/nxserver --serverdel SERVERNAME:PORT

and re-add it so that the new key will be automatically added:

installation directory/bin/nxserver --serveradd SERVERNAME:PORT
```

As an alternative, it's possible to execute a manual procedure to remove the old RSA public key (*) and add the new one (**) on the federated server.

STEP 6- Only if you have two Cloud Servers in a **failover cluster**, update the cluster configuration to synchronize the new RSA key pair. Run on the primary or on the secondary server the following command:

```
installation directory/bin/nxserver --clusterupdate
```

(*) How to manually remove the old server RSA key from the federated server host
1) *On the Cloud Server host read the current server RSA key that is going to be replaced:*

```
cat installation_directory/etc/keys/node.localhost.id_rsa.pub
```

2) *On each of the federated server hosts remove the line containing the current server public key from the following files:*
nx_home_directory/.nx/config/authorized.crt for server-to-server connections by NX protocol and
nx_home_directory/.ssh/authorized_keys2 for server-to-server connections by SSH protocol.

(**) How to add the new RSA public key (node.localhost.id_rsa.pub) on the federated server
To add a RSA public key on the federated server, you can run the following command on the federated server:

```
/etc/NX/nxserver --keyadd public_key_file
```

This will add the key to the authorized.crt file if server-to-server protocol is NX or to the authorized keys file if server-to-server protocol is SSH.

Alternatively, the RSA server public key can be manually added to the proper files:
nx_home_directory/.nx/config/authorized.crt if protocol is NX and
nx_home_directory/.ssh/authorized_keys2 if protocol is SSH.

'authorized_keys2' is the standard name used in the SSHD configuration, replace it with the appropriate name if your SSHD has custom settings.